Final Project Report for CS 184A/284A, Fall 2019

Project Title: Histopathologic Cancer Detection

Student Name(s)

Dilsher Bhat, 62871770, dilsherb@uci.edu

Timothy Hakobian, 56197426, thakobia@uci.edu

Link for Data (Since data size is bigger than 5MB):
https://www.kaggle.com/c/histopathologic-cancer-detection/data

1. Introduction and Problem Statement

Detecting metastatic cancer is a challenging task for pathologists. Pathologists have to look over large amounts of tissue to identify metastases, which have the possibility of being as small as single cells. This process leaves room for a lot of errors for pathologists. Errors which can have detrimental effects on patients. Early and accurate detection is crucial when it comes to treating cancer. Through our project, we would like to create an algorithm which takes in digitized lymph node sections and analyze them for metastases. Our machine-learning algorithm would use Convolutional Neural Networks to train our AI to learn cancer detection skills.

This project will use binary classification utilizing CNNs to identify metastatic cancer in small image patches taken from a subset of larger digital pathology scans of the CAMELYON16 dataset and CAMELYON17 dataset. Our classification accuracy will be measured by area under the Receiver operating characteristic curve.

2. Related Work

There has been a lot of work done in finding ways how to detect cancer cells quickly because AI in medicine can have a lot of potential. An article from Volume 16 of Informatics in Medicine Unlocked by Sumaiya Dabeer discusses using image processing techniques on histopathology images. It discusses using a dataset of images of sample breast tissue taken from a biopsy. The article states “We have described different Deep Neural Network architectures, especially those adapted to image data such as Convolutional Neural Network.” Our project fits into what’s being discussed in this article because we also built a convolutional neural network to scan images and determine if they contain cancer cells.

The book called “The Applications of Bioinformatics in Cancer Detection” by John F. McCarthy states “Feature selection and reduction techniques help with both computation time and overfitting problems by reducing the number of data attributes used in creating a data model to those most important for characterizing the hypothesis.” Our project fits into this work

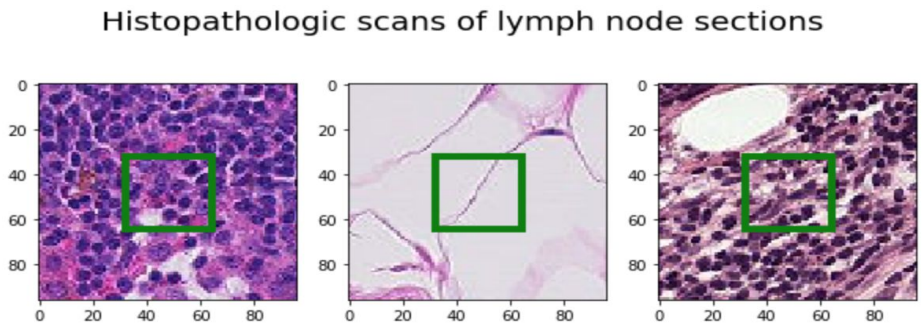
because we also found great use feature selection to ensure our model didn't become too complex based on only the training data. Feature training was also very useful in our model because it allowed the data to get trained faster by reducing its complexity.

3. Data Sets

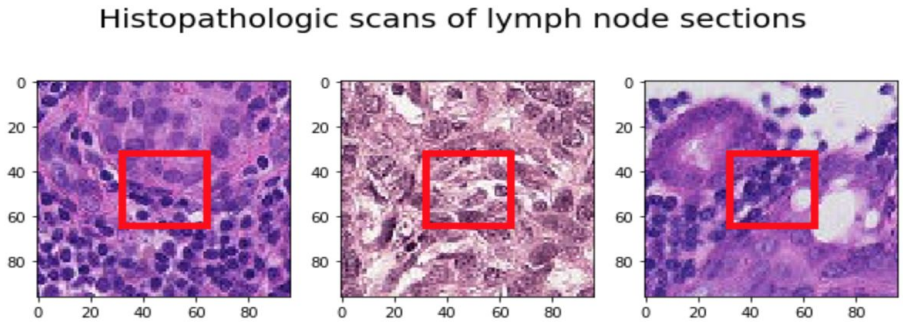
Our data set was provided to us by the Histopathological Cancer Detection competition on Kaggle. We will be utilizing the PatchCamelyon (PCam) dataset.

The dataset consists of histopathological images. It is a unique dataset of annotated, whole-slide digital histopathology images of glass slide microscope images of lymph nodes that are stained with hematoxylin and eosin (H&E). This dataset is, in fact, a combination of two independent datasets collected in Radboud University Medical Center (Nijmegen, Netherlands), and the University Medical Center Utrecht (Utrecht, the Netherlands) who produced the slides by routine clinical practices and a trained pathologist would examine similar images for identifying metastases. Thus, both the relevancy and the quality of our dataset is unquestionable.

Our data also contains a sample_submission.csv file and a train_labels.csv file which both contained information on whether the middle 32x32 px region of each picture has at least one pixel of tumor tissue. The csv files, one used for training and the other used for training, label every image with a 0 or 1. 1 meaning that there is cancer tissue associated with the image that the name is provided for. The region outside of the middle region is not associated with the labels and therefore cancer tissue isn't accounted for in those regions in our algorithm. The outside regions are given so that issues with zero-padding aren't a problem with some models.
Negative Tumor Samples:



Positive Tumor Samples:



3. Description of Technical Approach

We considered using SVM, Support Vector Machine, which separates the data using a hyperplane because it can get a relatively accurate model with less-grid searching. However, we found that neural networks would be better because of their ability to learn the features of any data structure. Since tumor tissue is difficult to detect and can usually involve many features, we thought constructing a neural network would be a good idea.

We used openCV for pre-processing images and perform image augmentation in terms of geometric transformations such as axis flipping, color space changes, cropping and rotation. We knew that we had to build convolutional layers but wondered if we should use the pytorch or tensorflow library. We decided to use tensorflow because of the keras library which portrayed a lot of potential in what we were trying to do. After pre-processing our images, we used keras library to generate a convolutional neural network model based on binary image classification to identify metastatic cancer in the small image patches of the larger digital pathology scans.

We decided to use a sequential model and the architecture is as follows:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
batch_normalization_4 (Batch (None, 8, 8, 64)          256
_________________________________________________________________
dropout_4 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
batch_normalization_5 (Batch (None, 4, 4, 128)         512
_________________________________________________________________
dropout_5 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 4, 4, 128)         147584
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
batch_normalization_6 (Batch (None, 2, 2, 128)         512
_________________________________________________________________
dropout_6 (Dropout)          (None, 2, 2, 128)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                16416
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 66
=================================================================
Total params: 258,594
Trainable params: 257,954
Non-trainable params: 640
```

After training and testing, we evaluated our model using the area under the receiver operating characteristic (ROC) curve which is a good metric because it clearly shows the data pertaining to whether a patient has metastatic cancer or not since its utilized usually to check the performance of binary classifiers.

4. Software

We coded in Python (Jupyter Notebook) and used openCV, numpy, and keras libraries. OpenCV is a library that will help with our functions related to computer vision. Numpy helped us with creating large, multidimensional arrays and matrices. Keras gave us access to an open-source neural network library.

We also found the Kaggle notebooks very helpful because they allowed us to have the data available without downloading it. The dataset for this project is extremely large, which made it very hard to import into Jupyter Notebook on one of our computers. Therefore, Kaggle's notebook software allowed us to access the data without using up a lot of storage. It also allowed us to run multiple models at the same time, which gave us a lot of room to experiment without wasting time waiting for each run. This was especially important because training the model takes a long time when the data is big.

We looked at a lot of other notebooks in the competition to further our knowledge of how exactly others have approached the data. We examined the models that others had used to get high accuracy percentages and came to the conclusion that using a sequential model and adding layers by using the tf.keras.layers.Conv2D would most likely get us high results. Looking at other related pieces of code having to do with conv2D layers, we also realized that we would have to generate the data from the images and rescale them so they could fit into the model we were creating.

5. Experiments and Evaluation

We decided to first experiment with a baseline single layered CNN model which we trained for 10 epochs and got the following results:

```
('Test loss:', 0.6238402219772339)
('Test accuracy:', 0.7664)
```

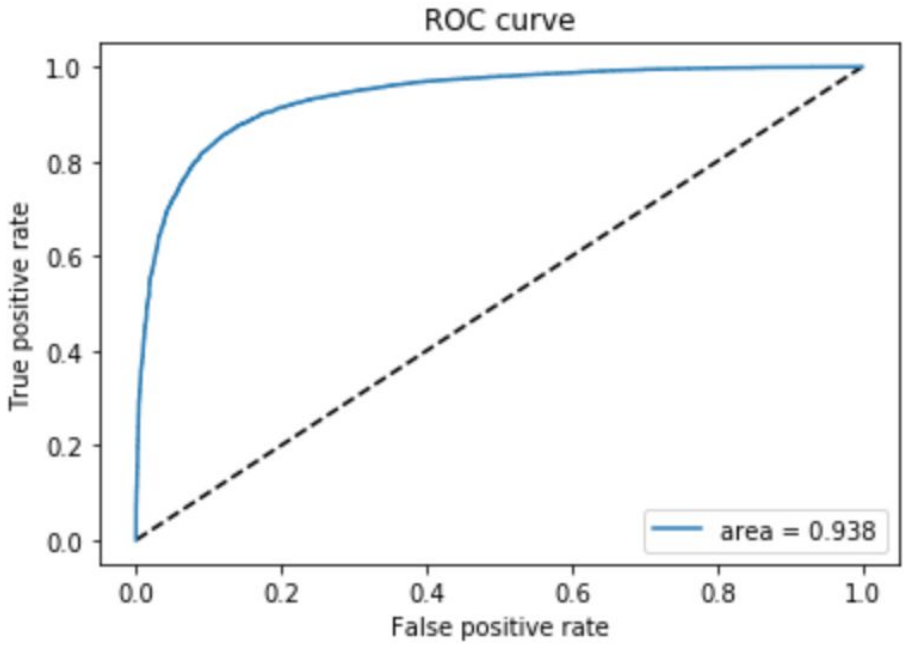
Then we added 2 more layers and got the following results:

```
Test loss: 0.24964626643657684
Test accuracy: 0.9079
```

Then we decided to introduce BatchNormalization() and dropout layers because our training accuracy was high yet then the val_loss and test accuracy wasn't as high. We got the following the results:

```
Test loss: 0.1776888778269291
Test accuracy: 0.9352
```

Our final results were validated by calculating and plotting the area under the ROC curve which yielded the following results:



6. Discussion and Conclusion

Our area under the curve is .938/1 which states that our model performs approx 93.8% of the classifications successfully to predict which scans have metastatic cancer. This model can be improved further to reach higher success rates to ensure successful positive cancer detection.

7. A separate page on *Individual Contributions*

Timothy Hakobian:

I met up with my partner at the beginning of the quarter and we discussed a few topics that we could do. We liked the topic of Histopathological Cancer Detection, so we decided on that. The project proposal was worked on jointly. For the powerpoint presentation slides, we worked on them jointly. Before presenting, we discussed who would present on each part. I presented on my parts and my partner presented on his. The code to the project was worked on jointly but my partner had more knowledge on neural networks, so I did ask for his help a lot. My

specific task for the code was building a baseline model. The project report was worked on jointly.

Dilsher Bhat:

I met up with my partner at the beginning of the quarter and we discussed a few topics that we could do. We liked the topic of Histopathological Cancer Detection, so we decided to pursue that. The project proposal was worked on jointly. For the powerpoint presentation slides, we worked on them jointly. Before presenting, we discussed who would present on each part. I presented on my parts and my partner presented on his. The code to the project was worked on jointly where I worked on pre-processing the images and improving the model. The project report was worked on jointly.

Sources

Dabeer, Sumaiya, Maha Mohammed Khan, and Saiful Islam. "Cancer diagnosis in histopathological image: CNN based approach." *Informatics in Medicine Unlocked* 16 (2019): 100231.

MCCARTHY, J.F., MARX, K.A., HOFFMAN, P.E., GEE, A.G., O'NEIL, P., UJWAL, M.L. and HOTCHKISS, J. (2004), Applications of Machine Learning and High-Dimensional Visualization in Cancer Detection, Diagnosis, and Management. Annals of the New York Academy of Sciences, 1020: 239-262. doi:10.1196/annals.1310.020.