

# CLASSIFICATION OF CT SCAN IMAGES BASED ON ANATOMICAL PARTS & LESION LOCATION USING CONVOLUTIONAL NEURAL NETWORKS



Bhairavsingh Sunil Ghorpade, Akhil Sanjay Potdar, Uma Manognya Kaipa  
The University of Oklahoma

## Abstract

Medical imaging and CAD is one of the most challenging and important futuristic artificial intelligence work which has been carried out since the past few decades. A major milestone in this development is image detection in CT scan reports of medical imaging. As a part of this exploratory study we attempted to locate the lesion using comparatively less computational power on the DeepLesion dataset from PACs. Our unique approach divided the entire project into two phases, first phase involved building a model for the classification of the CT images into the different anatomical parts and the second phase focused on locating the lesion in each of these body parts. As a result we have achieved 95% classification accuracy and we were successful in locating the lesion. The paradigm of occurrence for lesion is detected as we go deeper into the convolutional layers.

## Introduction

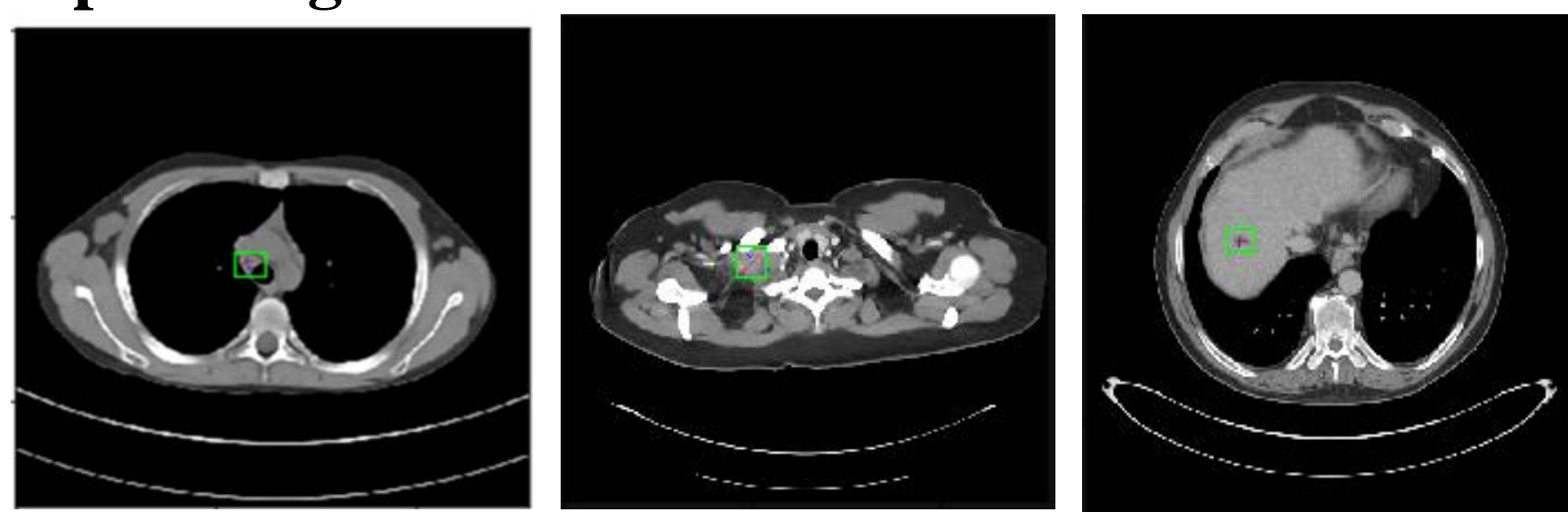
- Convolution neural networks have proven to be highly significant in the field of biomedical image processing and have provided extremely insightful predictions in models concerning object classification and detection.
- Our dataset, named DeepLesion, is composed of 32,735 lesions in 32,120 bookmarked CT slices from 10,594 studies of 4427 unique patients. We have used a subset of 9816 images containing the CT scans of all the 8 body parts for both classification and lesion detection.
- In our project, we have undertaken the challenge of locating the lesion without the use of a pretrained network.
- The use of CNNs to locate the lesion for different body parts is not just interesting but equally challenging since each of part has a different structure and the location of the lesions are random.

## Problems

**Problem Statement:** Locating lesion from given CT scan image without pretrained weights and with lesser computational force with fewer train data.

**Data Description:** Input data is a RGB image with size 512 x 512.

**Input image:** 9816 CT scans



**Output:** The output is the location of the lesion in the given image. We have defined grids within the image and the output shows the location of grid with lesion.

**Data Source:** Deep-Lesion dataset containing 32,120 CT scanned images from PACs of which 9816 images are annotated

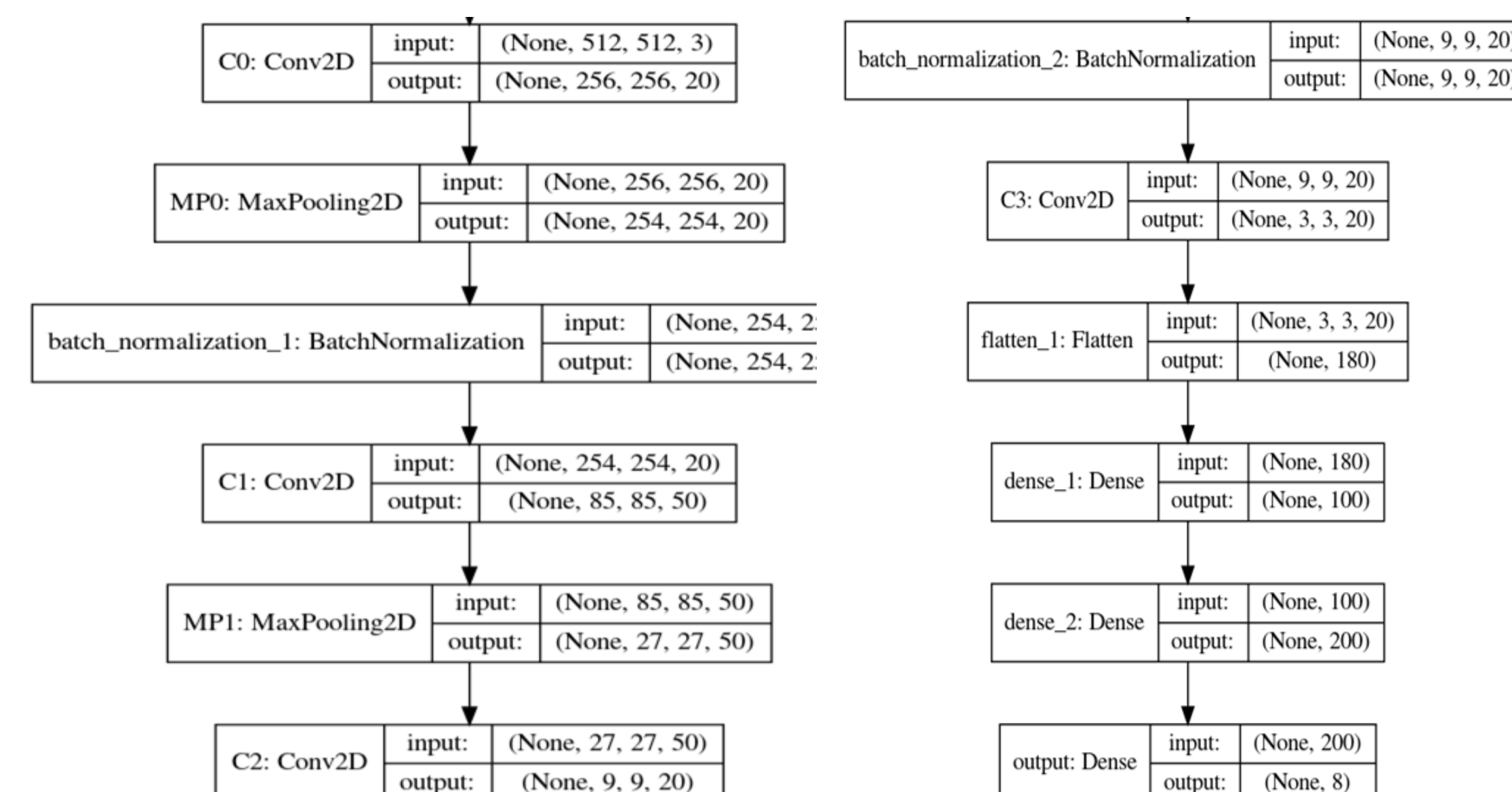
Body Parts	Lungs	Abdomen	Mediastinum	Liver	Pelvis	Soft Tissue	Kidney	Bone
Number of Images	2426	2166	1638	1318	869	677	490	232

## Phase 1: Classification of images into the anatomical parts.

- Modeling approach : 3 model architectures were created by each of the team members.
  - Model 1: Deep Network with High Parameters
  - Model 2: Deep Network with Low Parameters
  - Model 3: Shallow Network with High Parameters

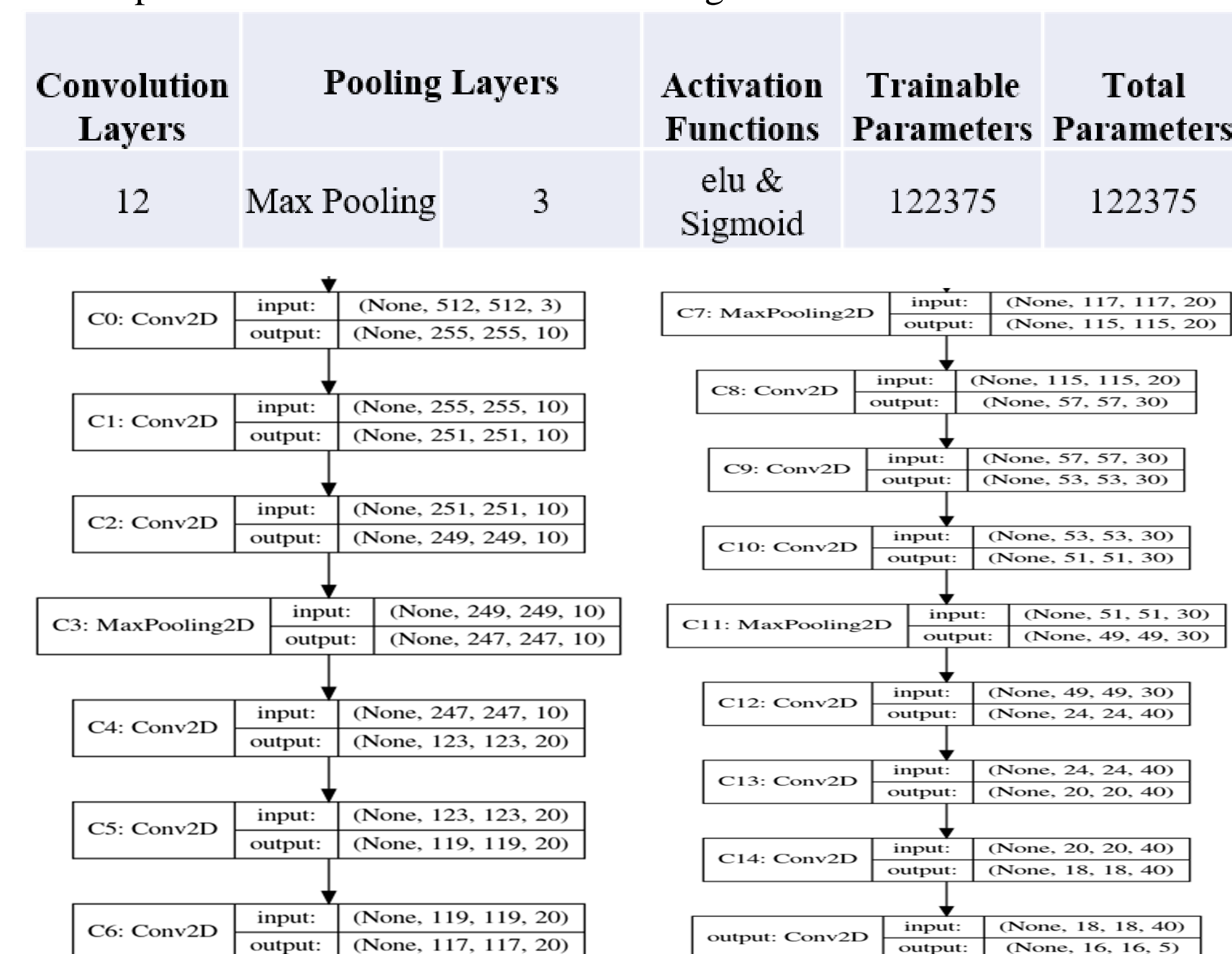
	Model 1	Model 2	Model 3
Convolution Layers	4	5	2
Pooling Layers	Max Pooling	Max -- Avg Pooling	Max Pooling
Batch Normalization Layers	2	2 -- 2	2
Dense Layers	3	3	2
Dropout Layers	0	0	1
Activation Functions	Relu & Sigmoid	Elu & Softmax	Relu & Sigmoid
Trainable Parameters	62684	28266	74008
Non-Trainable Parameters	526	38	0
Total Parameters	63210	28304	74008

- Outputs: Probabilities for each image amongst the 8 classes.
- After the comparisons of results of all three models, we selected Model 1 as the best classification model. The architecture of our best model is shown below.



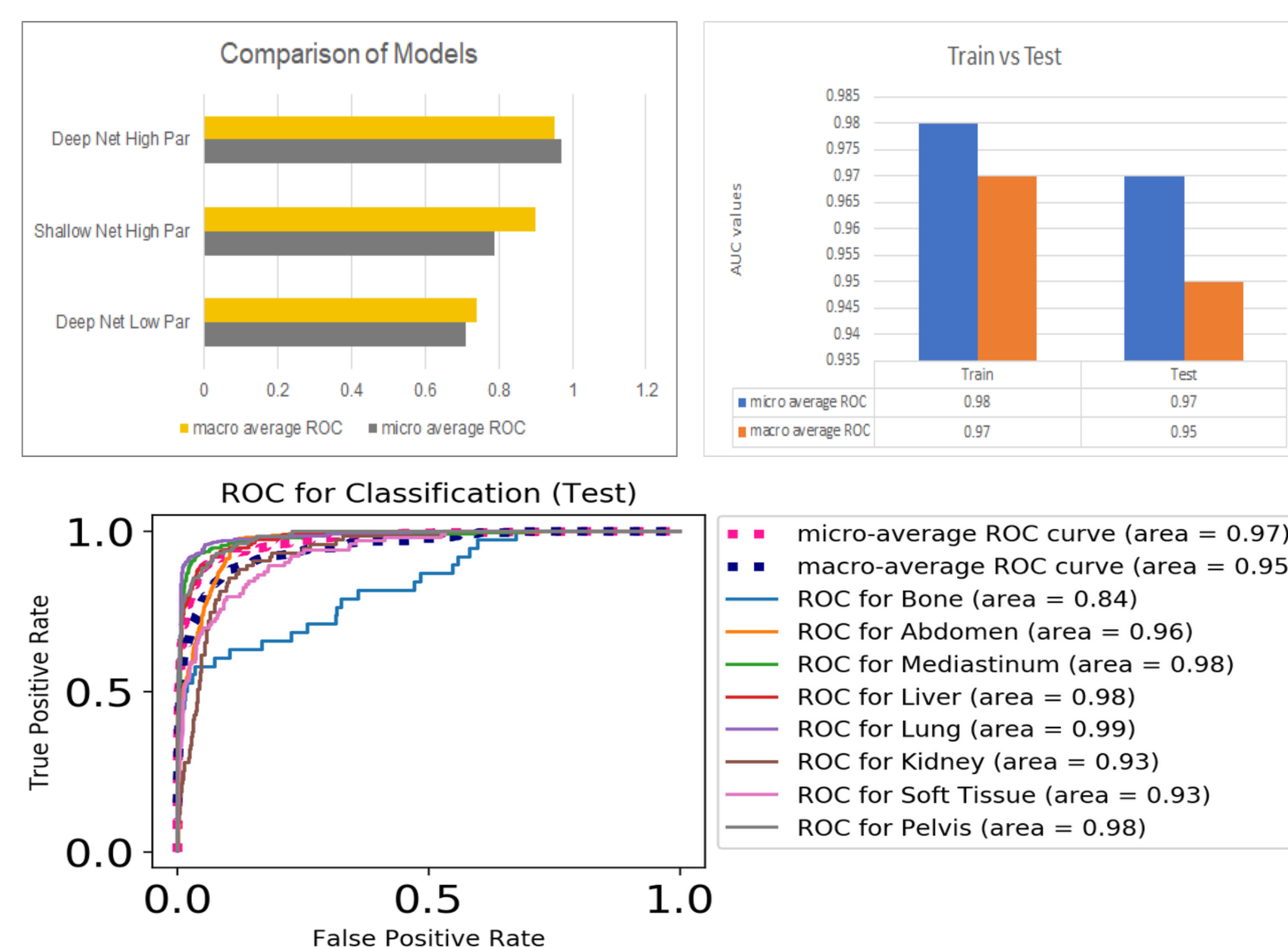
## Phase 2: Locating Lesion

- Classified images are used to train 8 models for lesion detection.
- These models are used to detect lesions within the same body part.
- Modeling approach: The model predicts the location of grid containing the lesion.
  - Initially image is divided into s x s grids (here s = 16).
  - After the division, we created output labels of size 16 x 16 x 5.
  - Each grid has five parameters, as follows.
    - P = probability of lesion in given grid.
    - x & y = location of center of lesion in grid.
    - h & w = height and width of lesion.
  - For our project, we are prediction only p and rest are not considered.
- Outputs: Probabilities of lesion in each grid.

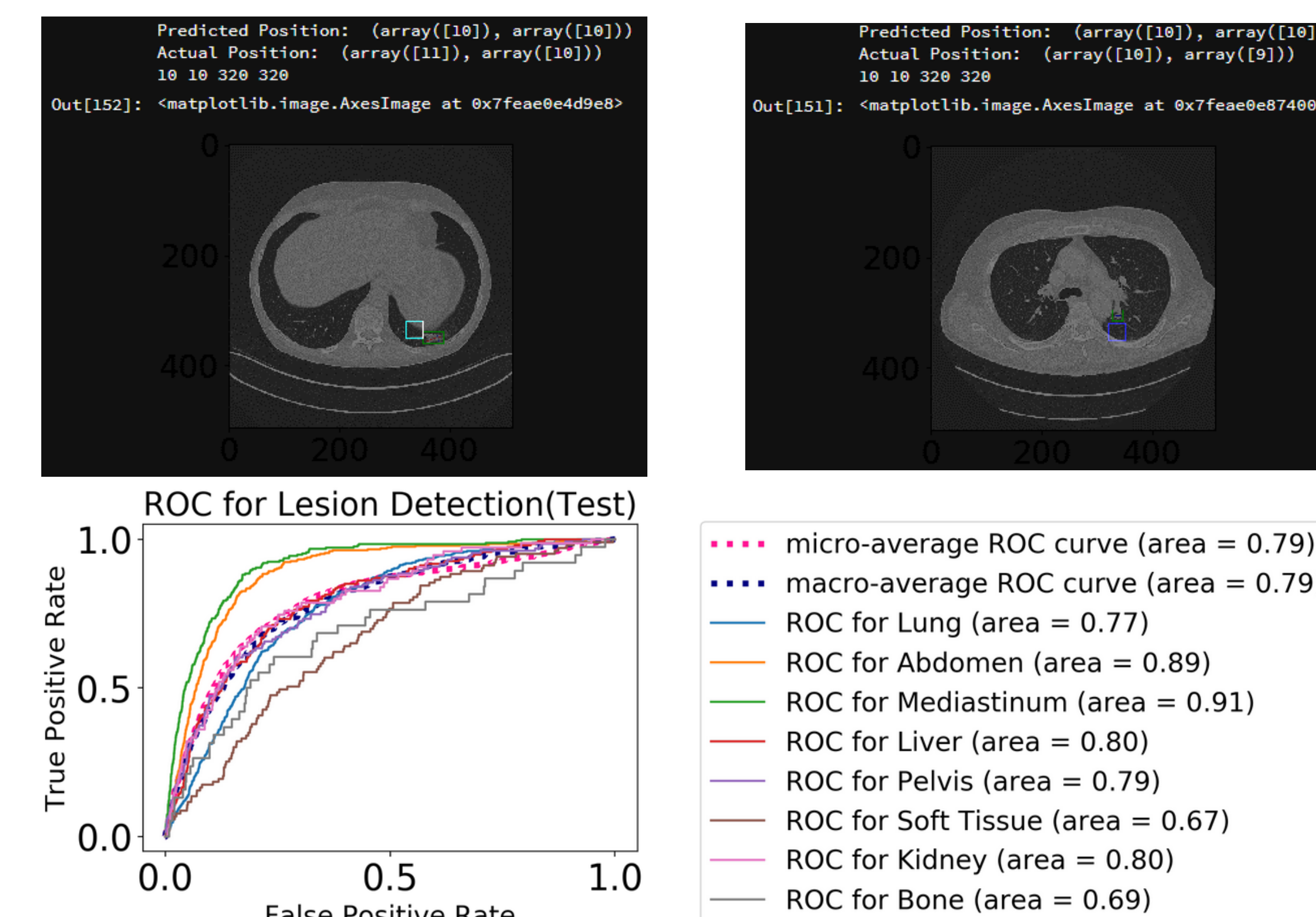


## Results

### Phase 1: Classification of images



### Phase 2: Locating Lesion



## Conclusion

Although this project is predominantly focused on locating the lesion, classification of the body parts provided a much needed vision for our network to learn the pattern of each body part and differentiate them from one another which helped in the progression towards our lesion detection model. The fundamental highlight of this project is successfully achieving lesion detection without the use of pretrained networks. In the future, we plan on improving our accuracy for locating lesion and predict the exact location with more precision including the midpoint and size of lesion. This could be accomplished by increasing our dataset with collection of more heterogeneous data from radiology reports. Further advancement of this project would be to develop a universal lesion detection framework using a unified detection system which would be beneficial for the medical society.

## Acknowledgements

We would like to express our gratefulness and gratitude to Dr. Dee H. Wu, Associate Professor, Department of Radiological Sciences, OUHSC and Dr. Andrew Fagg, Department of Computer Science, University of Oklahoma for their valuable and constructive suggestions during the planning and development of this research work. Their willingness to offer their time so generously has hugely contributed towards the success of this project.

## References

- “DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning,” KeYan, XiaosongWang, Le Lu, Ronald M. Summers, J. Med. Imag. 5(3), 036501 (2018)
- “A Lightweight YOLOv2: A Binarized CNN with A Parallel Support Vector Regression for an FPGA” by Hiroki Nakahara,Haruyoshi Yonekawa, Tomoya Fujiand Shimpei Sato.
- “Gentle guide on how YOLO Object Localization works with Keras”  
<https://www.dlology.com/blog/gentle-guide-on-how-yolo-object-localization-works-with-keras-part-2/>