

**Introduction to deep learning**

**Midterm project report**

**YOLO for Medical Image Segmentation - Case study: Lung cancer Nodules**

Group 9’s members:

|  |  |
| --- | --- |
| Vũ Đức Duy | 22BI13127 |
| Nguyễn Hoài Anh | 22BI13021 |
| Nguyễn Việt Anh | 22BI13032 |
| Phạm Hoàng Anh | 22BI13034 |
| Lâm Chí Cường | 22BI13068 |
| Nguyễn Át | 22BI13047 |

Contents

[**I.** **Introduction** 3](#_Toc178616074)

[**II.** **Dataset** 3](#_Toc178616075)

[**1.** **Data visualization** 3](#_Toc178616076)

[**2.** **Dataset description** 3](#_Toc178616077)

[**3.** **Example** 4](#_Toc178616078)

[**III.** **Model** 5](#_Toc178616079)

[**IV.** **Result** 6](#_Toc178616080)

[**V.** **Resources** 6](#_Toc178616081)

# **Introduction**

**You Only Look Once (YOLO)** is a real-time object detection model that is widely used in computer vision tasks and is known for its speed and accuracy in detecting objects within images and videos. YOLO proposes using an end-to-end neural network that makes predictions of class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection.

In this project, we use **YOLO-v8** for **Medical Image Segmentation – Lung Cancer Nodules**. This report focuses on applying **YOLOv8**, the latest version of the series of real-time object detectors, offering cutting-edge performance in terms of accuracy and speed. We specifically investigate how well it can detect lung cancer nodules in medical images.

# **Dataset**

## **Data visualization**

Our data has 343 images divided into 70% for training, 20% for validating and 10% for testing, each image size is 640 x 640 pixels. The data set is segmented in YOLOv8-seg and each test has 1 validation actual and prediction.

## **Dataset description**

-       x\_train contains 70% grayscale images.

-       y\_train corresponds to 70% labels denoting classes for each image in

x\_train

-       x\_test holds 10% grayscale images.

-       y\_test consists of 10% labels indicating classes for the images in x\_test.

## **Example**

In this test, we have run a small sample and here is the result and specific data for YOLOv8-seg and:

|  |  |  |
| --- | --- | --- |
| YOLOv8-seg |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Size  (pixels) | mAPbox  50-95 | mAPmask  50-95 | Speed CPU  ONNX  (ms) | Speed A100  TensorRT  (ms) | params(M) | FLOPs(B) |
| YOLOv8x-seg | 640 | 53.4 | 43.4 | 712.1 | 4.02 | 71.8 | 344.1 |

# **Model**

The design of YOLOv8 is based on earlier versions of YOLO algorithms. YOLOv8 makes use of a convolutional neural network (CNN) that is comprised of two main components: the backbone and the head. The foundation of YOLOv8 is based on an altered iteration of the CSPDarknet53 design. This structure includes 53 convolutional layers and utilizes partial crossover phase connections to enhance communication between the various layers. YOLOv8's main component is made up of numerous convolutional layers and is followed by several fully connected layers. These layers predict bounding boxes, objectness score and class probabilities for detected objects in an image. A crucial aspect of YOLOv8 is the incorporation of a self-attention mechanism in the network's head. This mechanism enables the model to concentrate on various sections of the image and modify the significance of different characteristics depending on their relevance to the task. Another significant aspect of YOLOv8 is its capability to conduct object detection at multiple scales. The model uses a feature pyramid network to identify objects of various sizes and scales in an image. This feature pyramid network is made up of several layers that identify objects of various sizes, enabling the model to recognize both big and small objects in an image.

# **Result**

* **YOLOv8-seg**

|  |  |
| --- | --- |
|  |  |
|  |  |

* **The bar chart images:**

The bar chart shows scatter plots and a bar chart. It indicates there are around 250 instances labeled as "Tumor". The scatter plots show the distribution of data points based on x, y coordinates and width/height.

* **F1-Confidence Curve**

Shows the balances precision and recall, peaks at around 0.91 at a confidence threshold of 0.326.

* **Recall-Confidence Curve**

Demonstrates a Recall-Confidence Curve. It shows how the recall of the model changes with different confidence thresholds. The curve indicates high recall at low confidence levels, which gradually decreases as confidence increases.

* **Precision-Confidence Curve**

Illustrates a Precision-Confidence Curve. The precision of the model increases with higher confidence thresholds, reaching 1.0 at a confidence of 0.678.

* **Precision-Recall Curve:**

The last image is a Precision-Recall Curve. It illustrates the trade-off between precision and recall as the model's decision threshold is varied. The curve shows high precision across most recall values, with an area under the curve (mAP@0.5) of 0.937.

# **Resources**

1. [https://universe.roboflow.com/kfupm-7hvnm/lung-tumor-2xcgn/dataset/3](https://universe.roboflow.com/kfupm-7hvnm/lung-tumor-2xcgn/dataset/3?fbclid=IwY2xjawFm90xleHRuA2FlbQIxMAABHYwwOtgz_DjNWOFzzugF4XLkM18bBUsU7up4vf6suhC79WiRC-TIs4lYOg_aem_Wd0kkIv-VnPttCsEvbt8ng)
2. [https://docs.ultralytics.com/model s/yolov8/](https://docs.ultralytics.com/model%09s/yolov8/)
3. <https://docs.python.org/3/library/shutil.html>
4. <https://docs.python.org/3/library/os.html>
5. <https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html>