Skin Cancer Classification using MNIST: HAM10000

Project for **CS6553** Deep Learning Data:HAM10000

Project Member Team Names

Abdias Baldiviezoaguilar, Sam Himes, Caleb Cranney

We were inspired by Mahbod et al. to used pertained models to used multiple pre-trained models to generate feature maps that would be fed into our classifier. However, in there work, they included an additional step of training an SVM classifier at the end of each pre-trained model. We, on the other hand, simply concatenated our feature maps and fed them into our fully connected classifier. It would have been interesting to train classifiers for each pre-trained model, and then combine them to get a prediction.

Additionally, it would be interesting to test a variety of combinations of pre-trained networks. Due to time constraints we could only combine DenseNet and ResNet, but it would be interesting to see if combinations of other pre-trained networks faired better or worse.

Contributing

Any kind of enhancement or contribution is welcomed.

Background: Project Hard Pivot

We originally proposed as our course project to generate a deep learning model for basecalling raw nanopore sequencing data. We investigated and attempted to run the SACall repository as a starting point, as they employ transformers, an element of deep learning all project members are interested in. As a backup, we attempted to run bonito, the standard Oxford Nanopore basecaller, for generating models. In both cases, we ran into a number of errors over the course of a week and never succeeded generating a deep learning model using raw data. As such, we requested approval for changing our project to what is listed above.

Acknowledgments

We thank Mathew Kouch, the author of the original notebook that we started from. We used this notebook as a starting point, and, from there, made our modifications to the input data and pertained models.

Initially our project was related to nanopore-basecalling, after many hours of work and debug, we reconsidered and shifted our focus to a another area: Skin Cancer Detection/Classification.

Requirements

- yacs (Yet Another Configuration System)
- PyTorch (An open source deep learning platform)
- ignite (High-level library to help with training neural networks in PyTorch)

Table Of Contents

Project Background

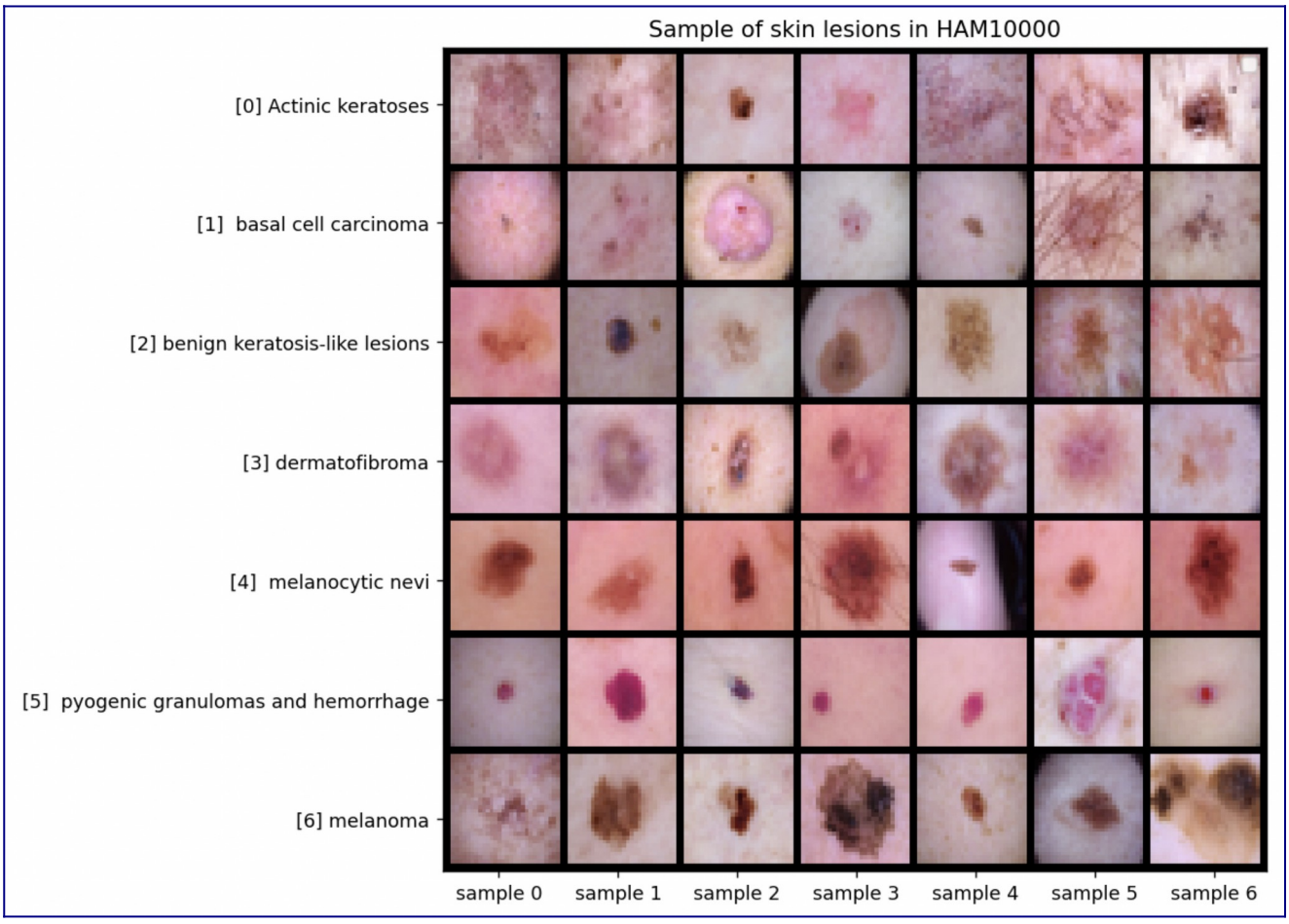
Skin Cancer is an extremely prevalent form of cancer. In the US, about 9,500 people in the US are diagnosed with skin cancer every day. When detected early, patients with skin cancer have an extremely high survival rate. Steps should be taken to improve the accessibility of early detection. With an accurate deep learning model, patients could take pictures of their own skin abnormalities and detect cancer early. We aim to modify a deep learning model to more accurately classify cancer.

Data Description

This dataset contains 10015 dermatoscopic images of pigmented lesions for patients in 7 diagnostic categories. For more than half of the subjects, the diagnosis was confirmed through histopathology and for the rest of the patience through follow-up examinations, expert consensus, or by in-vivo confocal microscopy. More information about the dataset and the diagnosis categories, features and patience conditions besides the links to download the dataset can be found on either Harvard Dataverse or on Kaggle.

The categories include; Actinic keratoses and intraepithelial carcinoma / Bowen's disease (AKIEC), basal cell carcinoma (BCC), benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, BKL), dermatofibroma (DF), melanoma (MEL), melanocytic nevi (NV)

vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, VASC). Of these categories, 3 are cancerous (BCC, AKIEC, MEL) and 4 are non-cancerous (BKL, DF, VASC, NV).



Model Experimentation Description

We approached our model generation from 2 different model architectures with 2 different data preparations, resulting in 4 different model runs.

Model Architectures

1. densnet121 - We investigated how other programmers had addressed this problem in the past. We found the following notebook, which details using the densenet121 model for lesion type identification using this dataset. The model came with pretrained data, though the classifier step was altered to be trained specific to the dataset.
2. densenet121/resnet18 hybrid - We found the following paper which outlines the hybridization of pre-trained models for identifying lesion types. Following this pattern, we combined the densenet121 and resnet18 pretrained models using the same classifier modification step as model architecture 1.

Data Preparation

1. Categorization - the data is originally divided into the 7 categories listed above. We trained the model architectures to categorize a photo by skin lesion type.
2. Binary - we determined which of the 7 categories were considered malignant and which were considered benign (sometimes labelled in the code as "cancer" and "nonCancer," respectively) and set up the models for binary classification between these two classes instead of all seven.

Model Notebook Links

1. densnet121, Categorization
2. densenet121/resnet18 hybrid, Categorization
3. densnet121, Binary
4. densnet121/resnet18 hybrid, Binary

Expected Data Structure after downloading from source

```
./archive
├── HAM10000_metadata.csv
├── hmnist_28_28_L.csv
├── hmnist_28_28_RGB.csv
├── hmnist_28_28_RGB_binary.csv
├── hmnist_8_8_L.csv
├── hmnist_8_8_RGB.csv
└── lesion_images
    └── all_images [10015 entries]
```

Results

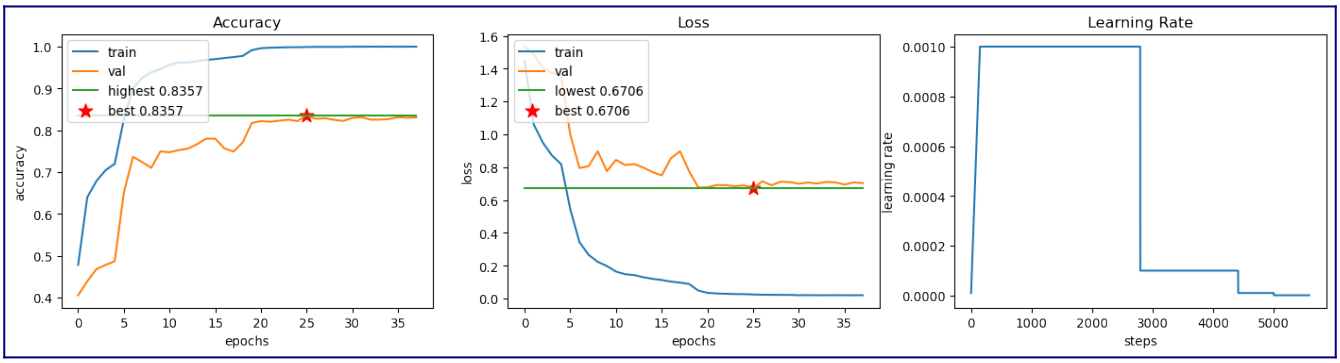
These were the final accuracies for each of the runs.

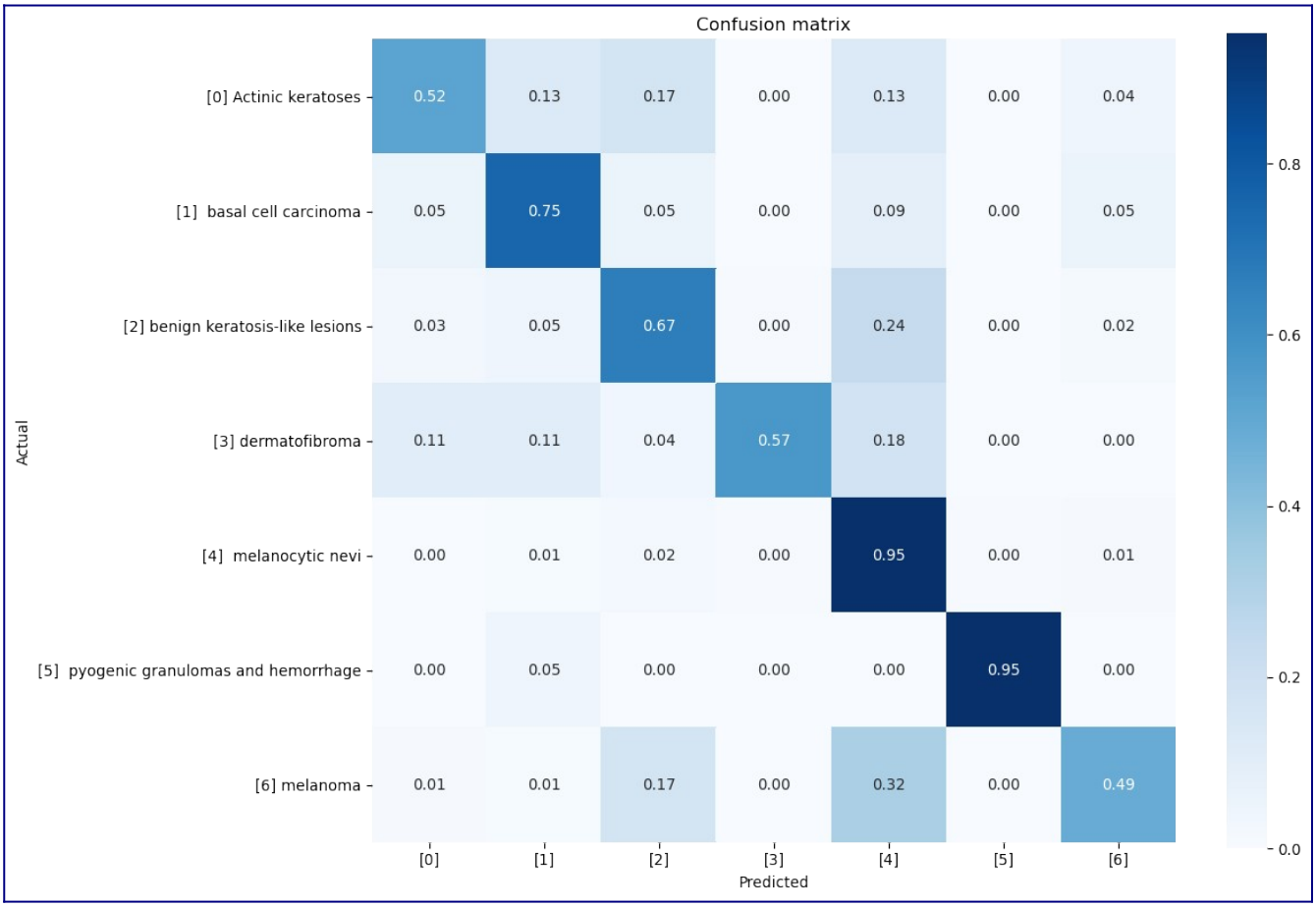
- DenseNet, multi class: 0.8248
- Mix, multi class: 0.8208
- DenseNet, binary: 0.8867
- Mix, binary: 0.8877

It appears that changing the problem to a binary classification improves the accuracy of the classifier. This is encouraging, as the primary intent of our program is to reduce the likelihood for mistaking a non-cancerous lesion for a cancerous lesion generally. While identifying the specific subtype and implied danger level is important, this may be best confirmed by a medical professional who can recommend proper medical treatment.

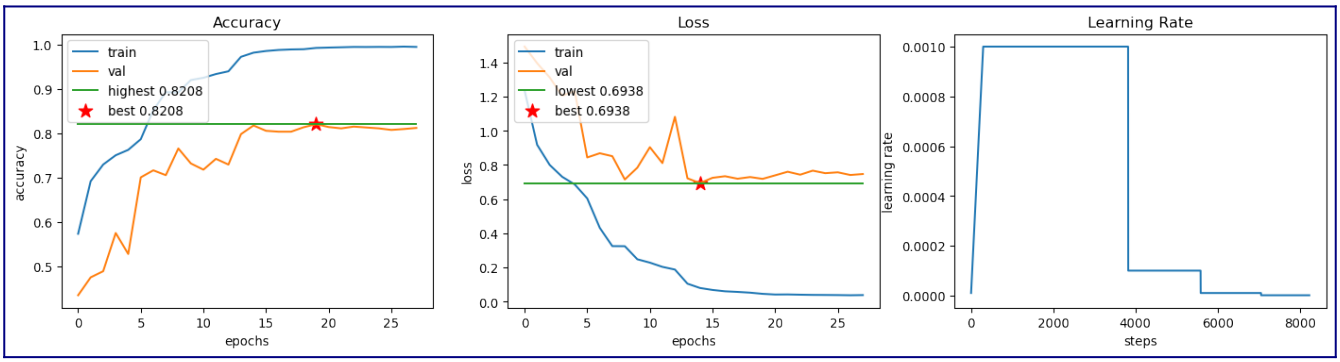
Furthermore, it appears that mixing ResNest and DenseNet pre-trained classifiers does not significantly alter the accuracy for either binary or multi class categorization. See Future Works for potential future considerations for improving the accuracy of these models.

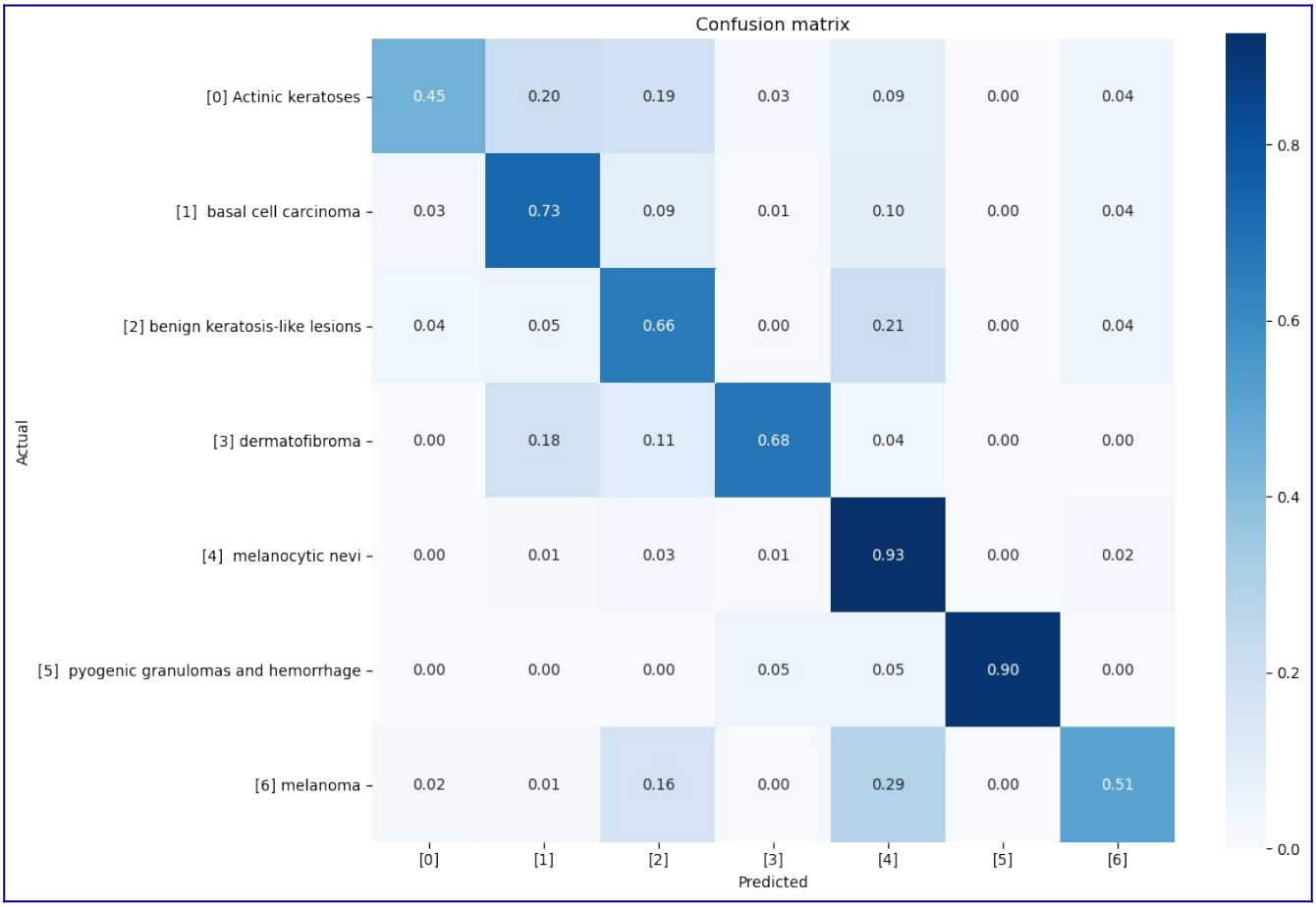
Multi Class DenseNet (Original)



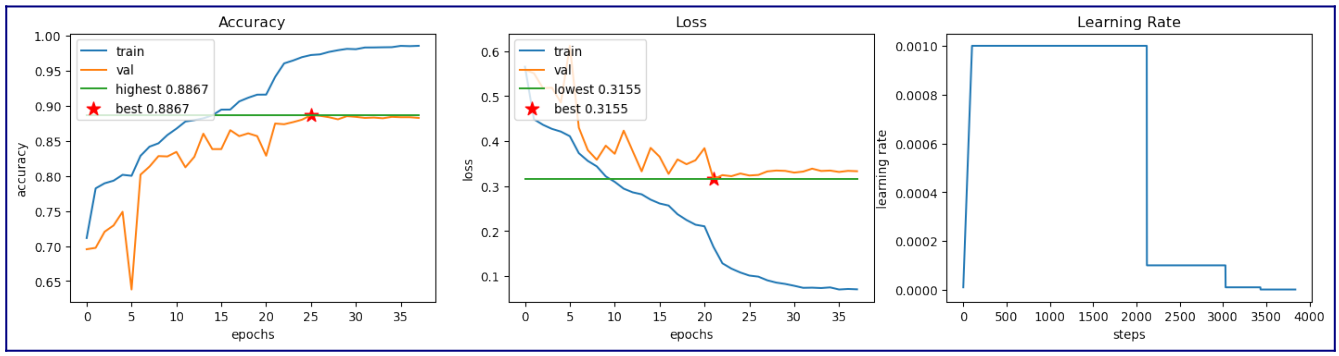


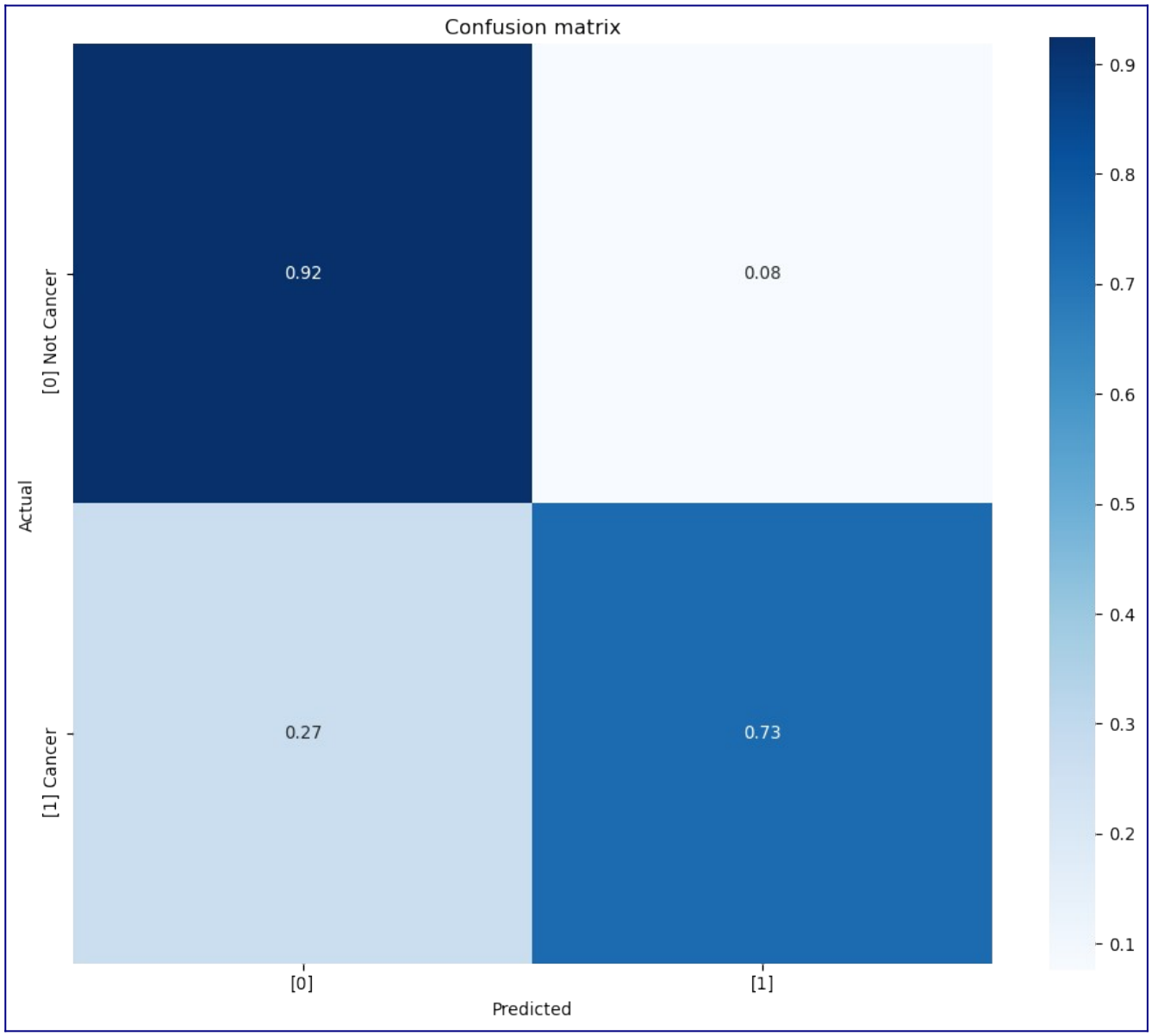
Multi Class Mix



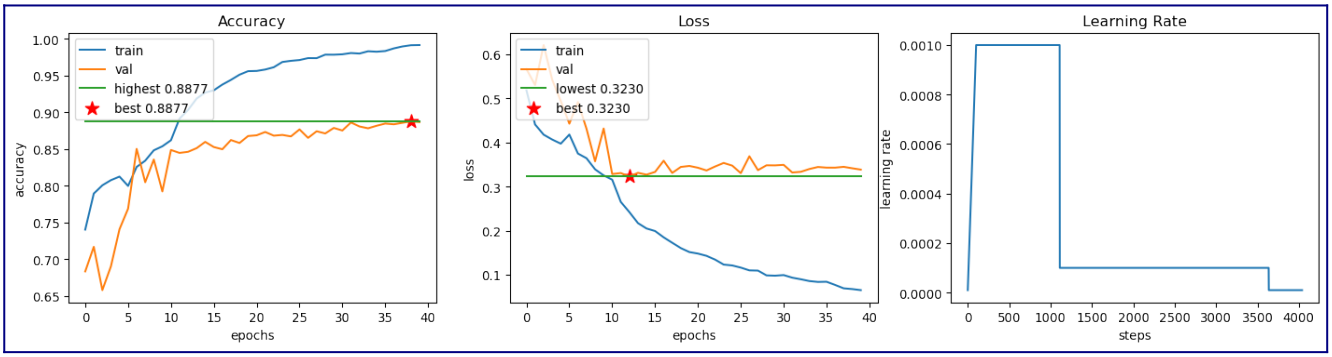


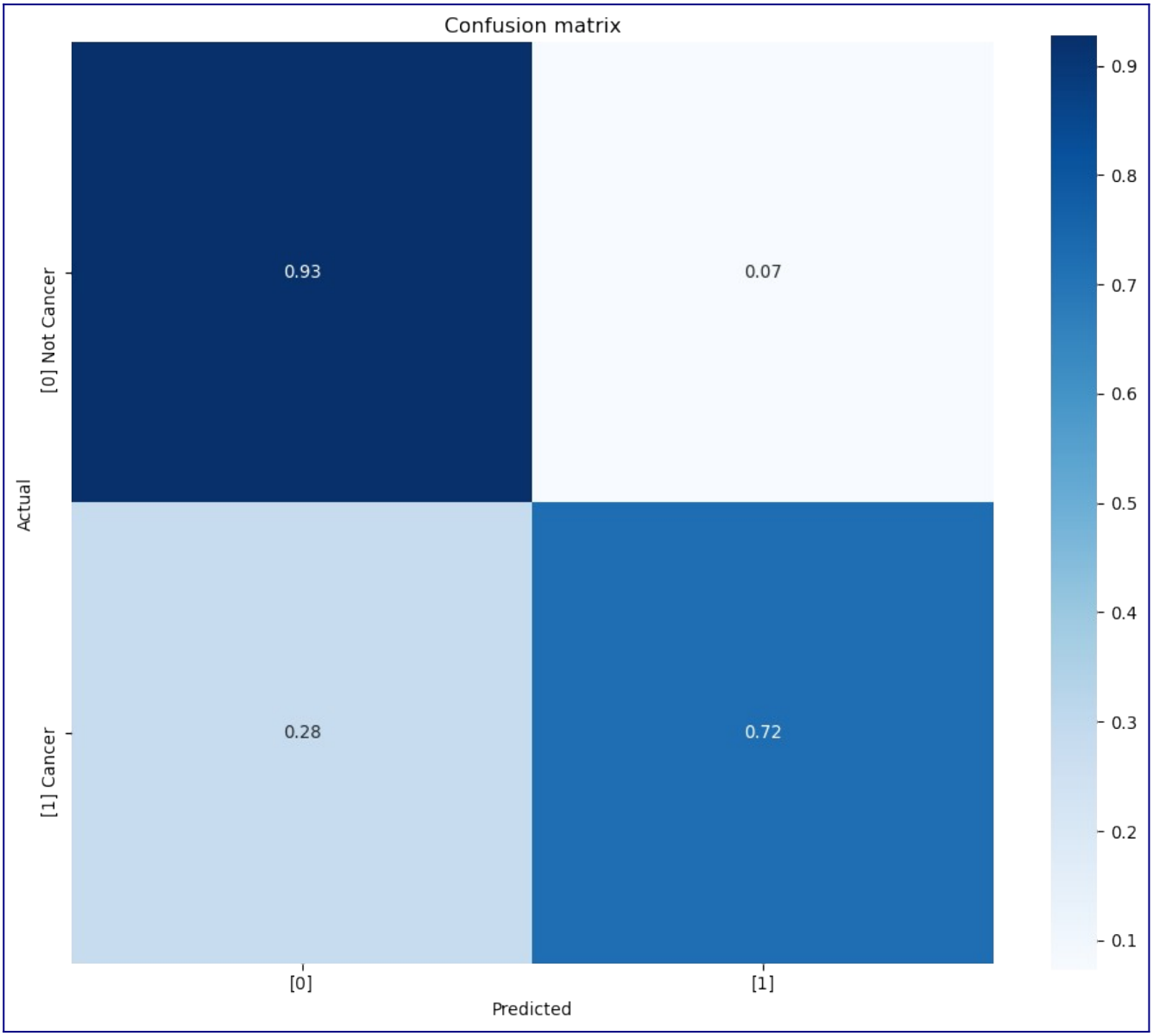
Binary DenseNet





Binary Mix


Accuracy
train
val
highest 0.8877
best 0.8877
accuracy
epochs
Loss
train
val
lowest 0.3230
best 0.3230
loss
epochs
Learning Rate
learning rate
steps


Confusion matrix
0.93
0.07
0.28
0.72
[0] Not Cancer
[1] Cancer
Actual
[0]
[1]
Predicted

Future Work

We were inspired by Mahbod et al. to used pertained models to used multiple pre-trained models to generate feature maps that would be fed into our classifier. However, in there work, they included an additional step of training an SVM classifier at the end of each pre-trained model. We, on the other hand, simply concatenated our feature maps and fed them into our fully connected classifier. It would have been interesting to train classifiers for each pre-trained model, and then combine them to get a prediction.

Additionally, it would be interesting to test a variety of combinations of pre-trained networks. Due to time constraints we could only combine DenseNet and ResNet, but it would be interesting to see if combinations of other pre-trained networks faired better or worse.

Contributing

Any kind of enhancement or contribution is welcomed.

Background: Project Hard Pivot

We originally proposed as our course project to generate a deep learning model for basecalling raw nanopore sequencing data. We investigated and attempted to run the SACall repository as a starting point, as they employ transformers, an element of deep learning all project members are interested in. As a backup, we attempted to run bonito, the standard Oxford Nanopore basecaller, for generating models. In both cases, we ran into a number of errors over the course of a week and never succeeded generating a deep learning model using raw data. As such, we requested approval for changing our project to what is listed above.

Acknowledgments

We thank Mathew Kouch, the author of the original notebook that we started from. We used this notebook as a starting point, and, from there, made our modifications to the input data and pertained models.