Andrew Plum

Department of Computer Science

CS 474: Deep Learning

Skin Cancer Detection

Section 1: Project Introduction

The focus of this project is to design a deep learning model which can detect cancer. To evaluate the model’s performance, the test accuracy of the model will be assessed. I will be using the skin cancer detection dataset provided. The dataset contains images of skin moles which can be classified as either benign or malignant. Because the project entails classifying images into two categories (benign or malignant), the deep learning model that I will use is a convolutional neural network with convolutional layers and max pooling layers where the last layer will be a dense layer with a single node which will be using sigmoid as its activation function. I will also be using binary cross-entropy as the loss function of the model. The initial code template I started with and the data for the project was downloaded at the links listed in the references below. The initial code template prepares the image data to be used in a model. The data is currently split into training and test data, but I have further split the data into training, validation, and test sets.

Section 2: Preliminary Results

The baseline model’s test performance is roughly 75%. Thus far, I have achieved a test performance of 78.63% with a convolutional neural network. The training accuracy was at 79.81%. The current strategy I have to improve upon this performance is to increase the model’s complexity until the model no longer sees any test performance improvements from increased model complexity because it is starting to drastically overfit to the training data. After overfitting is present, I will finetune other model hyperparameters than the structure to see if I can further reduce the test loss during the model’s training do reduce the overfitting. In the current model iteration, I have split the dataset into 60% training data, 20% validation data, and 20% test data. I am continuing to experiment with the model hyperparameters to improve the model’s test performance.

Section 3: Project Timeline

|  |  |  |  |
| --- | --- | --- | --- |
| Step Number | Step Name | Sub step | Deadline |
| 1 | Understand the data | Look at images | 11/30/2024 |
|  |  | Look at distribution | 11/30/2024 |
|  |  | Familiarize handling the data | 11/30/2024 |
| 2 | Build basic model | Make convolutional neural network using keras | 11/30/2024 |
| 3 | Train the basic model | Run the training data through the model | 11/30/2024 |
| 4 | Make predictions using the basic model | Be able to make a prediction with the model | 11/30/2024 |
| 5 | Evaluate predictions of the basic model | Get a numeric percentage of how accurate the model is on the training and test data. | 11/30/2024 |
| 6 | Improve model after basic model works | Modify the structure of the model | 12/10/2024 |
|  |  | Modify the other hyperparameters of the model. | 12/10/2024 |
|  |  | Make predictions and evaluate improved model performance | 12/10/2024 |
| 7 | Write report | Have a finalized model to write the report on | 12/13/2024 |

Section 4: References

<https://keras.io/api/layers/>

<https://www.tensorflow.org/tutorials/images/cnn>

<https://canvas.uidaho.edu/courses/30734/files/3673113?wrap=1>

<https://canvas.uidaho.edu/courses/30734/files/3268232?wrap=1>