DERM Dataset notes

Cropped all 6,000 images of benign and malignat. Focused on center of image and reduced 60-70% around the center, where the cells were.

Did this because of the noise that was taking place during the validation, when we ran our feature visualizations , noticed that it was taking datapoints from “noise” in the image, such as a ruler in the bottom, or hair and other objects surrounding the cell that was not to be used for prediction and modeling. Since we were just focused on the cell itself.

Cropping allowed a much greater reduction in noise, but did cut off the cell in some cases where the imag did not need to be cropped. But this was worth it in saving time because it was much easier to crop every image instead of manually going through and evaluating each and every image. Saving a lot of time.

Ran SIFT on the cropped images

Ran SFT feature visualizations, much better results! No rulers in the image, and a lot less noise.

Onto the next steps of running 7 models

1. SVM
2. KNN
3. Random Forest
4. Decision Tree
5. XGBoost
6. CNN (We are required to use this one)

LTSM BONUS 7?

RESULTS

**SVM**

Accuracy: 0.7491888774063303

Classification Report:

precision recall f1-score support

0 0.34 0.33 0.34 7999

1 0.84 0.85 0.85 33610

accuracy 0.75 41609

macro avg 0.59 0.59 0.59 41609

weighted avg 0.75 0.75 0.75 41609

**KNN**

Accuracy: 0.8355163546348146

Classification Report:

precision recall f1-score support

0 0.59 0.49 0.53 7999

1 0.88 0.92 0.90 33610

accuracy 0.84 41609

macro avg 0.73 0.70 0.72 41609

weighted avg 0.83 0.84 0.83 41609

**Decision Tree**

Accuracy: 0.8315508663990964

Classification Report:

precision recall f1-score support

0 0.63 0.31 0.41 7999

1 0.85 0.96 0.90 33610

accuracy 0.83 41609

macro avg 0.74 0.63 0.66 41609

weighted avg 0.81 0.83 0.81 41609

**XGBoost**

Accuracy: 0.846523588646687

Classification Report:

precision recall f1-score support

0 0.74 0.31 0.44 7999

1 0.86 0.97 0.91 33610

accuracy 0.85 41609

macro avg 0.80 0.64 0.68 41609

weighted avg 0.83 0.85 0.82 41609

**LSTM**

Classification Report:

precision recall f1-score support

0 0.79 0.84 0.82 620

1 0.82 0.76 0.79 583

accuracy 0.80 1203

macro avg 0.80 0.80 0.80 1203

weighted avg 0.80 0.80 0.80 1203

**CNN**

Training the CNN model...

Epoch 1/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 184s 1s/step - accuracy: 0.6872 - loss: 0.5492 - val\_accuracy: 0.8015 - val\_loss: 0.4118

Epoch 2/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 151s 1s/step - accuracy: 0.8025 - loss: 0.4160 - val\_accuracy: 0.6050 - val\_loss: 0.7099

Epoch 3/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 145s 1s/step - accuracy: 0.8327 - loss: 0.3857 - val\_accuracy: 0.8420 - val\_loss: 0.3464

Epoch 4/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 121s 999ms/step - accuracy: 0.8773 - loss: 0.2992 - val\_accuracy: 0.8721 - val\_loss: 0.3029

Epoch 5/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 121s 1s/step - accuracy: 0.8871 - loss: 0.2875 - val\_accuracy: 0.8753 - val\_loss: 0.2958

Epoch 6/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 127s 1s/step - accuracy: 0.9034 - loss: 0.2590 - val\_accuracy: 0.9085 - val\_loss: 0.2598

Epoch 7/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 125s 1s/step - accuracy: 0.9007 - loss: 0.2479 - val\_accuracy: 0.9064 - val\_loss: 0.2445

Epoch 8/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 131s 1s/step - accuracy: 0.9099 - loss: 0.2207 - val\_accuracy: 0.9168 - val\_loss: 0.2379

Epoch 9/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 120s 988ms/step - accuracy: 0.9264 - loss: 0.2095 - val\_accuracy: 0.8919 - val\_loss: 0.2815

Epoch 10/10

121/121 ━━━━━━━━━━━━━━━━━━━━ 149s 1s/step - accuracy: 0.9276 - loss: 0.1980 - val\_accuracy: 0.9075 - val\_loss: 0.2548

38/38 ━━━━━━━━━━━━━━━━━━━━ 10s 271ms/step - accuracy: 0.9150 - loss: 0.2194

Test Accuracy: 0.916043221950531

38/38 ━━━━━━━━━━━━━━━━━━━━ 12s 302ms/step

Classification Report:

precision recall f1-score support

0 0.87 0.99 0.92 620

1 0.98 0.84 0.91 583

accuracy 0.92 1203

macro avg 0.93 0.91 0.92 1203

weighted avg 0.92 0.92 0.92 1203