Deep Learning based Needle Localization in CT Images for Biopsy Procedures: Classification and Segmentation Analysis

Kimaya Kulkarni

UID: 805528337

University of California, Los Angeles
(UCLA)

Abstract

This paper presents a study on needle-based CT image classification and segmentation for biopsy procedures. The aim is to enhance the accuracy and efficiency of needle localization within tumors, which is crucial for precise tissue sampling. Various deep learning models, including CNN, DeepCNN, AlexNet, ResNet, Multihead Attention, and Vision Transformer, are evaluated for classification. The best-performing model, ModAlexNet1, achieves a test accuracy of 0.8033, surpassing more complex architectures. However, segmentation results using a pretrained Unet model are unsatisfactory, and limited resources hinder supervised learning convergence. Despite this, the study contributes by demonstrating the potential of deep learning techniques in accurately identifying needles in CT scans, achieving a classification accuracy of 80%. The development of a method to convert annotated images into binary masks for segmentation provides valuable ground truth for future research.

1 Introduction

A biopsy is a procedure in which a tissue sample is taken out of your body for testing. Biopsy is a reliable way of determining if a tumour is benign or malignant. However, parts of tumour may be benign while some parts may be malignant. Hence, it is necessary to know the precise location from which the tissue sample was collected inside the tumour.

CT scans are often captured during a biopsy to guide a needle to a tumour tissue. These scans provide detailed images of the tumor and its surrounding tissues, allowing healthcare providers to precisely locate the tumor and target the biopsy needle accordingly. These can also be referenced later to see from which part of the tumour was tissue collected by observing the needle position. This can help ensure that the biopsy sample is taken from the correct location within the tumor and improve the accuracy of the diagnosis.

The process of manually searching through CT images for location of needle can be error prone and tedious considering the number of images captured. Deep learning techniques can automatically locate and segment needles in CT scans which may be more accurate and faster. These techniques can enable more efficient detection, classification and segmentation of needles in large volumes of data.

The goal of this work is to perform Needle based CT images classification and segmentation. In this, I make the following contributions:
Contribution A: Achieved 80% test accuracy for classification model.
Contribution B: Provide a way to convert images into masks as ground truth for segmentation task.
Contribution C: Study of several models to determine the disadvantages of each model and to choose the best model for the task.
The paper is structured as follows. Section 2 talks about methods, in which we see the pipeline of dataset details, data preprocessing, description of models used, statistical evaluation goes over how pipelines for classification and segmentation are structures. Section 3 goes over results and analysis which is further divided into classification and segmentation. In section 4, we conclude the paper with all the insights we have gained from the research.

2 Methods

2.1 Data Description and Pre-processing

The data used in this project consists of 632 2D CT slices in '.jpg' format, with each slice having a shape of 1*256*256. The data was obtained from a hospital's database and was collected from patients who underwent biopsies for suspected lung cancer.

For the classification task, we were provided with a labels.csv file that contains binary labels 'Yes' or 'No' indicating the presence or absence of needles in each CT slice. This labeling was done manually by trained medical professionals who analyzed the CT images and identified the presence or absence of needles. For the classification task, 'Yes' and 'No' were converted to 1s and 0s to make the code easier.

441 images in the dataset contain a needle, 191 images are in class 0, which do not contain a needle. For pretrained models, a 3,256,256 image is generated by giving same values of grayscale for RGB channels.

For the segmentation task, 576 images with red dots and start and end point of needle were available. Hence only 576 images were used for training and testing. The issue with these images was that as all images were saved using paint, the images were converted from grayscale to RGB format. Hence, the red channel contained more information that just two red dots. This could not be used as binary mask. Hence, it was necessary to convert these annotated images into binary masks. The following procedure was used for the same.

The annotated images were used to get the binary mask. The RGB images were first converted into HSV space as it is easier to isolate colors in this space. Then, lower and upper threshold was given to red color. A binary mask was created using this threshold and then morphological operations of opening and closing were performed on the image to get rid of any pixel noise. Now, for the remaining regions a centroid was found. Considering each image only had two red dots, we had two centroids which were joined to each other to create the final binary mask which would be used as ground truth for segmentation.

In segmentation task, original images were used as input, whereas binary masks were used as desired output for training. Examples of binary masks can be seen in figure 1 and 2.

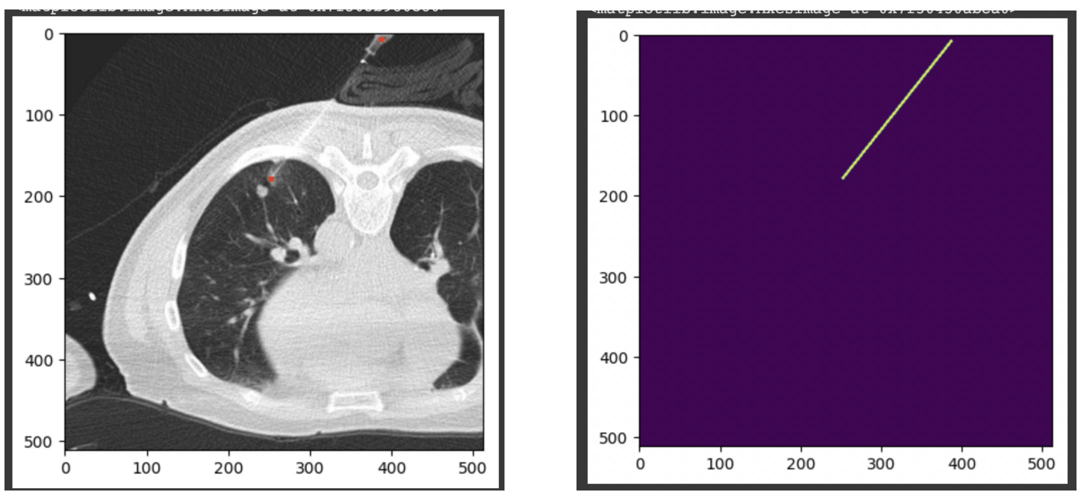

Figure 1: Example of original image vs binary mask generated(with needle)

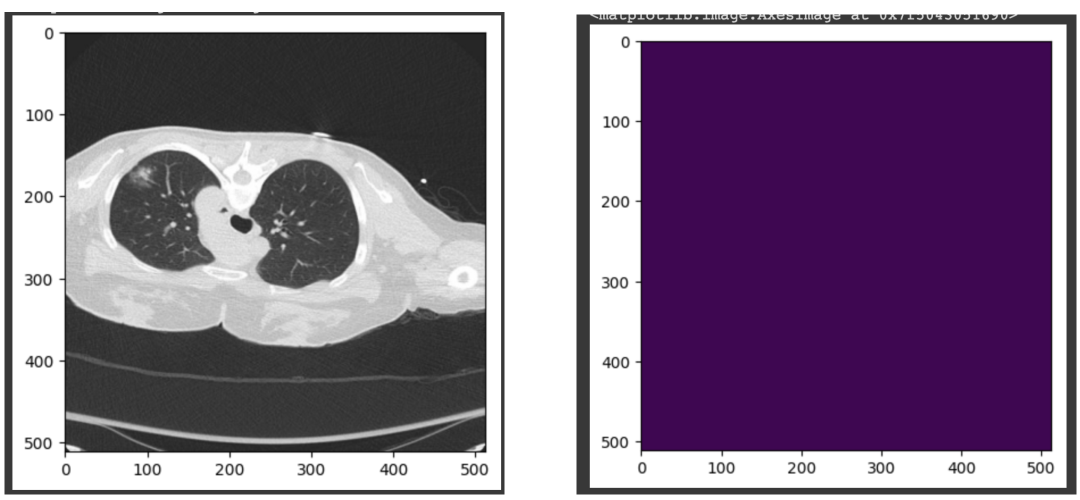

Figure 2: Example of original image vs binary mask generated(without needle)

2.2 Model Description

Throughout this study several CNN models were tried. The descriptions of these models is given briefly in the remaining section.

2.2.1 CNN

Two basic CNN architectures with varying number of layers and varying input, hidden units of convolutional layers were used. The basic architecture was three convolutional layers with relu activation followed by max pool layers. And finally multiple three fully connected layers. The convolutional layers progressively extract features from the input image, reducing spatial dimensions and increasing the number of channels. The fully connected layers further process the extracted features to perform the classification task. The architecture uses ReLU activation functions for non-linearity and max pooling operations for downsampling.

2.2.2 Alexnet

It consists of eight layers, including five convolutional layers and three fully connected layers, with a total of 60 million parameters. AlexNet introduced several key innovations, such as the use of ReLU activation functions, overlapping pooling, dropout regularization, and GPU acceleration. The original architecture as well as modified architecture by reducing the number of layers was used for experimentation.[2]

2.2.3 Resnet

ResNet-50 consists of 50 layers, including convolutional layers, pooling layers, fully connected layers, and skip connections. These skip connections allow for the direct flow of information from earlier layers to later layers, enabling easier training and better gradient flow. Resnet pretrained on imagenet with anothr fully connected layer added at the end was used. Also, resnet without any pretraining was tested. [1]

2.2.4 CNN + Multihead Attention

The network incorporates multi-head attention for image classification. It starts with a convolutional layer (Conv2d) to extract features from the input image. The attention mechanism (MultiheadAttention) is then applied, allowing the network to focus on important regions of the image. The output of the attention is reshaped and passed through a fully connected layer (Linear) with a ReLU activation function. Finally, the output layer predicts the class probabilities. This network architecture leverages the power of attention mechanisms to capture informative image features and has the potential to achieve good performance in image classification tasks. [6]

2.2.5 Transformer

The network incorporates a transformer module for feature extraction and classification. It starts with a series of convolutional layers (Conv2d) followed by max pooling operations to extract hierarchical features from the input image. The fully connected layers (Linear) further process the extracted features. The addition of the transformer module allows the network to capture global dependencies and long-range interactions within the feature representations. The transformer applies self-attention mechanisms across the feature dimensions, enhancing the network's ability to model complex relationships in the data. This network architecture combines the strengths of both convolutional and transformer models for improved image classification performance. [6]

2.3 Statistical Evaluation

2.3.1 Classification

After importing necessary libraries, a custom dataset class, "MedDataset new", for loading and processing medical image data is defined. It reads image file paths and corresponding labels from an annotations file. The labels are transformed to binary values (0 or 1). The dataset class provides the len and getitem methods required by PyTorch's Dataset class. The getitem method loads an image, resizes it to a fixed size of 128x128 pixels, applies optional transformations to the image and label, and returns the processed image and label as a tuple.

Following this, a train-test 80 -20 split for a medical image dataset, where the dataset is loaded using the "MedDataset new" class and divided into train and test sets using a specified ratio, and then creates data loaders for the train and test sets with a batch size of 64 and shuffling the data.

The code includes all model architectures.Furthermore, training and testing is implemented on a given dataset for a specified number of epochs. It uses the Adam optimizer and the CrossEntropyLoss as the loss function. During each epoch, it iterates over the training data in batches, computes the predictions, loss, and accuracy, and performs backpropagation to update the model's parameters. The training loss and accuracy are tracked for each epoch. After each epoch, the code evaluates the model on the test data to compute the test loss and accuracy. The progress is printed during training, including the epoch number, training and test loss, accuracy, and elapsed time.

The test accuracy is tracked to see which model parameters give the best results. Accuracy is a commonly used metric in classification problems. It is defined as the ratio of correctly classified samples to the total number of samples in the dataset.

2.3.2 Segmentation

As some image anotations were not available, the pipeline started by the code that reads image labels from a CSV file, checks for the existence of corresponding image and label files in specific directories, and removes the entries from the CSV file for which either the image or label file is missing. The updated DataFrame is then saved back to the CSV file.

We have images which have two red dots for start and end of the needle. However, the red channel of image also contains other information. So to create a binary mask we have to write a function which reads an image file, converts it to the HSV color space, applies color thresholding to extract regions of interest, performs morphological operations to enhance the regions, applies connected component labeling to identify separate objects, filters the objects based on their area, and finally generates a binary mask representing a line connecting the two red dots. This will be our label image.

Dataloader will basically take this label image and original image which will be used to train model. Then we will use a similar code to train and test on the dataset.

3 Results and Discussion

3.1 Classification

The table 1 presents the performance results for various models used for needle classification in CT images. The models include CNN, DeepCNN, AlexNet, Modified AlexNet, Modified AlexNet2, pretrained ResNet50, ResNet50, Attention Network, and Vision Transformer. The test accuracy and training time for each model are provided. The best performing model in terms of test accuracy is ModAlexNet1 with an accuracy of 0.8033, while the model with the fastest training time is Deep CNN with a time of 2.8549 seconds.

Model	Test Accuracy	Time
CNN	0.7958	8.1570
DeepCNN	0.7561	2.8549
ModAlexNet1	0.8033	356.4940
ModAlexNet2	0.7872	629.7521
AlexNet	0.7638	514.5660
Resnet50(pretrained)	0.7795	342.834
Resenet50	0.7715	302.949
Multihead Attention	0.6610	198.1117
Vision Transformet	0.7013	173.4780

Table 1: Classification Test Accuracy

The results give some interesting insights into inner workings of deep learning methods. Number of layers increases significantly as we go from CNN to Resnet50.

While checking the train and test losses of CNN and DeepCNN, we notice the exact epoch where model starts overfitting(i.e. the train loss keeps decreasing however the test loss decreases and then starts increasing), this is the point whenre we get the best model weights and inturn the best test accuracy values.

The pretrained Resnet50 and resnet50 give similar performance, hence we can say that for transfer learning from imagenet trained dataset to medical images training, more preprocessing is needed on images to see clear improvements due to transfer learning.

The difference in performance of Alexnet and modified alexnet1-2 show that fine tuning the model for a specified task can enhance performance of these widely used CNNs.

The logic behind using attention and transformer was that it would be able to focus on specific regions of interest and help improve accuracy of the models. However,transformer and attention models exhibit low accuracy. The Attention Network and Vision Transformer models achieved test accuracies of 0.6610 and 0.7013, respectively. These models may require further refinement or adjustments to better capture the important features for needle classification. The failure of the Multihead Attention and Vision Transformer models in achieving high performance can be attributed to their complexity, limited dataset size, and sensitivity to hyperparameter settings, which hindered their ability to effectively learn and generalize from the available data.

3.2 Segmentation

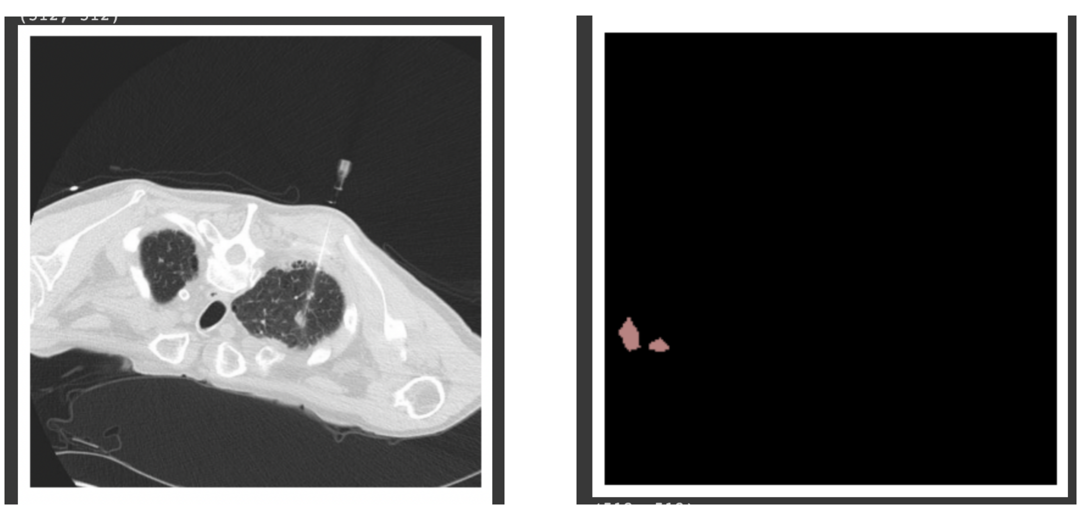

Figure 3: Original Image vs Pretrained Unet Segmented Image

The pretrained Unet[5] model, although commonly used for segmentation tasks, failed to classify needles accurately in this particular scenario. Instead, it mistakenly identified random regions, which might be relevant for tasks like tumor detection but were irrelevant for needle segmentation. This suggests that the pretrained model's learned representations did not align well with the specific features required for needle identification.

Additionally, supervised learning, which could have been an alternative approach, faced limitations due to restricted RAM, preventing the model from training to convergence. The model was only able to run for a few epochs and these were the results for those. As we can see the images, in few epochs at least the model is not able to identify and segment out the needle.

This highlights the importance of carefully selecting and adapting models to the specific requirements of the task and considering resource constraints during model development.

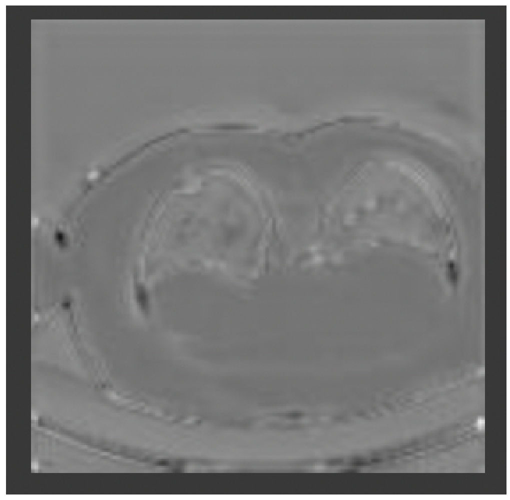

Figure 4: Supervised learning output after a few epochs

In future, other segmentation techniques[4] [3] could be used for this task.

4 Conclusion

In conclusion, this paper presents a study on needle-based CT image classification and segmentation for biopsy procedures. The objective was to improve the accuracy and efficiency of needle localization within tumors, which is crucial for determining the precise location from which tissue samples are collected.

For the classification task, various models including CNN, DeepCNN, AlexNet, ResNet, Multihead Attention, and Vision Transformer were evaluated. The best performing model was found to be ModAlexNet1, achieving a test accuracy of 0.8033. Interestingly, this model outperformed more complex architectures, highlighting the importance of model selection and customization for specific tasks.

For the segmentation task, the pretrained Unet model failed to accurately identify and segment needles, while limited resources prevented supervised learning from converging. This indicates the need for careful model adaptation and resource considerations when tackling segmentation challenges.

Overall, this work makes significant contributions in the field of needle-based CT image analysis. The achieved classification accuracy of 80% demonstrates the potential of deep learning techniques in accurately identifying needles in CT scans. The development of a method to convert annotated images into binary masks for segmentation provides a valuable ground truth for future research. Additionally, the study compares multiple models, identifying their advantages and disadvantages, helping guide future investigations.

Moving forward, further exploration can be done to improve segmentation performance, considering alternative architectures and data augmentation techniques. Additionally, the integration of physics-based methods with deep learning approaches can be explored to enhance the accuracy and efficiency of needle localization. This work lays the foundation for future research in this domain, contributing to advancements in the field of medical image analysis and improving the quality of biopsy procedures.

Acknowledgement

I express my sincere appreciation to Prof. William Speier and Prof. Corey Arnold for their exceptional guidance and unwavering support throughout this course. Their expertise and dedication greatly enriched my learning experience. I am grateful for the opportunity to embark on this challenging yet immensely rewarding project under their guidance. Their encouragement, feedback, and suggestions shaped my research. I also thank my fellow students for their active participation in annotating the images, ensuring the accuracy of our dataset. I am truly grateful for the contributions of all involved in completing this paper.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.