Organ Detection on CT Medical Images Using YOLO and UNet Frameworks

Anjia Wang and Yudong Sun

University of South Carolina, Columbia SC 29201, USA
{anjia,yudongs}@email.sc.edu

Abstract. Object detection and image segmentation are widely used for medical image processing nowadays. With a small data set, UNet itself, a CNN-based framework designed for medical image processing, may not provide good enough prediction. In the case of eye nerves, Unet cannot even provide meaningful results. To solve this problem, in this paper we proposed to combine YOLO object detection system and UNet together. A area containing marker object and goal object are detected by YOLO first. With the proposed approach, YOLO is able to detect human eyes with 86% accuracy. Then only this area is preserved and all the rest of image are wiped out. With the YOLO-preprocessed images in the same small data set, we significantly improved the prediction of UNet. It's proved that the combination of YOLO and UNet is a good approach to increase prediction accuracy under limited conditions.

Keywords: Object Detection · Image Segmentation · Convolutional Neural Network · YOLO · UNet

1 Introduction

Nowadays, the deep learning techniques are widely used in more and more fields, especially in medical industry due its huge amount data. CT images are collected to visually determine whether some certain organs of patients are unhealthy. It requires medical experts to check a large set of images in limited time, which brings lots of pressure due to insufficient resources. Therefore, to apply deep learning in this subject would be a good fit to accomplish the tasks more efficiently. Convotuional neural network is adopted and makes big contribution to object detection and image segmentation. YOLO, UNet and other frameworks are implemented to solve the real world problems.

This project is conducted to combine YOLO and UNet together to improve the prediction accuracy of organ detection on medical CT images. It provides better performance than applying only UNet framework.

2 Backgournd and Related Work

2.1 YOLO

YOLO is introduced in 2016, a new approach to object detection[1]. Before Yolo came into being, we repurposes classifiers to perform detection like R-

CNN. The detection system classifies the object and evaluates it at different locations and scales of the test image to detect an object. Now we can treat object detection as a regression problem to frame the bounding box of spatial separation and the associated class probability. In one assessment, a single neural network predicts bounding boxes and class probabilities directly from the full image. Yolo trains the complete image and can directly optimize the detection performance. The advantage of Yolo is simple and fast. Yolo sees the entire image during training and testing, so it implicitly encodes contextual information about the class and its appearance. The Yolo system divides the input image into S*S grid. The grid cell is reponsible for detecting the object whose center falls into the grid cell. Each grid cell predicts bounding boxes and confidence scores $(Pr(Object) * IOU_{pred}^{truth})$ in Fig. 1.

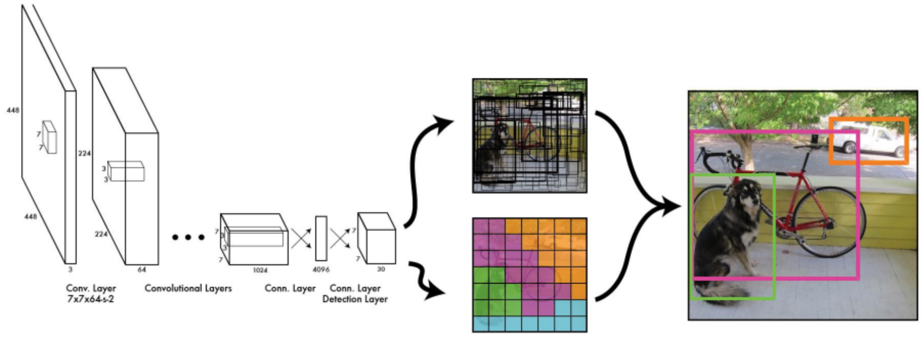


Fig. 1: YOLO CVPR 2016 (Redmon et al, 2016)

2.2 UNet

UNet is convolutional neural network developed at the Computer Science Department of the University of Freiburg, Germany in 2015[2]. It is implemented using Tensorflow and widely used for medical image segmentation. The architecture of UNet is shown in Fig. 2 and it has four types of layers.

The first layer is general convolutional layer using a 3×3 kernel. The second layer is max pooling layer for downsampling. It is designed to create feature maps. The third layer is to concatenate multiple feature maps. The fourth layer is upsampling to generate the output.

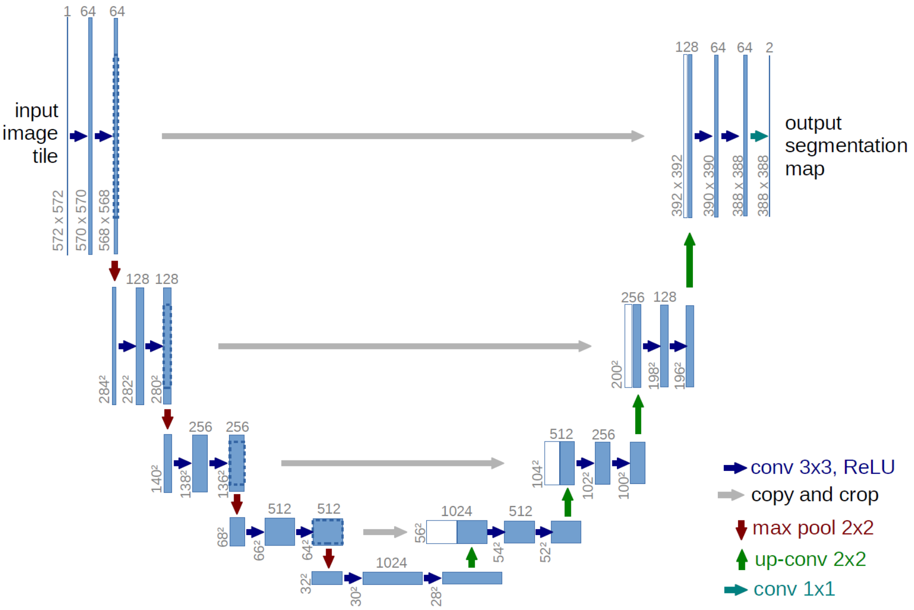


Fig. 2: UNet architecture (Ronneberger et al, 2015)

3 Design and Methods

3.1 YOLO

Data Preprocessing The first thing is to create text file of images. We need to create txt file for each png image in the same directory and with the same name (Fig. 3). For example for 000001-72.png, we create the corresponding 000001-72.txt. which contains 5 numbers: object class, x, y, width, height. Because we only frame the lefteye, the object class is 0. And the x,y are the center of rectangle. And the width, height represent the bounding box's width and height.

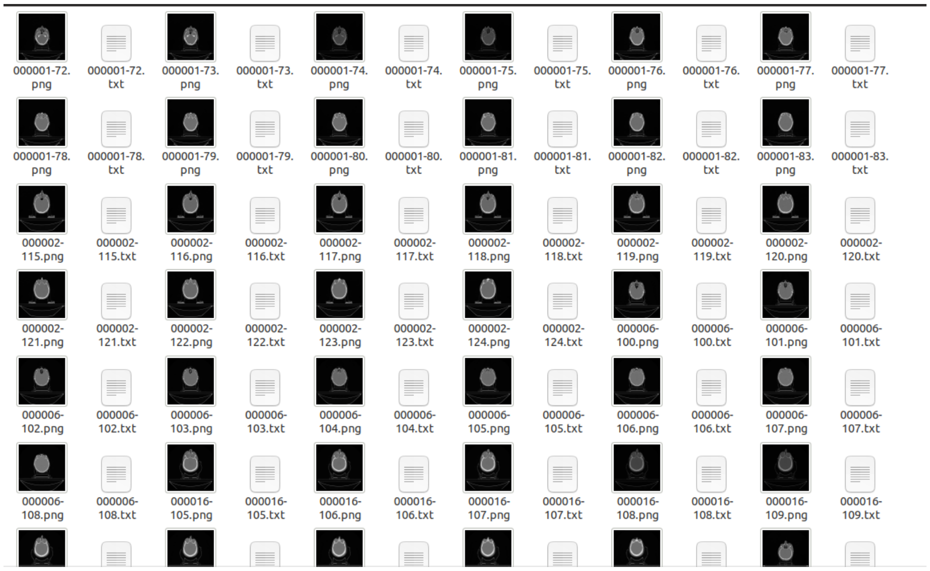
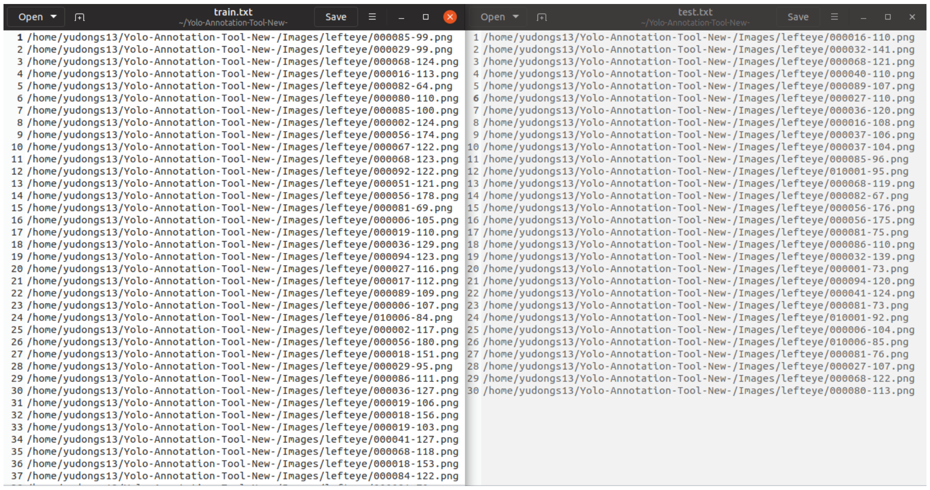


Fig. 3: YOLO Training Data

Yolo Training The images are divided into train set and test set. And then a few YOLO configure files are prepared. For instance, the batch size is set to 24, and filters is set to 30. Then the model is trained with following hardware (Fig. 4). In this project, a server with Intel Xeon E2699 and NVIDIA Tesla K80 GPU using Kepler architecture is used for all the training and testing.



```
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000085-99.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000029-99.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000068-124.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000016-113.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000082-64.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000080-110.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000085-100.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000002-124.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000056-174.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000067-122.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000068-123.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000092-122.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000051-121.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000056-178.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000081-69.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000006-105.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000019-110.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000036-129.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000094-123.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000027-116.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000017-112.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000089-109.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000006-107.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/010006-84.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000002-117.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000056-180.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000018-151.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000029-95.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000086-111.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000036-127.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000019-106.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000018-156.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000019-103.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000041-127.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000068-118.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000018-153.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000084-122.png
```

```
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000016-110.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000032-141.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000068-121.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000040-110.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000089-107.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000027-110.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000036-120.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000016-108.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000037-106.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000037-104.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000085-96.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/010001-95.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000068-119.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000082-67.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000056-176.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000056-175.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000081-75.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000086-110.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000032-139.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000001-73.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000094-120.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000041-124.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000081-73.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/010001-92.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000006-104.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/010006-85.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000081-76.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000027-107.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000068-122.png
/home/yudongs13/Yolo-Annotation-Tool-New-/Images/lefteye/000080-113.png
```

Fig. 4: YOLO Training and Testing Set

3.2 YOLO and UNet

A typical use case of UNet is CT image segmentation. From the original CT image (Fig. 5), certain aimed organs would be recognized automatically for medical experts. For example, eye and its nerve are determined as mask in Fig 6. It's obviously that eye nerve is much smaller than eye ball. Thus it would be more difficult to detect eye nerve since it cannot be distinguished from complicated background and noise.

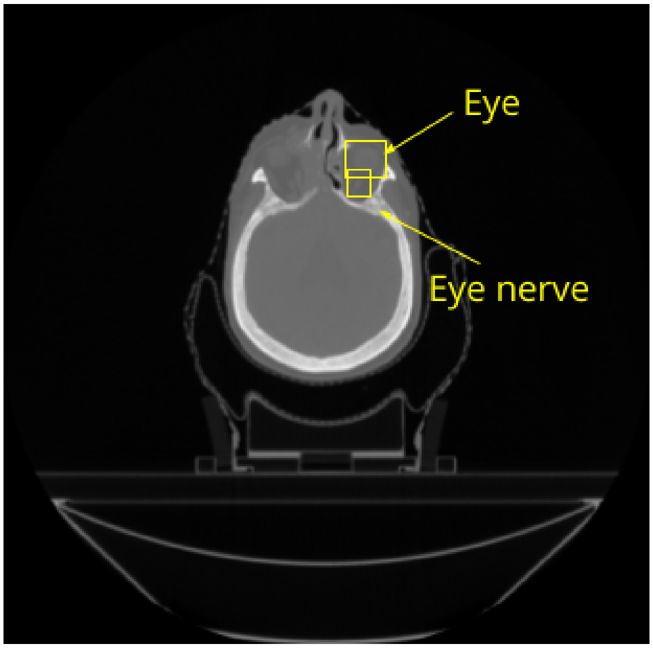
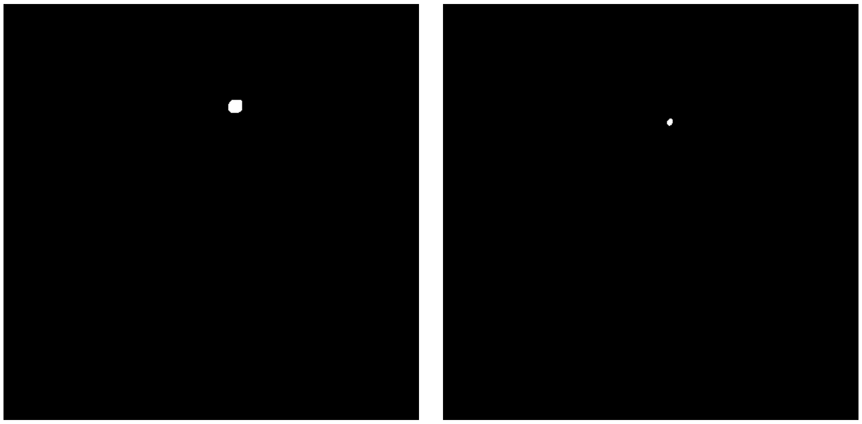


Fig. 5: Original CT image of human head



(a) Eye mask

(b) Eye nerve mask

Fig. 6: Masks for eye and nerve

To improve the prediction accuracy, only the small area containing the eye nerve should be preserved and all the rest of images should be wiped out to reduce distraction. Considering the eye nerve is right next to the eye ball and eye ball is larger and easier to be detected, we proposed to use YOLO to detect eye ball first and crop that area based on its prediction. As Fig. 7 shown, YOLO can detect the eye ball. Then if a slight larger area in the blue box is cropped around eye ball, it would contain the actual goal eye nerve with most of irrelevant background removed(Fig. 8).

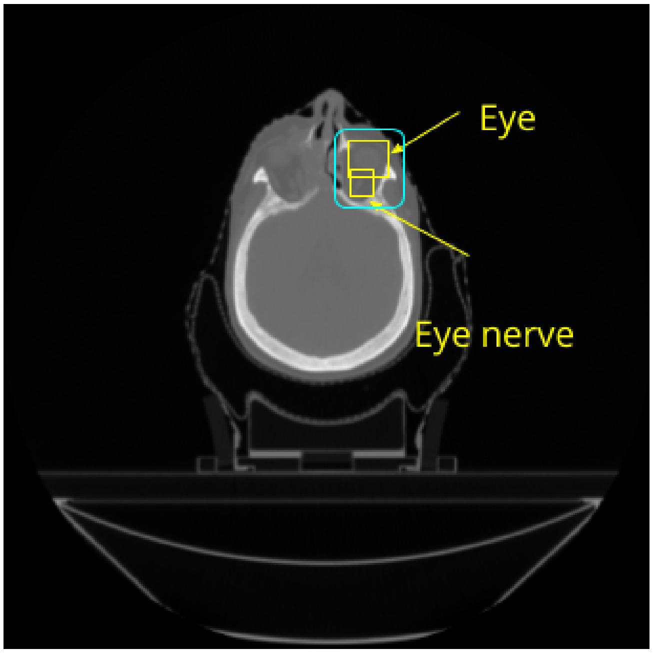
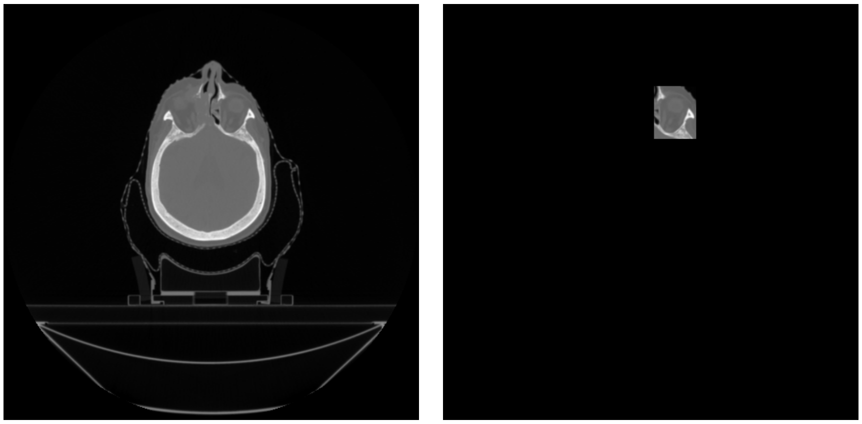


Fig. 7: The area surrounding eye and containing eye nerve



(a) Original CT image

(b) Cropped CT image

Fig. 8: Image preprocessing by YOLO

4 Results

4.1 YOLO

After the training process using our data set, we get a yolofinal.weights after 120000 iterations to represent our model. Now we can predict the lefteye in the original image with a bounding box, and the accuracy is 86%. (Fig. 9)

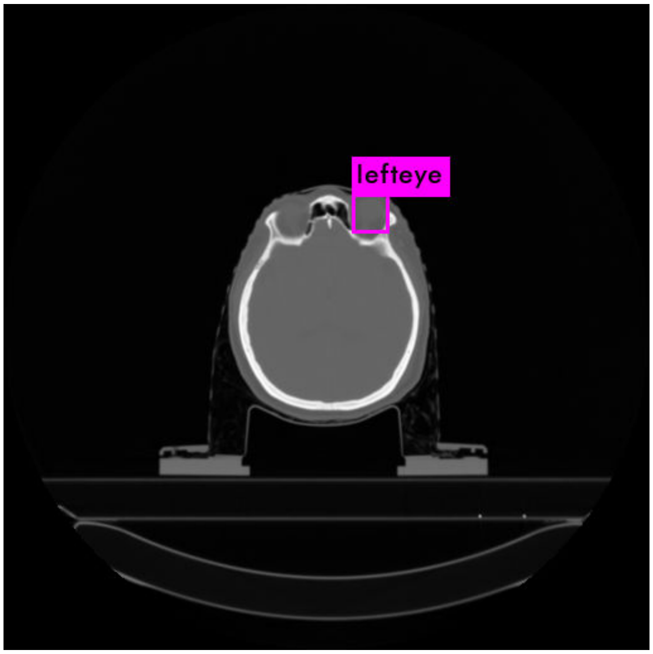
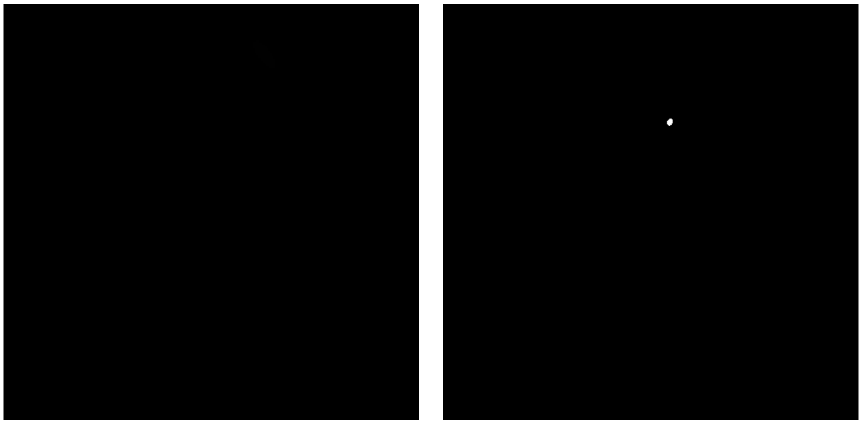


Fig. 9: Prediction using YOLO

4.2 UNet

The CT images containing eye nerves are used for UNet training directly. After tuning by MD Anderson cancer center, the accuracy is around 74.8% using a large data set containing about 1300 images.

In this project, due to limited resources 63 images with both left eye and its nerve were used as both training and testing sets. The performance is expected to very good since the training and testing sets are the same. However, only one false positive prediction is made. For the rest 62 images, the output is plain black image. Because of very small training set, the model is poorly established and cannot provide any meaningful prediction.



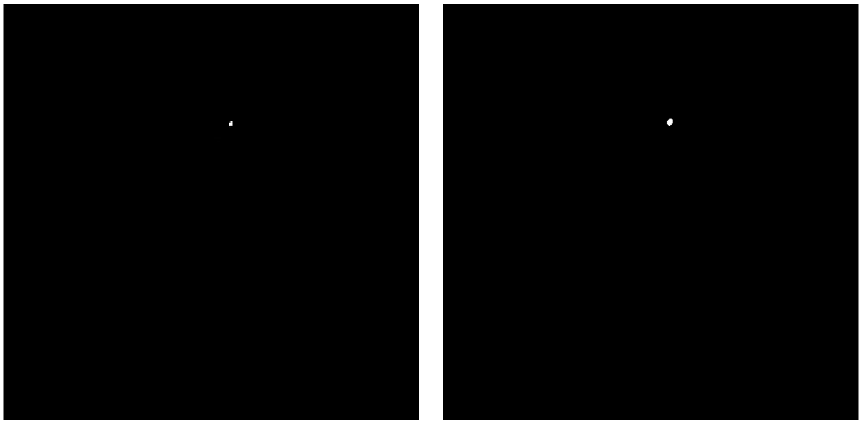
(a) Prediction (b) Ground truth

Fig. 10: Prediction evaluation using only UNet

4.3 YOLO and UNet

Before feeding the original CT images into UNet, they are fed into YOLO. The area around eyes are determined, which contains eye nerves and much less irrelevant background.

Eventually, we could obtain a much better prediction using processed images (Fig. 11).



(a) Prediction (b) Ground truth

Fig. 11: Prediction evaluation using YOLO and UNet

5 Conclusion

With a small data set, UNet itself cannot provide any meaningful prediction. Even with larger data set, UNet can only obtain about 74.8% accuracy due to tiny size of object. To solve this problem, we use YOLO to detect a much smaller area containing the object first and cover the rest of image. Yolo is able to detect eyes with 86% accuracy. With the preprocessed images in the same small data set, we significantly improved the prediction of UNet. The quantitative accuracy is still not high enough because any mismatching would be relected on the accuracy greatly and the size of data set affects the model a lot. However, it's proved that the combination of YOLO and UNet is a good approach to increase prediction accuracy under limited conditions. In future research, we plan to obtain the whole set of CT images for training to evaluate our approach better. Also attention map would be involved to improve the YOLO model training since YOLO's performance is the upper bound of UNet model in this approach.

References

1. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788
2. Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham