BREAST CANCER SEGMENTATION ON NUCLS DATASET: PROJECT REPORT

Mustafa Umut BÖLÜK
MSc. Data Science
Koç University
mboluk15@ku.edu.tr

Serhan TÜRKAN
MSc. Data Science
Koç University
sturkan20@ku.edu.tr

***Abstract*— The evolution of Deep Learning technology has greatly assisted Semantic Segmentation. Image segmentation has been a benefit to the medical profession in both application and research fields, allowing for the detection and identification of abnormal tissue sections. In this study we trained different architectures and hyper-parameters to achieve highest accuracy for segmenting cancer cell in breast histopathology images.**

I. INTRODUCTION

Breast cancer is one of the most prominent causes for women's deaths, and the cases are increasing day by day especially in the last two decades. According to the World Health Organization (WHO), 2.3 million women worldwide were diagnosed with breast cancer in 2020, with 685,000 deaths. As of the end of 2020, there were 7.8 million women alive who were diagnosed with breast cancer in the past 5 years which makes this cancer world's most prevalent cancer [1].

Traditionally, the tissue sections are observed by the histopathologists' naked eyes directly, and the visual information is analyzed based on the prior medical knowledge manually. However, due to the complexity and diversity of histopathological images, this manual analysis can take much time. At the same time, the objectivity of this manual analyzing process is unstable [2].

Nuclei segmentation, on the other hand, is a complex task which includes tissue type, staining variations, and cell type, and all express various visual characteristics, making it difficult to develop conventional image segmentation algorithms that perform well in all these different cases [3]. On the other hand, deep learning algorithms have been used recently with great success to complex segmentation tasks in biology.

In this project, we tried different deep learning architectures and hyperparameters to obtain best results on segmentation of nuclei cells. The remaining sections of this paper are provided as follows. Section 2 presents dataset. Section 3 demonstrates the methods used to conduct this project. Section 4 shows experimental results and Section 6 highlights the conclusion.

II. DATASET

In this project we used NuCLS: A scalable crowdsourcing, dataset. The NuCLS dataset contains over 220,000 labeled nuclei from breast cancer images from The Cancer Genome Atlas (TCGA) [4]. This is a large dataset with high quality annotations, which were created by non-pathologists, and corrected by study coordinators under the supervision of a pathologist. There are 1744 images and mask with 59.485 nuclei and 19680 boundaries.

III. METHODOLOGY

For breast cancer segmentation, we tried several different models and approaches to get good results with our chosen dataset. Some of these methods were the ones we discussed in our lectures and others were from the papers we found on the internet that were modified for a specific task. In this part of our report, we will discuss the methods and models we used in our project and how we implemented them.

FCN-ResNet

The first approach we tried to use was basic segmentation models called FCN-ResNet (Fully Convolutional Networks-ResNET). We chose these models because they are one of the basic semantic segmentation models that are used in object segmentation tasks. For the implementation of the model, we followed the Homework 3's implementation of AlexNet for classification tasks and changed the code to fit a segmentation model.

As for the implementation process, we first started with creating a segmentation class for our NuCLS dataset. Unlike our homework, we used masks of the images that were given in our dataset instead of the labels. Because of this difference, we also changed our data loader to fit the new problem. For the model, we used the pretrained version of FCN-ResNet that is provided by the torchvision library and used transfer learning, similar to our homework, to use the pretrained model. Then, for the trained model, we used an Adam optimizer and F1 score criteria for training them, where we had 4 output channels corresponding to the number of classes we had.

We used two different FCN-ResNet models called FCN-ResNet50 and FCN-ResNet101. The difference between these two models are the backbones they are using. FCN-ResNet50 model uses ResNet-50 as its backbone and FCN-ResNet101 model uses ResNet-101 as its backbone. When we trained these two models, we trained them using a small partition of our dataset. The reason for this was that, when we were working on the project, we observed that even when we used the whole dataset, the results we were getting did not change much and because there was a time constraint for the project, we decided to use a small partition of the dataset.

DeepLabv3-ResNet

The second approach we tried was to use segmentation models called DeepLabv3-ResNet. It is similar to FCN-ResNet model in the sense that they are both using ResNet models as a backbone for the model. On the other hand, when we compare the IoU scores of these two base models, we see that the DeepLabv3 approach scores better than the FCN approach. For the reason that our scores were not good for the first two models, F1 score of 0.58, we decided to try this base model as our second approach for the breast cancer segmentation problem.

For the implementation process of the approach, we followed the same steps with the previous approach. As we mentioned earlier, the approaches we used for these two models are using pretrained models from the torchvision library and implementing the model by following our homework 3 implementation. We again changed the labels with masks and used transfer learning to learn our dataset.

The specific models that we used in this approach were DeepLabv3-ResNet50 and DeepLabv3-ResNet101 models that used ResNet-50 and ResNet-101 for the backbone of the model respectively. Compared to the previous models, this approach was faster and more accurate. However, ultimately it was not accurate enough in the end as a segmentation model when we compare the results of our model with the ground truth. Hence, we try to use U-Net, a Dense Prediction algorithm we saw in the class.

U-Net

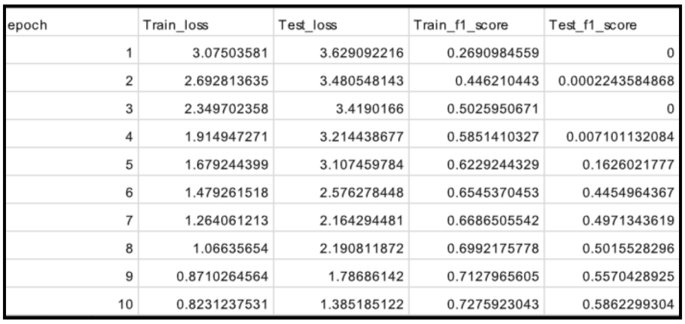
The last approach we tried was U-net architecture in Keras API. U-net is a convolutional neural network (CNN) that was developed for biomedical image tasks such as cell detection and shape measurements [5]. This time we tried the binary classification approach to solve this problem. For this, we converted the mask to 2D matrix that contains only two values zero and one. For more accuracy we used all dataset. We divided the dataset into two parts as training (1308 images) and test data (436 images). But this time we have cropped frames that cause noise from images and masks to avoid false results and resized them to 256x256 pixels. Since we approached this problem as a binary classification, we used sigmoid activation function in the output layer. We also used binary cross entropy and dice loss function. We trained the network using nuclei cell images as the input and binary masks as the target.

The number of trainable parameters was 8,648,689 and after we run the model output was a 2D mask that each pixel in the predicted mask denotes likelihood that pixel is a part of a cell. To convert this probability to binary we thresholded the mask values from 0.5. We used a binary accuracy function to calculate the accuracy and the highest score was 82% and the average IoU was 0.76.

IV. Experimental Results

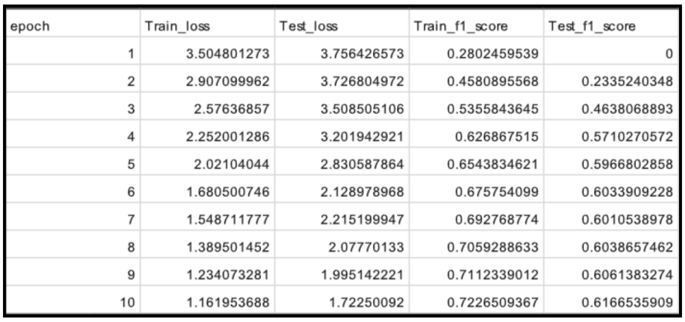
For the experimental results, we have both qualitative and quantitative results of our models. Quantitative result data is for F1 score for FCN-ResNet, DeepLabv3-ResNet models and IoU for U-net model. To explain these scores, F1 score is a measure that shows the accuracy of the model and its between 0 and 1, where 0 is the worst score and 1 is the best score. IoU score is also between 0 and 1, where scores higher than 0.5 is considered good, and it represents the accuracy of the bounding box. We also calculated the loss for every epoch of the models and presented it in a table.

Below tables show the losses and F1 scores for different models in both training and test steps.



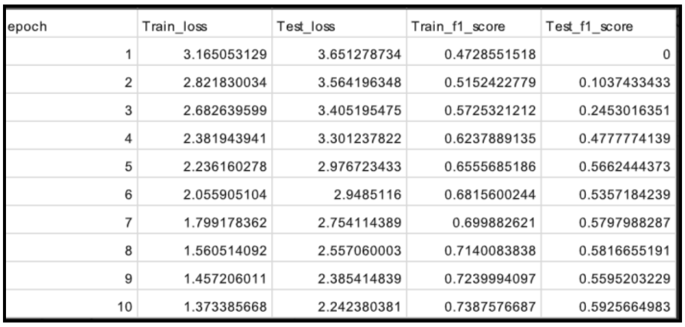
epoch	Train_loss	Test_loss	Train_f1_score	Test_f1_score
1	3.07503581	3.629092216	0.2690984559	0
2	2.692813635	3.480548143	0.446210443	0.0002243584868
3	2.349702358	3.4190166	0.5025950671	0
4	1.914947271	3.214438677	0.5851410327	0.007101132084
5	1.679244399	3.107459784	0.6229244329	0.1626021777
6	1.479261518	2.576278448	0.6545370453	0.4454964367
7	1.264061213	2.164294481	0.6686505542	0.4971343619
8	1.06635654	2.190811872	0.6992175778	0.5015528296
9	0.8710264564	1.78686142	0.7127965605	0.5570428925
10	0.8231237531	1.385185122	0.7275923043	0.5862299304

TABLE I. Loss and F1 scores for FCN-ResNet50 model



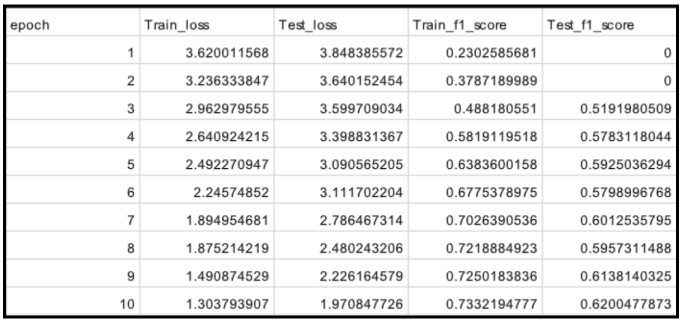
epoch	Train_loss	Test_loss	Train_f1_score	Test_f1_score
1	3.504801273	3.756426573	0.2802459539	0
2	2.907099962	3.726804972	0.4580895568	0.2335240348
3	2.57636857	3.508505106	0.5355843645	0.4638068893
4	2.252001286	3.201942921	0.626867515	0.5710270572
5	2.02104044	2.830587864	0.6543834621	0.5966802858
6	1.680500746	2.128978968	0.675754099	0.6033909228
7	1.548711777	2.215199947	0.692768774	0.6010538978
8	1.389501452	2.07770133	0.7059288633	0.6038657462
9	1.234073281	1.995142221	0.7112339012	0.6061383274
10	1.161953688	1.72250092	0.7226509367	0.6166535909

TABLE II. Loss and F1 scores for FCN-ResNet101 model



epoch	Train_loss	Test_loss	Train_f1_score	Test_f1_score
1	3.165053129	3.651278734	0.4728551518	0
2	2.821830034	3.564196348	0.5152422779	0.1037433433
3	2.682639599	3.405195475	0.5725321212	0.2453016351
4	2.381943941	3.301237822	0.6237889135	0.4777774139
5	2.236160278	2.976723433	0.6555685186	0.5662444373
6	2.055905104	2.9485116	0.6815600244	0.5357184239
7	1.799178362	2.754114389	0.699882621	0.5797988287
8	1.560514092	2.557060003	0.7140083838	0.5816655191
9	1.457206011	2.385414839	0.7239994097	0.5595203229
10	1.373385668	2.242380381	0.7387576687	0.5925664983

TABLE III. Loss and F1 scores for DeepLabv3-ResNet50



epoch	Train_loss	Test_loss	Train_f1_score	Test_f1_score
1	3.620011568	3.848385572	0.2302585681	0
2	3.236333847	3.640152454	0.3787189989	0
3	2.962979555	3.599709034	0.488180551	0.5191980509
4	2.640924215	3.398831367	0.5819119518	0.5783118044
5	2.492270947	3.090565205	0.6383600158	0.5925036294
6	2.24574852	3.111702204	0.6775378975	0.5798996768
7	1.894954681	2.786467314	0.7026390536	0.6012535795
8	1.875214219	2.480243206	0.7218884923	0.5957311488
9	1.490874529	2.226164579	0.7250183836	0.6138140325
10	1.303793907	1.970847726	0.7332194777	0.6200477873

TABLE IV. Styles Loss and F1 scores for FCN-ResNet50 model

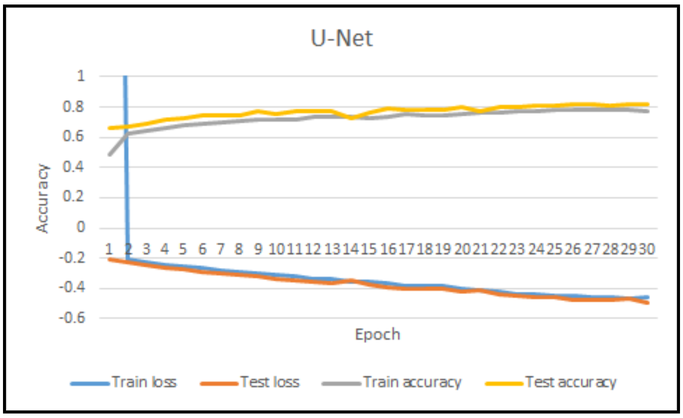


TABLE V. MODEL LOSSES AND ACCURACIES FOR U-NET

Here from the tables and quantitative results for these 5 models we can have some observations. We can see that when the backbone is changed, accuracy increases, meaning ResNet50 is worse than ResNet101 when used as the backbone for the model. We can also observe that as we change the base model to improve the score, accuracy of the model increases. This shows that the performance order goes from FCN-ResNet, DeepLabv3-ResNet and U-Net, worse to best.

We also have qualitative results that show that even though our models' accuracy is close to 0.8 in the end, our results are still not good enough compared to the ground-truth. Below figures show the qualitative results of the models.

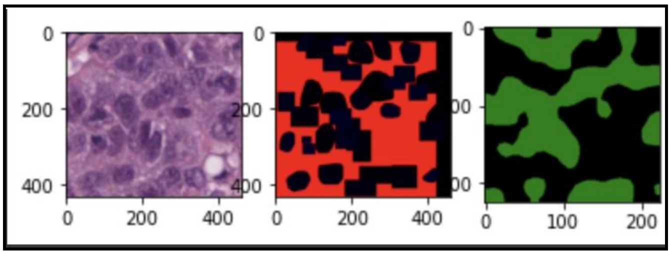


Fig. 1. FCN-ResNet50 Prediction

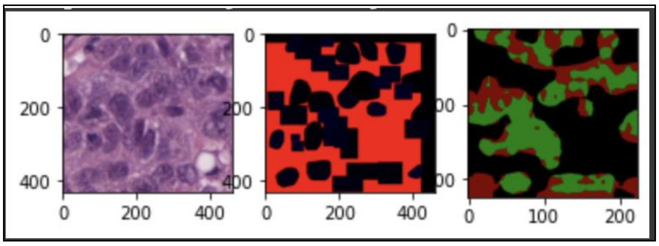


Fig. 2. DeepLabv3-ResNet50 Prediction

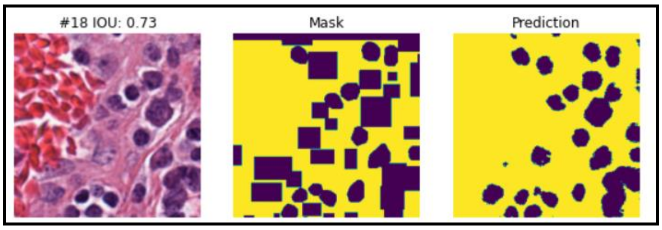


Fig. 3. U-net Prediction

V. CONCLUSION

In this project, we experimented on the NuCLS dataset. NuCLS dataset consists of cell images that contain both normal cell and cancerous cells. Here, we tried different approaches to do segmentation on the cell to get each cell from different classes. In these approaches, we used FCN-ResNet50, FCN-ResNet101, DeepLabv3-ResNet50, DeepLabv3-ResNet101 and U-Net. In our results we saw that we got accuracies between 0.58 and 0.8. We also showed the qualitative results for some of the models we used in our experiments.

REFERENCES

[1] *See* https://www.who.int/news-room/fact-sheets/detail/breast-cancer#:~:text=The%20objective%20of%20the%20WHO,globally%20between%202020%20and%202040

[2] Kowal, Marek & Filipczuk, Paweł. (2014). Nuclei segmentation for computer-Aided diagnosis of breast cancer. International Journal of Applied Mathematics and Computer Science. 24. 10.2478/amcs-2014-0002.

[3] Hameed, Zabit & Zahia, Sofia & Zapirain, Begoña & Aguirre Anda, José Javier & Vanegas, Ana. (2020). Breast Cancer Histopathology Image Classification Using an Ensemble of Deep Learning Models.

[4] See https://sites.google.com/view/nucls

[5] *See* https://en.wikipedia.org/wiki/U-Net