IBM Machine Learning - Deep Learning: Final Project

10/30/2023

Project Synopsis:

Using the TensorFlow Dataset: colorectal_histology to build a Convolutional Neural Network (CNN) in order to classify different colorectal cancer histology. The data for this project comes from the paper “Multi-class texture analysis in colorectal cancer histology” by Kather et al. 2016. The goal is to achieve a high accuracy for differentiating between all 8 cancer histology classes and evaluate which classes are the most difficult to classify. I built 3 different machine learning models for this project (CNN, CNN Augmented, and CNN with Transfer Learning).

Data Preprocessing and EDA:

The colorectal_histology dataset is a balanced dataset (Figure 3) that contains 5000 RGB images of 8 classes of different classes of histology samples. Figure 1 shows an example RGBimage for each of the 8 classes within the dataset.

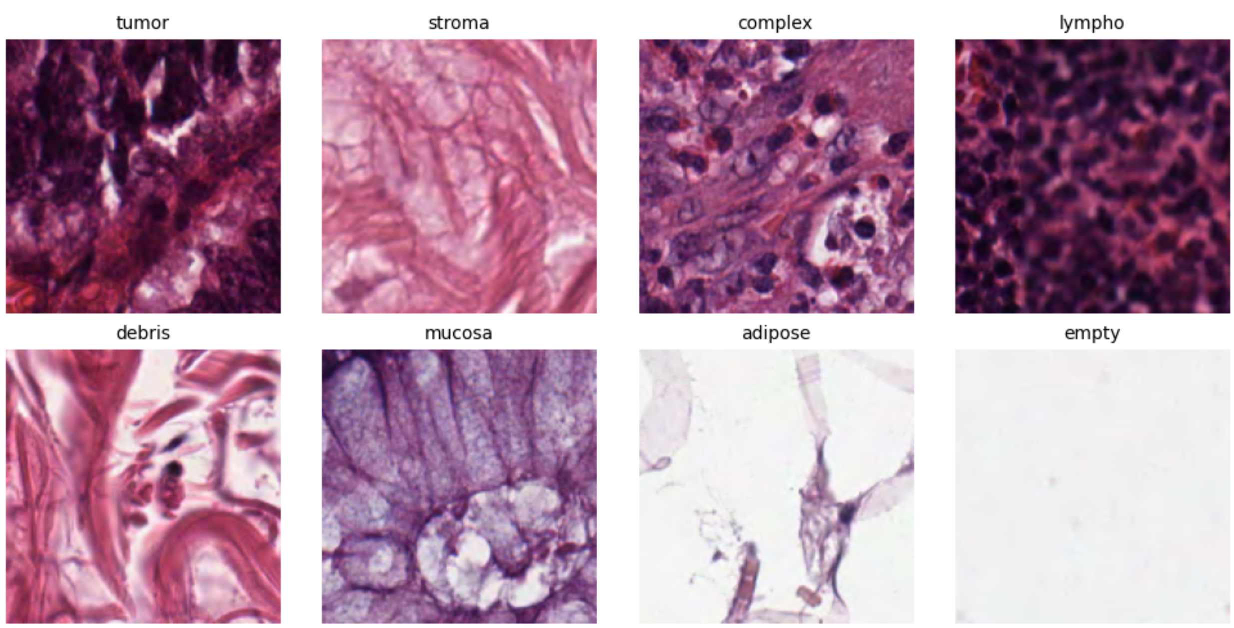

Figure 1. Random selection of different images for all of the classes within the dataset

I prepared the dataset for training and EDA using a preprocessing step to get the images into the correct shape and to store them in numpy arrays. I used a shape of (-1, 150, 150. 3) for reshaping. This resulted in an image array size of (5000, 150, 150, 3). I then decided to examine the average intensity of each pixel for different colors in order to determine if there were any trends with classification and color intensity. There did appear to be different signatures for each class, but their similarities varied significantly. Some classes had much similar distributions, while being vastly different distributions to other classes.

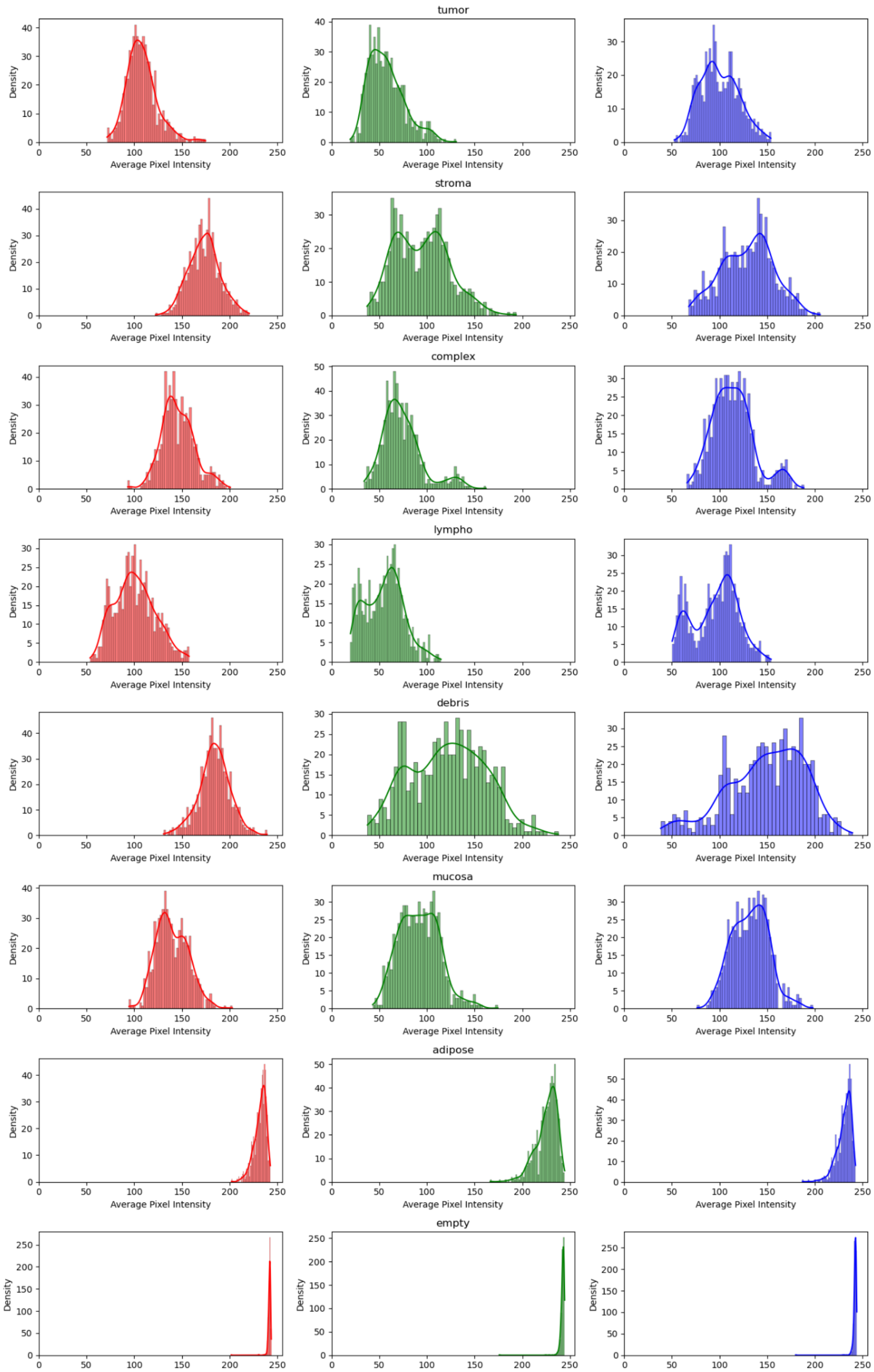

Figure 2. Average pixel intensity for red (left), green (center), and blue (right) for different classes.

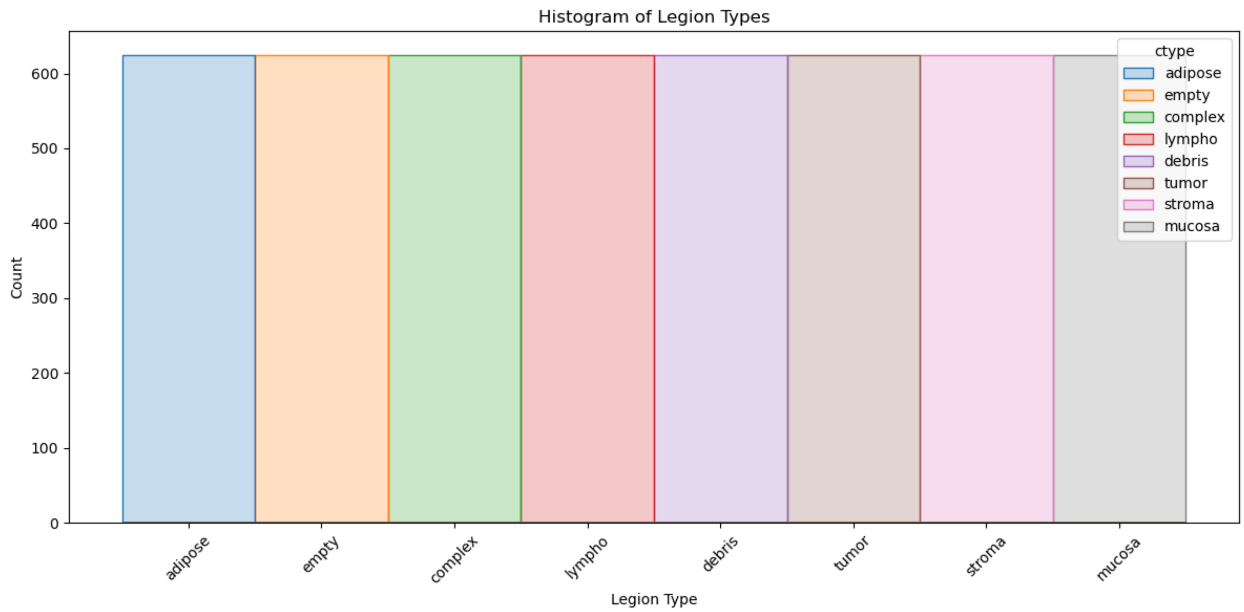

Figure 3. Distribution of the dataset showing a balanced sample.

To further prepare the dataset for training I created a train and testing split of the data. I used a training size of 80% and a test / validation size of 20%. For training with augmented images I created 4000 images using randomly selected images from the dataset. I also used one hot encoding for the labels in all cases.

CNN Architecture

The CNN for both the initial and augmented models used the same architecture. I build a CNN with three convolutional layers and one flattening layer with a Relu activation function. I then added two dense layers, with the first having a Relu activation function, and the last using a SoftMax activation function.

```
# CNN architecture
model = Sequential()

# CONV L1
model.add(Conv2D(24, (3,3), 1, activation='relu',input_shape=(img_width,
img_height, 3)))
model.add(MaxPooling2D())

# CONV L2
model.add(Conv2D(32, (3,3), 1, activation='relu'))
model.add(MaxPooling2D())
```

```
# CONV L3
model.add(Conv2D(24, (3,3), 1, activation='relu'))
model.add(MaxPooling2D())

# Flattening Layer
model.add(Flatten())

# Dense Layers
model.add(Dense(200, activation='relu'))
model.add(Dense(8, activation='softmax'))
```

<u>Transfer Learning</u>

In addition to the CNN I built, I used Inceptionv3 with pre-training on ‘imagenet’ in order to improve accuracy. The model is the third edition of Google's Inception Convolutional Neural Network and specializes in image detection. The original Inception model was introduced during the ImageNet Recognition Challenge. Figure 4 shows a schematic of the Inceptionv3 architecture.

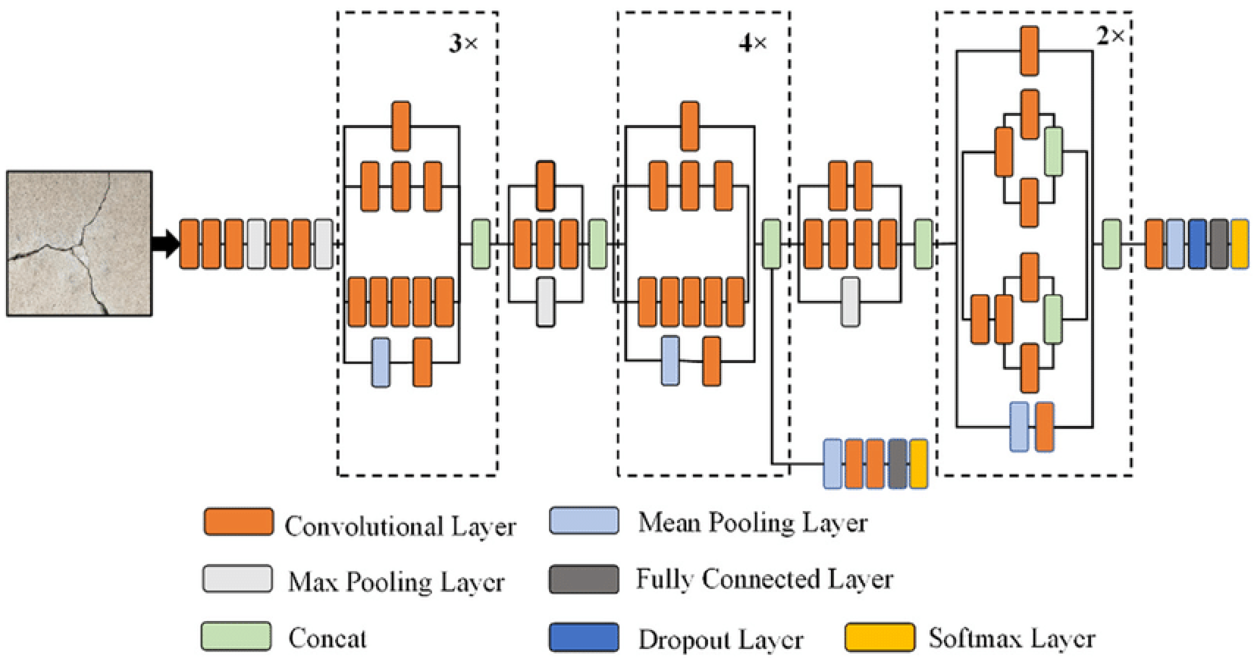

Figure 4. Inception-V3 model (Ali et al. 2021).

Training & Results

<u>CNN from scratch</u>

The first model achieved an accuracy of 80.5% on the test data set, over 15 epochs with a learning rate of 1e-4 (Figure 5). Figures 6 and 7 show the total predictions and accuracy for the model's ability to classify each class. The model struggled with accurately predicting Stroma, Complex, Mucosa and Debris classes. A possible cause could be the small dataset, which suggests some image augmentation is needed.

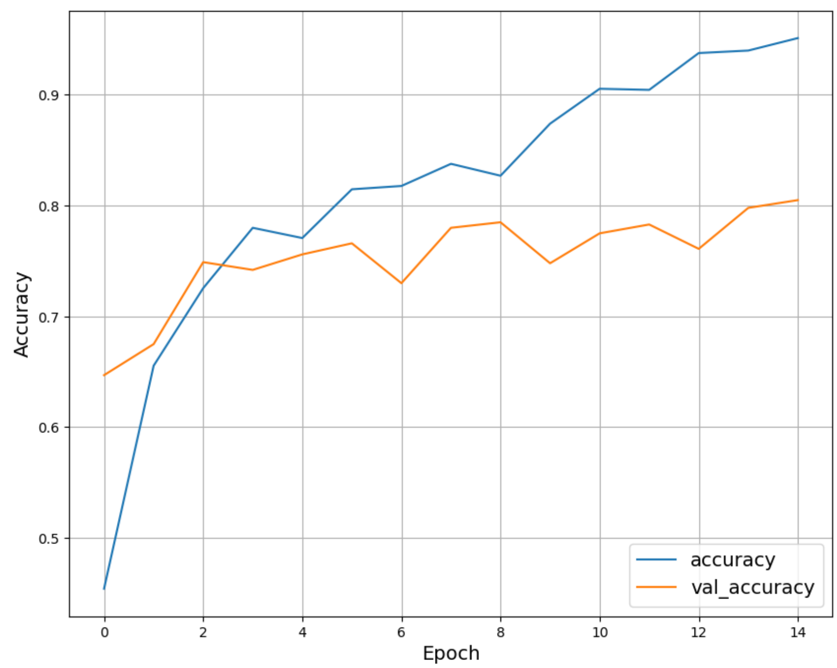

Figure 5. CNN from scratch accuracy over different epochs.

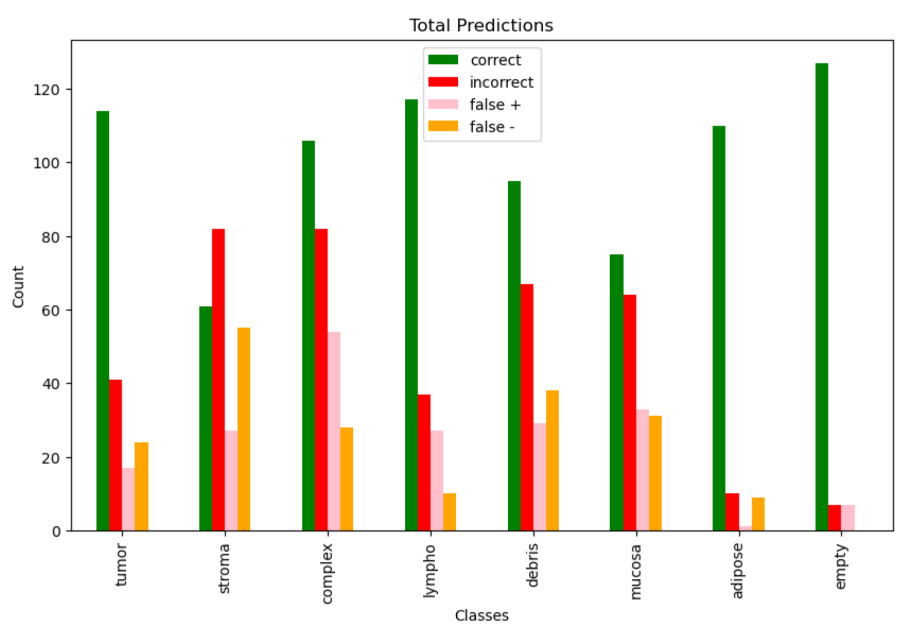

Figure 6. Total predictions correct from the CNN from scratch model.

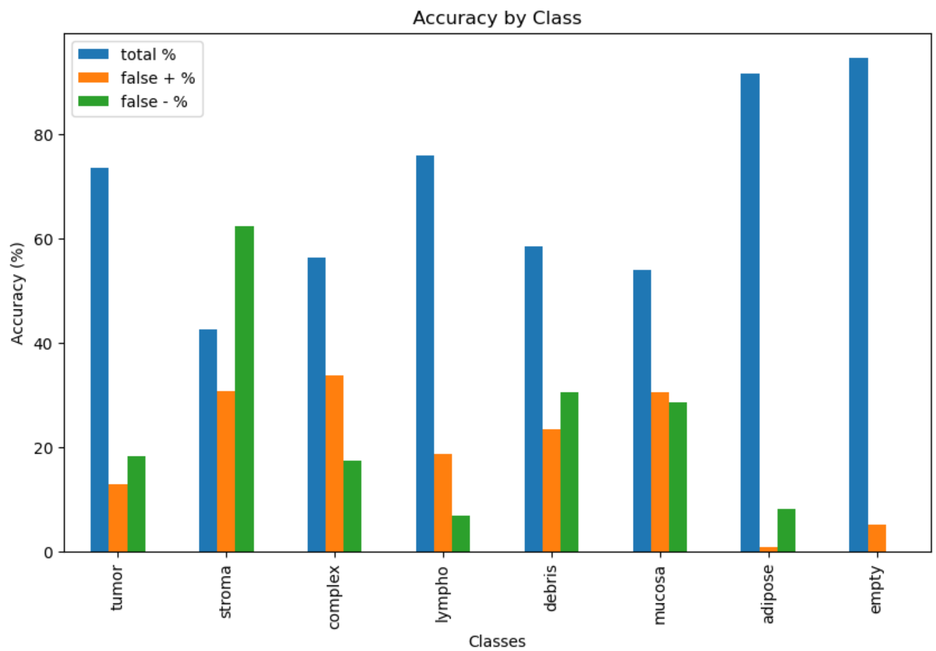

Figure 7. Accuracy for each class for the best model of the CNN from scratch.

	total_pred	true_total	correct	incorrect	false +	false -	false - %	false + %	total %
tumor	131.0	138.0	114.0	41.0	17.0	24.0	18.32	12.98	73.55
stroma	88.0	116.0	61.0	82.0	27.0	55.0	62.50	30.68	42.66
complex	160.0	134.0	106.0	82.0	54.0	28.0	17.50	33.75	56.38
lympho	144.0	127.0	117.0	37.0	27.0	10.0	6.94	18.75	75.97
debris	124.0	133.0	95.0	67.0	29.0	38.0	30.65	23.39	58.64
mucosa	108.0	106.0	75.0	64.0	33.0	31.0	28.70	30.56	53.96
adipose	111.0	119.0	110.0	10.0	1.0	9.0	8.11	0.90	91.67
empty	134.0	127.0	127.0	7.0	7.0	0.0	0.00	5.22	94.78

Table 1. Table showing results for the CNN from scratch model.

Image Augmentation

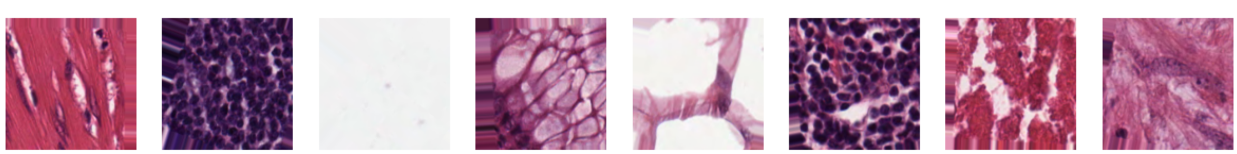

Figure 8. Random sample of augmented images.

I added 4000 images to the training set using image augmentation. The rerun of the previous model with augmented data did not result in any model improvements (79.6% accuracy). I suspect this might be due to the warping or distortion of features in the augmentation process important to classification of different legions. This can be seen in a few augmented images shown in Figure 8. Warped portions at the image edges could lead to increased difficulty in classification. This can be seen in Figure 9 with the increased false positive rate in the 'stroma' class.

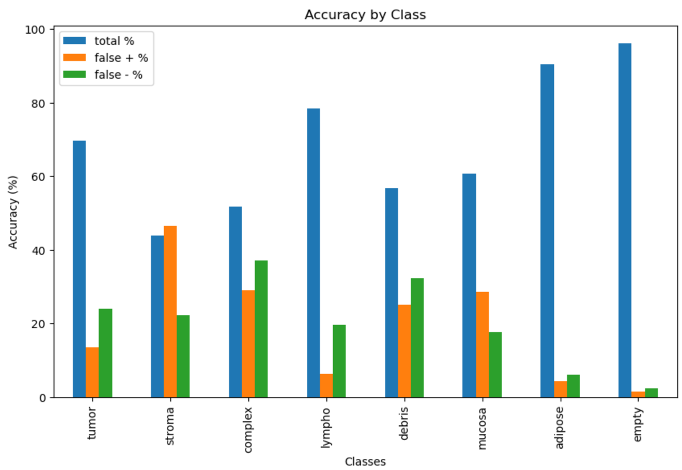

Figure 9. Augmented image model accuracy by class.

<u>Transfer Learning - InceptionV3</u>

Another solution to improve model accuracy was to implement transfer learning. Here, we take an InceptionV3 model previously trained on the 'imagenet' data, unfreezing the last 10 layers, and adding two dense layers to the top of the model. This resulted in a solid improvement on the previous two models. The overall model accuracy rose to 88.3%, an 8% improvement. Figures 10 through 12 show the noticeable improvements from previous models. Some of the biggest accuracy gains can be seen in the 'stroma' and 'complex' classes. It appears that 'stroma' suffers from higher false positive rates, whereas 'complex' contains too many false negative cases.

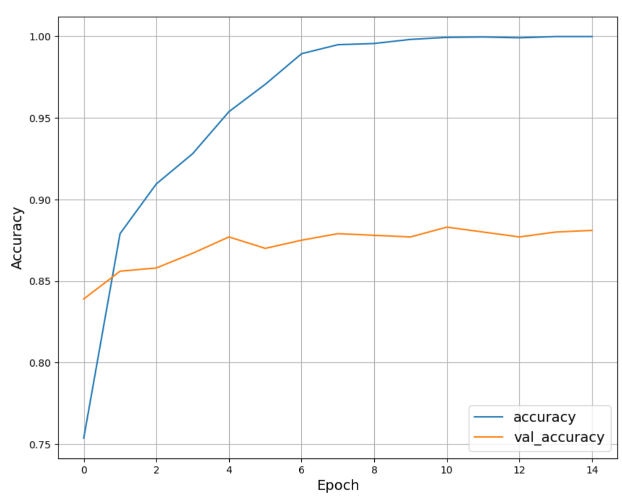

Figure 10. Transfer learning accuracy over different epochs.

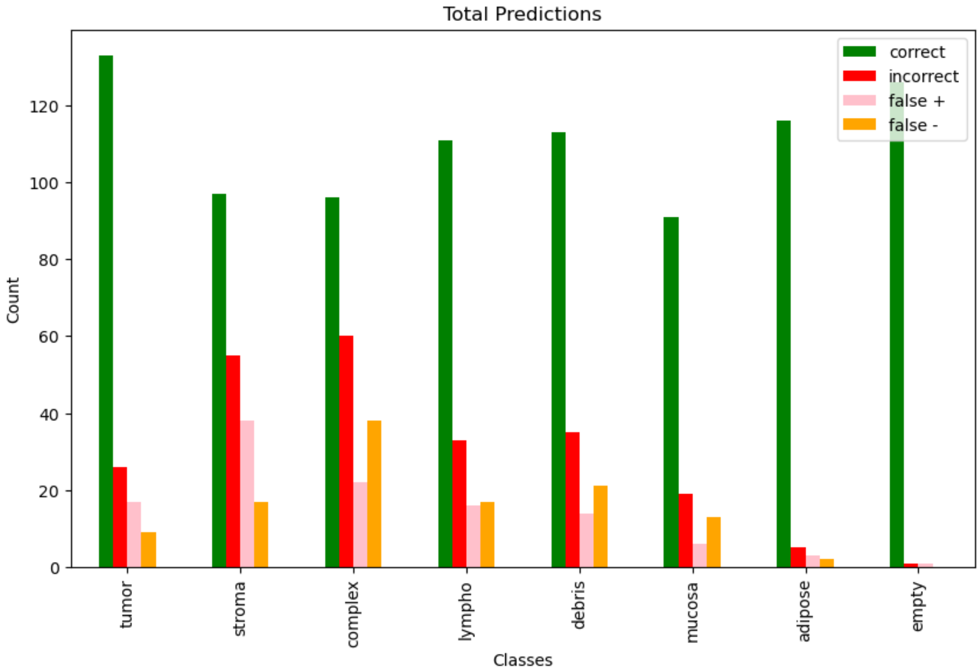

Figure 11. Total predictions of transfer learning model.

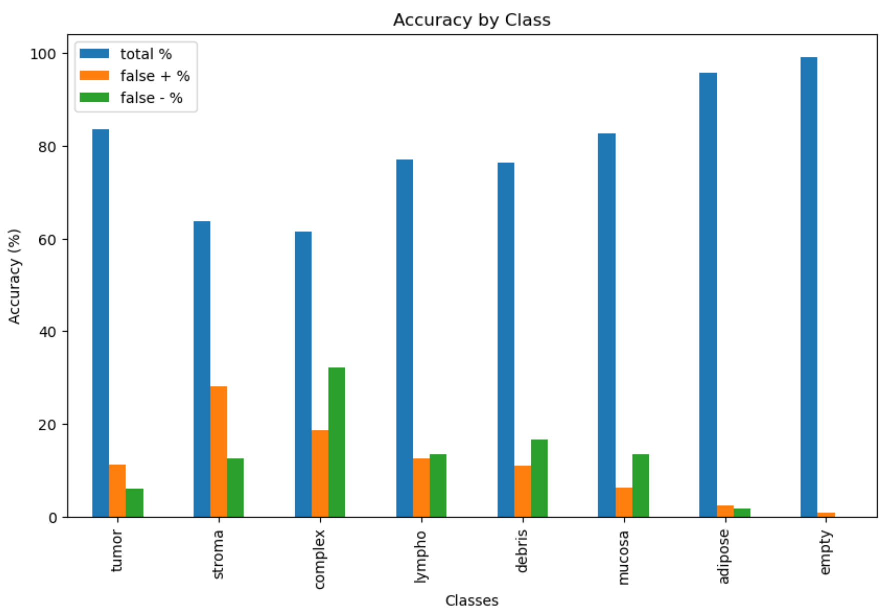

Figure 12. Transfer Learning class accuracy.

Concerns / Future Suggestions

I think there are a few more methods that can improve the accuracy of the model. One potential addition would be using GANs or other adversarial networks to improve both the ‘stroma’ and ‘complex’ classifications. Another issue is the dataset size of 5000. More data or a better image augmentation strategy (using techniques closely related to this field) would be needed to improve the accuracy even further. Another area of interest is improving / tuning the ‘learning rate’ and other parameters, which I did not have much time to explore with this project.

References

[1] Kather, Jakob & Weis, Cleo-Aron & Bianconi, Francesco & Melchers, Susanne & Schad, Lothar & Gaiser, Timo & Marx, Alexander & Zöllner, Frank. (2016). Multi-class texture analysis in colorectal cancer histology. Scientific Reports. 6. 27988. 10.1038/srep27988.

[2] Ali, Luqman & Alnajjar, Fady & Jassmi, Hamad & Gochoo, Munkhjargal & Khan, Wasif & Serhani, Mohamed. (2021). Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. Sensors. 21. 1688. 10.3390/s21051688.