

Histopathologic Cancer Detection using Convolutional Neural Network to do Binary Classification

Aidan Daly
Boston College
dalybu@bc.edu

Matthew Smallhouse
Boston College
smallhou@bc.edu

1. Introduction

The main motivation for our project was to take something we were both interested in and apply it to what we've learned in Computer Vision. We are both interested in the cross-section of technology and medicine, so naturally, cancer detection was one of the first project ideas that came to mind. We recognized a couple of factors that made cancer detection a good project. Firstly, the data was plentiful and easy to find. Finding a large volume of quality data is integral to developing a successful model, and we found quite a few different options that would work. Secondly, cancer detection is a binary classification problem, which is something we have previous knowledge of and felt comfortable with. Finally, cancer detection is something that has the ability to help people. We believe that this project fits the school's slogan of "Men and women for others," and has the potential to contribute to the important field of cancer detection. It has been approximated that in 2022 1.9 million people will be newly diagnosed with cancer, and we believe that this technology has the potential to help many of them.

Our high-level goal of the project was to build a model that could detect whether a given image contains cancer or not. We decided that the best approach was to use histopathology images because there is lots of readily available data, and the images are clear and contain more variation than other types of imaging, such as CT scans or ultrasounds.

The problem we are trying to answer is very important at this time. The rapid advancement of technology has proved to be important for solving problems like cancer detection, which is why there are entire companies dedicated to solving similar problems like PathAI and ImageneAI. The products these companies make are forever changing the way we discover and diagnose cancer. This goes to show that technology has greater applications than just cheap entertainment or a news delivery source. Rather, technology has the potential to provide life-saving capabilities, which is why this is such an important emerging field.

In regards to the production of the source code for this project, Aidan worked on downloading, visualizing, and processing the data, as well as some areas of the training and testing. Matt worked substantially on the training and testing of the model as well as filter visualization to see how the model modifies a given input image through each convolution layer. We both worked together on the model and offered an equal amount of work. More information on our contributions can be found in Section 6.

2. Related works

There are many models available for view under the Kaggle competition named "Histopathologic Cancer Detection." [2] This is where we got our dataset from and searched around to find what other people had done to solve this problem. The competition closed 4 years ago, but still allows non-competitive submissions. Currently, just under 1,150 teams have submitted models.

Related to the whole field of biotechnology, there are a number of companies that currently leverage deep learning and artificial intelligence to aid the diagnosis and detection of diseases. PathAI is one such company that focuses on the same histopathology images we used in our model [6]. However, they focus on complex diseases and are even able to predict the efficacy of new types of therapy.

Similar to PathAI, Imagene AI applies artificial intelligence to pathology scans and molecular testing. Their mission is to take advantage of the accuracy and speed (once already trained) of deep learning to provide patients with low-cost, fast diagnoses of cancer.

The National Cancer Institute found that deep learning outperformed doctors in cervical cancer detection, with an accuracy rate of 91% as compared to 69% and 71% by doctors using traditional methods [4]. Another advantage of using models for cancer detection is that they are useful for types of cancer that usually require particular expertise to diagnose. For example, the National Cancer Institute found that AI models greatly helped clinics detect prostate cancer, which can be difficult to detect accurately using typical

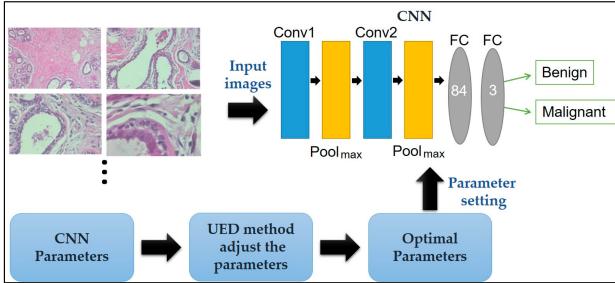


Figure 1. Flow of CNN used on Pathology Image. The image is processed and input into the model. At this point, it is passed through a number of hidden convolutional layers. Finally, it is given to a fully connected layer and receives its prediction. [3]

MRI imaging techniques.

3. Method

After deciding upon a project, the first step was to find a large set of quality training data, which is essential to building a good model. We started digging through lots of medical data of all different types, some MRI, some ultrasound, and lots of pathology. It soon became clear that histopathology would be the best fit for our project. We settled on this option because pathology images are both widely available and simpler than other types of images. After now looking through lots of histopathology datasets, we decided on a Kaggle dataset that contains over 220,000 images for training data. The dataset has approximately a 40-60 split of cancerous and non-cancerous images respectively. These images are scans of lymph node sections. This means that our model is limited to just the lymph nodes, but we believe that this is still applicable to our goal. Limiting the images to just the lymph nodes also gives a smaller scope of learning, hopefully leading to a more accurate model.

The next step was to decide on the structure of our model. We decided that using a Convolutional Neural Network (CNN) would be the best for our binary classification task. We have used CNNs extensively throughout the semester and therefore felt comfortable with their mechanics and confident that we would be able to build a model to the best of our abilities. Next, we needed to determine an activation function, the number of layers, and the structure of each layer. We decided upon a Rectified Linear Activation (ReLU) for our activation function since we know it is reliable and have used it previously in class. We tested 3, 4, and 5 hidden convolutional layers, and settled upon 4 layers after some testing. We also found that our fully connected layer worked best with three linear functions. With these parameters in place, the structure of our model was finished.

For the training and testing of our model, we mirrored

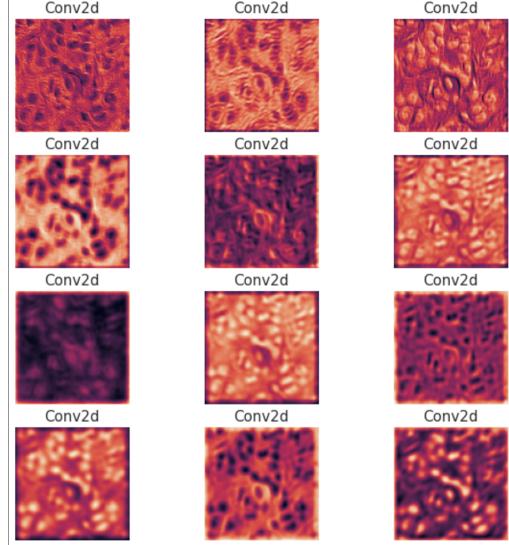


Figure 2. Convolutional layer filtering throughout model. This is an example image from our dataset that has gone through our model, which our model predicted to be Cancerous. This displays the filtering our model uses to make predictions after each convolution [7].

problem sets 3 and 4, along with some of the labs. We also took inspiration from projects online that were similar to ours. This proved to be helpful for writing our training section, where it wouldn't make sense to reinvent the wheel and build ours from scratch. Instead, this allowed us to spend more time fine-tuning our model. After formulating our training function, we built an accuracy checker to be used on the validation and data. We decided upon checking our accuracy after every epoch on our validation set, which we incorporated into our training function. This allowed us to watch the progression of the model's learning and see the effects of tuning hyperparameters. The training and validation took a lot of trial and error, which you can read about in Section 4.

After our third meeting with Professor Wei, he recommended that each of us investigate one hypothesis that we believed could improve the performance of our model. Aidan looked into the idea of randomizing training data order. Shuffling our training dataset allows for our model to gain a more generalizable set of hyperparameters when being trained [1]. This is because through randomness our model can never expect what could be coming next. Matt added a randomized vertical and horizontal flip to the data preprocessing section, also with the idea of greater generalizability. After implementing each of these changes, we saw an improvement of accuracy, and therefore conclude that they added to the quality of our model.

4. Experiments

Most of our experimentation was centered around training. To develop a model that would allow us to produce the best accuracy given a long period of time for training, we first conducted training on fewer data and for fewer epochs. This allowed us to train the model relatively quickly in order to see the effects of different hyperparameter settings. We spent hours running the model, getting an output, revising our parameters, and repeating. This allowed us to find an optimal learning rate and the number of epochs. We also tinkered with the batch size, for which we tried 128 and 256 images.

Because the dataset consists of over 220,000 images, we needed to train the model on only a portion of this if we wanted to quickly see the results of these changes. First, we ran the model on one-sixth of the dataset to tune the learning rate, number of epochs, and batch size. Once we found parameters that led to a validation accuracy we were satisfied with, we began to test on the full dataset, which led to better results.

We eventually settled on the following settings: a learning rate of 0.01, batch size of 128, and 30 epochs run on the full dataset. More on the conclusions we found from these settings in Section 5.

We also spent a lot of time tinkering with the format of our training function. Currently, our function checks accuracy of our validation set after every epoch, to give a general idea of how training is going. We also experimented with checking the accuracy after every epoch with a smaller subset of the training data. We realized that this produced an overly optimistic accuracy of our model and did not represent the accuracy of our model on unseen data. This could lead to a model that does not generalize well to real-world data. This is why we ended up deciding to check the accuracy of the validation set after each epoch.

5. Conclusions

Our model ended up with a 98% accuracy on the set-aside testing dataset. Although our model performed well on our particular dataset, it has some limitations that prevent it from being implemented as a legitimate cancer detection model. Firstly, the data is limited to only scans of the lymph nodes, so the model may have undetermined behavior on scans of other parts of the body. Additionally, the cancerous regions within the images are located within a 32x32 pixel region in the center of the image.

Reflecting upon our model, one thing we could have incorporated was preprocessing some of our images to be augmented. By randomly rotating some images, shifting the center, or adding noise, the model may have learned to be more robust. This also would allow us to better prevent overfitting the model to our training data.

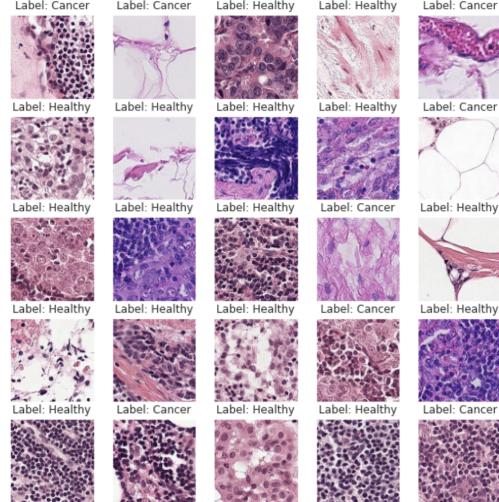


Figure 3. Examples of output on unlabeled test data. These images were run through our model, and given a label based on the model’s prediction. There is no guarantee as to whether these labels are accurate. This is similar to if the model was given a fresh image from someone being tested for cancer.

As far as deep learning models being used within the medical field as a whole, the Association of American Medical Colleagues recognizes some current limitations. These models are great at doing one particular task, but they do not necessarily generalize well [5]. Realistically, they believe that these models are best used in conjunction with a doctor’s judgment, where they have the ability to increase the accuracy of detection and reduce the number of false positive errors. These deep learning models may have the most potential to make a difference in countries with limited access to healthcare and places that suffer from a shortage of doctors.

Overall, we liked this project as a whole because we felt that implementing something closer to a full-scale model rather than a smaller PSet or lab really helped cement some of the ideas from the course. We had done many of the individual pieces on their own, but putting everything together proved to be challenging and a bit more work than we had originally anticipated. However, we enjoyed creating something mostly from scratch that could actually be applied to a real world problem. We recognize the limitations of our model, but both of us are very satisfied with what we were able to build.

6. Contribution

Throughout this project, our goal was to split up the work as evenly as possible and we both believe we put a lot of effort into the project.

Aidan specifically worked on the data downloading, processing, and visualization that you will find in Sections 1

and 2 of the Google Colab. He also worked on the accuracy computation within the training section and did hours of experimentation for the parameters of the training function. He built the visualization of the accuracy of the model through the epochs and the predictions of the model on the test set. He also worked heavily on the production of the model itself. Finally, in terms of source code he worked on the saving of the model to Google Drive so it could be loaded back into the Google Colab for viewers to experiment with.

Matt worked on building out the training function for our model. This took a lot of time and tinkering to fix all of the bugs that would arise. He also worked on the testing of the model on unlabeled data. He made our visualization that you will find in Section 6 of our project that shows how our model actually filters our image as it goes through. Finally, Matt worked a lot on the build of our model. We spent a lot of time in person coding out the parameters and padding for each layer. Finding the optimal number of layers for the model and how many convolutions we should use took lots of trial and error.

In regard to this report, Aidan wrote the Introduction, Method, and Experiments sections. Matt wrote the Related Works, and Conclusions, and gathered all of our References.

References

- [1] DeepwizAI. Why random shuffling improves generalizability of neural nets. Article on explanation of improvements with randomness in a dateset (bit.ly/3YDiyse). [2](#)
- [2] Kaggle. Histopathologic cancer detection, 2018. Competition to Identify metastatic tissue in histopathologic scans of lymph node sections (bit.ly/3hC1jXT). [1](#)
- [3] Cheng-Jian Lin. Optimization of deep learning network parameters using uniform experimental design for breast cancer histopathological image classification, 2020. [2](#)
- [4] NCI. Artificial intelligence - opportunities in cancer research. AI job opportunities in cancer research (bit.ly/3FJTIDQ). [1](#)
- [5] Association of American Medical Colleges. Will artificial intelligence replace doctors? This article goes in-depth about the possibility of AI taking over doctor's roles in the health-care industry (bit.ly/3Wtdllc). [3](#)
- [6] PathAI. Pathai about us. About us section of PathAI Website (bit.ly/3WsSqI3). [1](#)
- [7] Ravi Vaishnav. Visualizing feature maps using py-torch. 2021. Feature map layering visualization tutorial (bit.ly/3v3MHDC). [2](#)