

UNIVERSITY *of* WASHINGTON

Technology Review

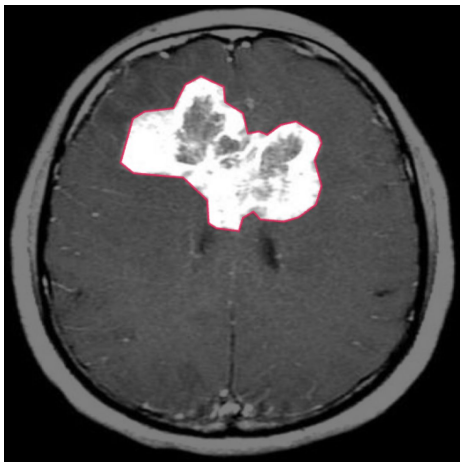
Elizabeth Holden | Elliott Sanger | Salah Elbakri | Diane Chiang



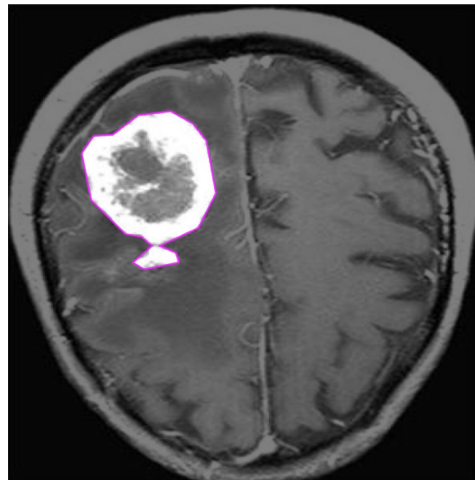
Background & Use Case

GOAL: BUILD WEBSITE TO UPLOAD BRAIN CT AND RETURNS SEVERITY OF BRAIN TUMOR

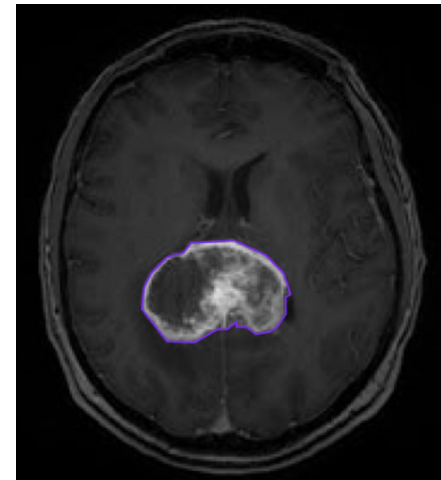
- > Need a model that can predict the seriousness of brain tumors given the brain image
- > Use instance segmentation (object detection + classification) to categorize the severity of a brain tumor



less chance



moderate chance



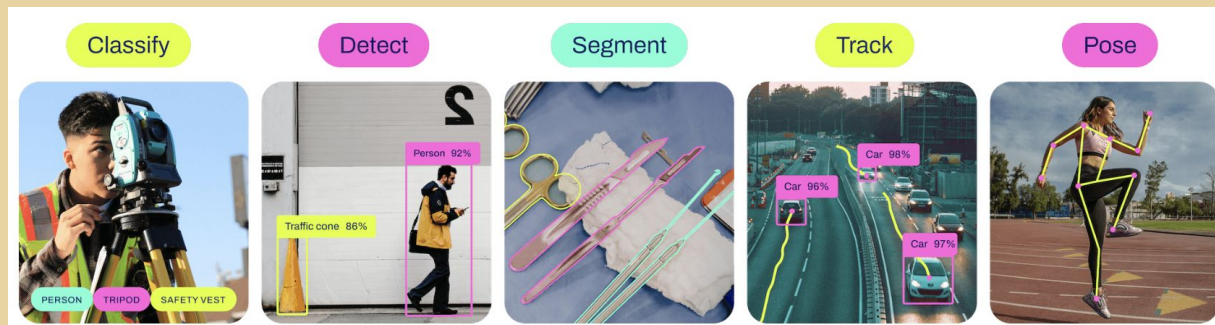
high chance



Python Package Choices 1-2

Choice 1: Ultralytics YOLOv8

- > Used for wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks.



Choice 2: Mask R-CNN

- > Instance segmentation that performs pixel-level segmentation on detected objects

Package Comparison: YOLOv8 vs. Mask R-CNN

YOLOv8

Pros:

- > High accuracy
- > Easy to use
- > Well documented pipeline

Cons:

- > Specific required label format (ex. coordinates needs to be between $(-1, 1)$; might need to rescale the label)
- > Long training time of 3000+ brain CTs (100 epochs took 17+ hours)

Mask R-CNN

Pros:

- > Flexibility on model configuration
- > Prediction on pretrained model

Cons:

- > Pre processing of image dataset required
- > Longer training time than YOLOv8 according to research



Our Choice: YOLOv8

- > Well defined functions for creating, training, and predicting with the model
- > No pre-processing of data needed due to the file format of training data
- > Created by UW alumni



Drawbacks & Remaining Concerns

- > Time consuming model training -> start training now
- > Low mAP (mean Average Precision) ~ 0.6 for instance segmentation
 - Scale back to semantics segmentation
- > “Black-box model” - with so many parameters and configurations available - it’s difficult to interpret the key features.
 - Research into YOLOv8 architecture and understand it
 - [YOLOv8 architecture explained](#)



Package Demonstration

LIVE DEMO

W