Multiple Instance Learning for Genetic Mutation Detection in Histopathology

Sebastian Steiner
sss2327@columbia.edu

1 Introduction

30-40% of breast cancer patients have a PIK3CA mutation [Owkin 2023], which causes cells to divide and replicate uncontrollably. Patients with this mutation are resistant to endocrine therapy, but respond well to PI3Kα inhibitor therapy [Owkin 2023]. Therefore, detecting this mutation and personalizing treatment can significantly improve patient outcomes.

After surgery or biopsies, the micro-structures of human tissue can be examined for disease via histopathology analysis. Through this analysis, experts in DNA sequencing are able to identify tumor genotypes and consequently genetic mutations, such as the PIK3CA mutation. Unfortunately, there are limited experts in DNA sequencing, making these examinations difficult to access. Hence, a model that can automatically analyze histopathology slides and correctly identify the genetic mutations would streamline the process of determining the optimal cancer therapy and improve treatment.

Histopathology slides are stored via whole slide images (WSI), which can have dimensions of up to 200,000 pixels by 150,000 pixels. Along with the fact that medical datasets are typically small (in this case the development dataset has 344 samples), training a convolutional neural network (CNN) on each image is impractical as it will simply overfit. Instead, 1000 tiles of size 248x248 were extracted from each WSI and will be used as inputs to the model.

A consequence of this approach is that the problem has now become weakly-supervised because in healthy tissue samples, all tiles will appear healthy; whereas in diseased samples most tiles will appear healthy with only a handful containing the mutation. Therefore, a significant step in the model will be developing a method to extract 'a needle from a haystack', with the needle being the diseased tile in this case. From a machine learning point of view, this is a multiple-instance learning (MIL) problem because each sample bag (i.e., the image) is labeled, but the instances in each bag (i.e., the tiles of each image) have different labels.

Another aspect to highlight is the fact that the WSI data is not annotated. That is, the diseased tissue has not been localized. While the use of annotated data would improve model performance, it is difficult to find experts for disease localization. Therefore, models are trained without annotations as it is more scalable.

In this project, 3 models are explored: a logistic regression model, a CHOWDER model and a hybrid model. The logistic regression approach acts as a baseline, a vanilla CHOWDER model [Courtiol 2018] is then implemented. This model is then tuned and modified for this specific breast cancer dataset. Finally, an investigation into a hybrid approach that uses CHOWDER embeddings along with standard metadata is performed.

2 Data

2.1 Data Source

The WSI data is provided by the National Cancer Institute in their TCGA-BRCA project. This data has been processed by Owkin and published on their PIK3CA mutation detection challenge page [Owkin 2023]. For the challenge, 344 samples are provided for model development and 76 samples are used for testing [1] [2].

2.2 Selecting Tiles

1000 tiles with dimensions 248x248 have been chosen by Owkin from each WSI. They have ensured that tiles are only selected inside the tissue edges and not the background. This is achieved by converting the WSI into HSV color space [3]. Then Otsu's method [Otsu 1979], which automatically separates images into foreground (i.e., tissue) and background via threshold, is used to produce 2 masks in the hue and saturation channels. The results from the 2 masks are combined for the segmentation of tissue and background.

The tissue in histopathology slides is stained to make the tissue stand out. However, the stain intensity varies across slides as different colors and dyes are used. Therefore, Owkin has rescaled the tissue samples to normalize the color channels [4].

Finally, a grid with a coordinate spacing of 248x248 pixels is overlayed on the tissue in each WSI. Then a uniform random sampling of 1000 non-overlapping grids has been taken and these grid boxes have been used as the tiles for the challenge. See Figure 1 for a visualization of the tiling process.

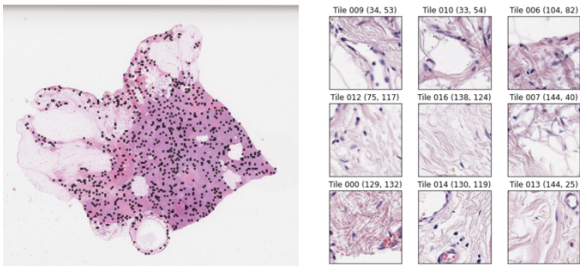

Figure 1: *The left shows a histopathology slide with the chosen tiles marked as the black grids. The right zooms in to show some of the tiles in the slide. These images are taken from the Owkin challenge page [Owkin 2023].*

2.3 Feature Extraction

Owkin also provides the results of their feature extraction from the tiles. In their feature extraction step, they pre-train a Wide ResNet-50-2 architecture [Zagoruyko and Komodakis 2016] on the TCGA-COAD dataset and then use this model to extract MoCo V2 features [Chen et al. 2020] from each tile. Breaking this down, a Wide ResNet-50-2 architecture is a variant of the ResNet architecture, which is a popular CNN model for feature extraction in computer vision tasks. In Wide ResNet-50-2, the 50 denotes the number of layers in the network and the 2 represents the doubling of channels in each convolution layer compared to the original ResNet. The TGCA-COAD dataset is also provided by the National Cancer Institute and contains genomic data on colon adenocarcinoma cancer.

[1] The test labels are not accessible

[2] Actually the test set contains 149 samples, but the AUC score is only made public for 76 samples and kept private for the other samples. As there is no access to the private AUC, this project will act as if only 76 samples are used for testing

[3] HSV stands for Hue, Saturation, Value. It is an alternative color space to RGB. For more information refer to the following link

[4] See [Courtiol 2018] for more details on this process

MoCo V2 stands for Momentum Contrast Version 2 and is a self-supervised method. It has been trained to extract features from the Wide ResNet-50-2 architecture and produces its own feature vector from each tile with 2048 dimensions. See Figure 2 for the extracted features across the 1000 tiles in one of the samples.

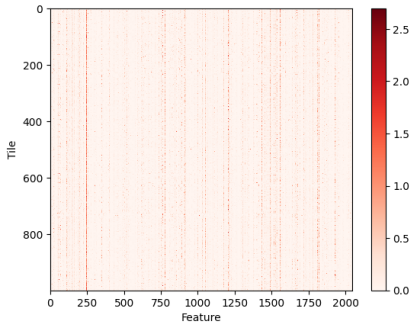

Figure 2: *After tiling and feature extraction, the sample data has this form: 1000 rows for tiles and 2048 columns for features.*

2.4 Data Split

There are 344 samples in the development dataset and the testing set has 76 samples. The test labels are not accessible, only the AUC score (Area Under Curve) of the predictions. The development data has a healthy-diseased split of 63-37 and it is assumed that the testing data has a similar split. Given the data imbalance, the AUC metric is preferred over accuracy. Also, the AUC is a ranked-based metric, which means that the analysis is focused on the primary goal of scoring samples with the mutation over samples without the mutation evaluated, rather than a downstream goal of tuning the raw output probability and classification threshold probability.

Some metadata is included in the dataset, such as the center ID where the sample was collected, the patient ID and the zoom level of the histopathology slide. The data was collected from 5 centers, 3 of which are in the development set with the other 2 being in the test set. Therefore, using the center ID as a feature would not generalize well. Mapping patient IDs to samples revealed that some patients contributed multiple samples to the dataset. As genomic data is unique to each patient, using multiple samples from each patient could lead to the model overfitting on patients, hence only one sample per patient is used during model development leading to 305 samples in the development set [5].

In terms of reporting results, the AUC is reported twice: once on the validation data and then on the test performance from Owkin's challenge page.

3 Logistic Regression

Introduction - Logistic regression is a common model for binary classification as it is simple to implement. In Logistic Regression, a weighted sum of the inputs is taken and passed through a sigmoid function that maps the logits to a number between 0 and 1, which can be viewed as a probability of being a positive sample. The results of logistic regression are to be used as baseline scores for the more advanced models that shall be explored.

$$\sigma(z) = \frac{1}{1+e^{-z}}$$
$$\text{where:} \quad z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + c$$

Method - Each sample has 1000 tiles and 2048 features per tile, so a sample has dimensionality $X_i \in \mathbb{R}^{1000\times 2048}$. For each sample, the MoCo features of each tile are summed together, resulting in $\hat{X}_i \in \mathbb{R}^{1000\times 1}$. Then 4 different processing strategies are implemented to train 4 different logistic regression models:

- Sample Average: all $\hat{X}_i$ are averaged, hence $x \in \mathbb{R}^1$
- Max Tile: $max(\hat{X}_1, \hat{X}_2, ..., \hat{X}_{1000})$, hence $x \in \mathbb{R}^1$
- Top 5 Tiles: $max_5(\hat{X}_1, \hat{X}_2, ..., \hat{X}_{1000})$, hence $x \in \mathbb{R}^5$
- All Tiles: All $\hat{X}_i$'s are used, hence $x \in \mathbb{R}^{1000}$

In each approach, the data is standardized and split into 5 stratified folds. Therefore, 5 models are trained and validated on the left-out fold, see Table 1 for the average AUC on the validation data. Each model also predicts the test results, the average test prediction from the 5 models is put forward to the challenge page and these AUC scores can also be found in Table 1.

In every model, C=0.01, where C is the inverse of regularization strength. Setting C to a small value imposed strong regularization to prevent overfitting through the convergence to a simpler decision boundary. Also, a liblinear solver is used because it is effective on small datasets with high-dimensionality, like this dataset.

Results and Discussion - In Table 1, all validation scores are higher than the test scores, indicating that there is overfitting during model development. This could be due to the samples coming from new centers in the test set, indicating the undesirable effect of implicitly fitting on the sample center. Another point to highlight is that the validation scores are slightly better than random chance and the test scores are worse than random chance (except for the Max Tile method). A reason for this is that logistic regression cannot capture the complex relationships between features in the data. Supporting this is the fact that the standard deviations of prediction probabilities in the test set range from 0.015-0.055, indicating that the model clusters predictions close together because it lacks certainty in distinguishing positive samples from negative samples. In all, these results suggest that a simple logistic regression model is not sufficient for this classification task.

Pooling Method	Val AUC	Test AUC
Sample Average	0.58	0.43
Max Tile	0.55	0.52
Top 5 Tiles	0.56	0.48
All tiles	0.56	0.43

Table 1: *Validation and Test AUC results from Logistic Regression with different tile pooling strategies.*

4 CHOWDER Method

4.1 Base

Background - The CHOWDER model was introduced by [Courtiol 2018] as a general model that could perform MIL to detect genetic mutations in histopathology images. It showed promising results on the TGCA-Lung dataset provided by the National Cancer Institute and the Camelyon-16 dataset. Now, its performance shall be evaluated on the TCGA-BRCA dataset.

The data preparation for CHOWDER follows the same flow outlined in Section 2, hence the model expects each sample to have 2048 features extracted from each of the 1000 tiles.

Structure - First, a 1D convolution filter with size 2048 and stride 2048 is applied. This acts as a **feature embedding layer** because the dimensionality of the samples go from 1000x2048 to 1000x1. For reference, in logistic regression, the 2048 features were summed, now the features are a weighted sum. As the size of the training data is $<$ 1000 samples, the advice of [Courtiol 2018]

[5] Refer to Appendix A for Exploratory Data Analysis Results

is followed to embed to a latent space of 1 dimension as it reduces the chance of overfitting. Also, L2-regularization is applied on this layer to prevent overfitting, with the regularization parameter set to 0.5.

Next, the 1000 embedded tiles are sorted. The top 2 tiles are used as top instances and the bottom 2 tiles are used as negative evidence, forming a **MinMax layer**. The benefits of including negative evidence are that it encourages the model to be more discriminative as the model must be able to differentiate between positive and negative tiles. Also, it reduces model bias because the model must be able to extract information from positive and negative tiles.

Finally, there is an **MLP classifier** with 2 hidden layers having 200 nodes and 100 nodes, respectively. The goal of these layers is to capture the complex interactions between top instances and negative evidence. All nodes have a sigmoid activation and to prevent overfitting include 0.5 dropouts. See Figure 3 for a diagram of the model's structure.

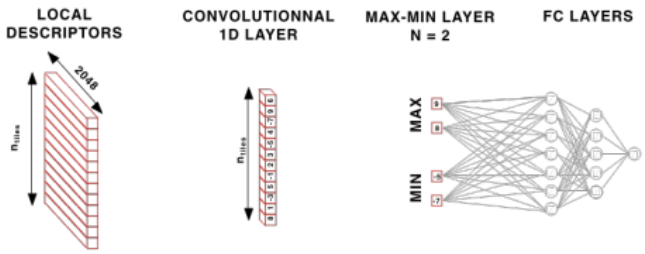

Figure 3: *CHOWDER architecture found in [Courtiol 2018]. In this case,* $n_{tiles} = 1000$.

A binary cross-entropy loss is used with an Adam optimizer with a learning rate of 0.001. Other parameters include batch sizes of 10 and 30 epochs. To further prevent overfitting, 10 CHOWDER models are trained and their ensemble average prediction is used, with the only difference amongst the models being their initial weights.

One difference in this implementation of CHOWDER is that tiles have multiple zooms. In the original implementation, all tiles had the same zoom; however, in this dataset, samples are taken at different zooms, as selecting only 1 zoom level would further reduce the number of samples and result in no predictions for a few samples in the test set.

In terms of data split, all models are trained on 85% of the development set, with the remaining 15% used in validation. Similar to the logistic regression model, AUC results are recorded for the validation and test sets.

Results and Discussion - Table 2 shows the results of the base CHOWDER model with a validation AUC of 0.50, which is worse than the validation AUC from logistic regression, and a test AUC of 0.54, which is better than the test AUCs in logistic regression. These results are not significant enough to make any claims that the complex CHOWDER model is better than the simple logistic regression.

4.2 Dimensionality Reduction

While the pre-processing by Owkin has resulted in a large dimensionality reduction, the development dataset still has 304 samples and 2048 features, making overfitting a significant obstacle in this project. Creating a custom feature extraction model that leads to fewer features from each tile would be ideal. This would, however, significantly increase the computing cost. As it already takes 1 hour to train each version of the CHOWDER model, developing a custom feature extraction method is not practical. Looking into dimensionality reduction has been explored; however, it was found that results significantly decreased. For example, PCA was used to halve the number of dimensions, but the decrease in AUC indicates that the low varying features may be the keys to detecting the presence of the mutation. Given the computing constraints and the poor initial results of dimensionality reduction methods, models will aim to prevent overfitting by simplifying the models and applying regularization strategies.

4.3 Weight Initialization

The ensembling of CHOWDER models reduces the variance of prediction and the chance of overfitting. However, reducing the prediction variance also has a drawback because the output results are less decisive as the probabilities are more likely to be clustered together. The fact that each time this complex CHOWDER model is trained, it is treated like a weak learner (such as those found in Random Forests) is an interesting concept, which may be worth investigating in the future.

In any case, these ensembled CHOWDER models are trained in the same way with the only difference being their weight initializations. Unfortunately, [Courtiol 2018] does not specify how these weights are randomly initialized. In the base model, weights were set by PyTorch's default settings [Paszke et al. 2017], where the weight is chosen from a uniform distribution between -1 to 1. However, to explore the effects of initial weights, a random uniform is applied between $-|E|^{-0.5}$ to $|E|^{-0.5}$, where $|E|$ denotes the number of edges in the layer. Weights are also selected from a random normal distribution with mean 0 and standard deviation $|E|^{-0.5}$. The weight initialization variance is connected to the number of edges to reduce the effect of initial conditions because there are few samples to train on, so choosing small weights that are initially close together might reduce overfitting. In both approaches the bias is always set to 0.

Both approaches have been evaluated on the validation data. The uniform strategy resulted in a validation AUC of 0.43 and the normal strategy resulted in a validation AUC of 0.55. As the random normal initialization yields a better AUC on the validation compared to the base and alternative uniform initialization, it will be used moving forward in the subsequent model variations.

4.4 Pooling and Architecture

The CHOWDER method uses a novel structure. For example, the MinMax layer contrasts top instances with negative evidence. Usually, models simply use max-pooling or mean-pooling. Mean pooling would not be a good choice because the goal of this model is to find the few tiles with mutation and to make the prediction. Taking the average across tiles does not contribute to finding the few mutated tiles; instead, it mixes the tiles together, making the classification task more difficult. Additionally, a good point that [Courtiol 2018] raised is that MinMax pooling allows users to retrace the top instances. Therefore, even if the model is not successful at classifying samples, transfer learning could be applied to locate tiles that are most likely to have the mutation in the sample. Simply locating these tiles in a 150,000 by 200,000 pixels image could speed up the process for the expert DNA sequencers.

The MLP classifier has an input of 4 nodes (from the MinMax layer), followed by 2 hidden layers with 200 nodes and 100 nodes, respectively. These hidden layers aim to capture complex interactions between the top instances and negative evidence. However, this significantly increases the number of trainable parameters, increasing the chance of overfitting. Therefore, to evaluate the effectiveness of the MinMax layers and the MLP, CHOWDER's performance is assessed after modifications to its structure. To see these modifications, refer to Appendix B, which contains the validation results in Table 4.

Through experimentation of the architecture, it has been found that feeding the outputs of the MinMax layer into a logistic regression model (i.e, a linear layer with a sigmoid activation) leads to a validation AUC score of 0.72, which is significantly higher than other architectural modifications. This model also achieves a 0.60 AUC score on the test data.

It is generally accepted that a 'good' model achieves AUCs scores of at least 0.6, where 'good' refers to better than definitively better than random chance [6]. The modifications to the CHOWDER model have shifted it from arguably better than random prediction to definitely better than random prediction. As this improvement comes from simplifying the model, that is replacing the MLP with 200 nodes and 100 nodes in its hidden layers to a logistic regression classifier, one could argue that the model was overfitting before. Moving forward, this model architecture will be used.

4.5 Scaling

The result of Owkin's feature extraction is that each tile is described by 2048 MoCo V2 features, which are not human-interpretable. This project assumes that each dimension has the same scale and distribution. Therefore, features are scaled using the global average and global standard deviation across all dimensions, tiles and samples:

$$x_{scaled} = \frac{x - x_{mean}}{x_{stdev}}$$

However, [Courtiol 2018] does not mention scaling in their CHOWDER model. Therefore, the base CHOWDER was trained without scaling.

Table 2 shows the results of the CHOWDER model without scaling. While the test AUC is 0.6, which is the same as the with scaling, the validation AUC is 0.51. As the AUC score on the scaled data is better on the validation set and the same on the test set, scaled data will continue to be used.

4.6 Hyperparameter Tuning

The hyperparameters suggested by [Courtiol 2018] are not guaranteed to be the global optimal hyperparameters because their optimization is data-dependent. The following hyperparameters are tuned: learning rate, batch size, number of epochs and L2-regularization parameter.

Each parameter is evaluated at the suggested value and an alternative value, resulting in the evaluation $2^4 = 16$ models. It would have been preferred to try more values and more hyperparameters for tuning[7]. However, each model takes around 1 hour to tune and limited compute resources are available, so the extra tuning is not possible.

The results on the validation data for each set of parameters are shown in Appendix C. It turns out the optimal set of hyperparameters are those used prior to tuning. That is, batch sizes of 10, learning rate of 0.001, 20 epochs and L2-regularization parameter of 0.1.

[6] An AUC of 0.5 is equivalent to random guessing, with random predictions coming from the positive and negative class distributions. For AUC scores between 0.5-0.6, the model is scoring better than random chance, but in this region, this improvement is not statistically significant. Models with AUC scores greater than 0.6 are better than random models with a degree of statistical significance

[7] Tuning these hyperparameters on different network architectures, with-/without scaling and different weight initialization was also desired. However, this would have significantly increased the resources needed for computing. So, tuning was done linearly as an approximation.

CHOWDER	Val AUC	Test AUC
Base	0.50	0.54
Random Normal Weight Init	0.62	0.55
Modified Architecture	0.72	0.60
Without Scaling	0.51	0.60
Tuned Hyperparameters	0.72	0.60

Table 2: *Validation and Test AUC results from the CHOWDER model and its modifications. The hyperparameters used before tuning are found to be the optimal hyperparameters, hence the AUC scores are the same.*

5 Hybrid Model

The hybrid model is inspired by wide and deep neural networks, which are commonly used in recommender systems. The wide component of the model aims to capture important low-order input interactions and the deep component aims to capture higher-order input interactions. There is an integration layer that combines the outputs of each layer and the output of the integration layer is the output of the model.

In this case, the deep component of the model is the feature embedding and MinMax layers. The wide component includes metadata about the tiles. The hypothesis of this approach is that by including the wide component, the logistic regression model can achieve better classification performance.

5.1 Basic Feature Engineering

Zoom - As mentioned, the original CHOWDER model was trained and evaluated on samples from 1 zoom level. Unfortunately, samples come from different zooms in this dataset and this heterogeneity may cause decreased performance. To address these differences, the zoom of each of the 4 tiles is included as input to the logistic regression model.

Tile Coordinates - Some genetic mutations are highly-localized. That is, tiles with the mutation are close together. From the literature, it is unclear whether the PIK3CA mutation is highly localized. Therefore, metadata about the tile coordinates are included in the model to test this hypothesis that genetic mutations are highly localized.

As the tiles were taken at different zooms, the same tile could have different coordinates at different zooms due to the change in resolution. The exact relationship between 2 zoom settings is unclear, making it impossible to convert the coordinates to a common reference frame. While the zoom settings of tiles chosen for each sample have not been formally analyzed, it seems that in the majority of samples, all tiles put forward are at the same zoom, meaning the coordinates are in the same reference frame. Moreover, in some tests, both the zoom level and coordinates are included, which could lead to the model learning how to relate coordinates at different zooms.

Results and Discussion - The results from Table 6 in Appendix D show that the AUC score is at least 0.60 in all but one of the tests. The best AUC result used only the coordinates from all selected tiles. Therefore, this setup was put forward to be used on the test set and achieved an AUC score of 0.56. The performance on the test data is lower than the validation, therefore the model will be tuned to try and yield more stable results.

5.2 Advanced Feature Engineering

Due to the simplifications of model architecture (an embedding layer, followed by MinMax pooling and logistic regression classification), it is more difficult to model the features of higher-order interactions. Therefore, more complex feature engineering is tested

to try and make up for the lack of the model's ability to capture higher-order interactions.

Feature Ratio - After the MinMax layer selects the top instance and negative evidence tiles, the features of the top instances are averaged together, and the features from the negative evidence are averaged. The ratio of these averages is then put forward as a feature.

$$Ratio = \frac{AverageTopInstanceFeatureScore}{AverageNegativeEvidenceFeatureScore}$$

Top Instance Distance - To help test the hypothesis that genetic mutations are highly localized, the Euclidean distance between the two top instance tiles is used as a feature.

Results and Discussion - The AUC results on the validation data with the feature ratio is 0.50 and with the top instance distance, it is 0.29. Given these poor results, further experiments involving advanced features will not be carried out. For example, tests using basic and advanced metadata have not been completed. Instead, efforts will be focused on tuning the model that uses all coordinates.

5.3 Hyperparameter Tuning

Due to the changes in model inputs and architecture, hyperparameter tuning has been repeated. The following hyperparameters are tuned: number of epochs, learning rate and L2 regularization parameters (for the convolution and linear layers).

In the original CHOWDER model, the MLP used dropout for regularization. In the simplified architecture only the 4 tiles were passed to the logistic regression classifier, so no regularization was imposed. However, since the coordinates add 8 extra inputs to the classifier, a second L2-regularization parameter has been included in the linear layer to prevent overfitting.

Also, the custom weight initialization strategy, which chooses initial weights from a normal distribution with variance dependent on the number of edges in the layer, has been substituted for Xavier weight initialization [Glorot and Bengio 2010]. This initialization strategy is similar to the custom approach because it also assigns the weights based on the number of inputs and outputs at each layer (which is related to the number of weights in the layer) and it ensures that the variance of the weights is similar across layers. The Xavier approach has been tested and shown to improve model performance and stability, therefore moving forward this initialization will be used [8].

Hybrid Model	Val AUC	Test AUC
All Coordinates	0.68	0.56
Tuned Hyperparameters	0.66	0.43

Table 3: *Validation and Test AUC results from the hybrid model with basic feature engineering and hyperparameter tuning.*

Results and Discussion - The tuning process has worsened the process. One reason for this could be the change in weight initialization method from custom to Xavier. However, it is believed that more likely causes include model instability and small evaluation datasets. Addressing the former, model performance drastically changes with slight changes to the architecture and hyperparameters, indicating the model is not stable because slight changes to the model result in large performance changes. Considering the evaluation datasets, the validation set only contains 40 samples [9] and the test 76 samples. Therefore, it is difficult to conclude how models compare because the AUC score has large variability simply due to the small evaluation dataset size. Combining these two sources of variability provides an explanation as to why model performance has been so volatile and consequently why it has been difficult to effectively tune the models.

6 Decisiveness Assessment

Every model evaluated in this project is a binary classifier, determining whether a sample has (or does not have) a genetic mutation. To measure the model's performance, the AUC has been used, which is a rank-based metric that measures how good the classifier is at predicting the positive class (samples with genetic mutations) above the negative class (samples without the genetic mutation). Due to the limited sizes in the dataset, it has been difficult to meaningfully quantify the AUC of models, therefore Figures 4 and 5 have been included to visualize the model performance.

A good binary classifier is one that is decisive. That is, the classifier must rank positive samples above negative samples (which is what the AUC measures) and has a clear prediction boundary (the critical point at which predictions change from positive to negative). The latter is implicitly measured because models with a soft prediction boundary perform worse than those with hard prediction boundaries. If one were to plot the predictions of a good binary classifier, one would expect to see a cluster of positive samples with a high score and a cluster of negative samples with a low score. While the cluster score could have some variance, positive samples should not have lower scores than negative samples (this is where the prediction boundary comes into play). Further, as the subspace of predictions is [0,1], it is expected that the cluster's separation and distance from the prediction boundary is on the scale of the prediction boundary. While this is not a necessary condition, it demonstrates the model's ability to leverage the entire prediction subspace.

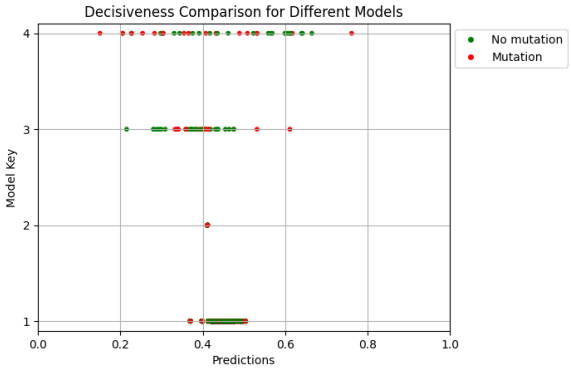

Figure 4: *Model predictions on the validation set. Red dots indicate that the sample contains the mutation, hence the true label would be 1. Green dots represent no genetic mutation in the sample, hence the true label would be 0. The model key mappings are: 1 = Max Tile Logistic Regression, 2 = Base CHOWDER, 3 = Modified CHOWDER, 4 = Hybrid. As the Max Tile Logistic Regression is trained using cross-validation, every sample in the training set has been predicted, whereas the other models only make 40 predictions.*

Figure 4 shows the distribution of predictions for different models. Recalling the AUC results, model 1 presents Max Tile Logistic regression and achieves 0.55 AUC. Model 2 is the base CHOWDER and scores 0.50 AUC. Model 3 is the modified and tuned CHODWER, which has an AUC of 0.72. Model 4 is the hybrid model

[8] I only just found out about this initialization strategy in class and did not want to re-run all my previous tests.

[9] The validation dataset actually has 45 samples; however, the model has been setup to process the data in batches of 10, so the last 5 samples are not validated. I should have modified the setup to adjust for the changes in batch size for more accurate model development.

which scores 0.68. The amount of mixing of positive and negative samples in this figure visually explains why the AUC is fairly low for all the models. Further, the worse-performing models use less of the sample space, which reinforces the requirements that the predictions should be well-spaced in the prediction space, so the prediction boundary is 'stronger'.

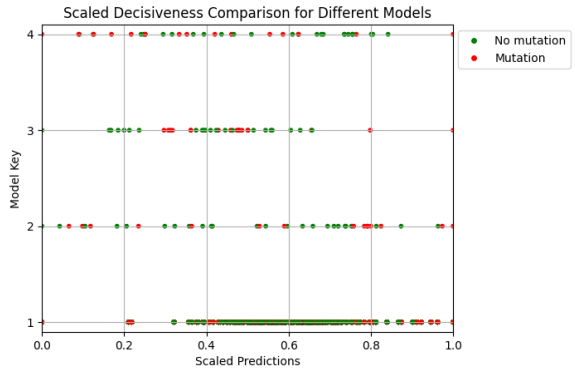

Figure 5: *Scaled model predictions on the validation set (all predictions are scaled between the min and max prediction).*

Figure 5 shows the MinMax scaled predictions, which are more aligned with how the AUC metric would view the data given its rank-based nature. The same points about sample mixing and weak prediction boundaries hold. Furthermore, in models 1 and 4, mutated samples are both the min and max predictions, which highlights the indecisiveness of the models. Nevertheless, a positive result in this figure is the clustering of samples in model 3 (and less so in model 2). As mentioned, a good classifier clusters the predictions of samples and has a clear prediction boundary. In these models, the data is beginning to cluster, so only the next step of separating the data on a prediction boundary is required (albeit this next step of classifier development is non-trivial).

7 Closing Remarks

Several deep learning models have been implemented to perform the task of detecting PIK3CA mutations from histopathology slides. The main model that this paper replicates is Owkin's CHOWDER model, which has been successful with other datasets involving genetic mutation detection. The creators of the CHOWDER model claim it is generalizable across similar datasets; however, its results were poor on this dataset as the model only achieved 0.50 and 0.54 AUCs on the validate and test data, respectively. These scores are equivalent to random guessing and indicate that the model is not as generalizable as its authors suggest.

Nevertheless, modifications to the CHOWDER were made and a variation achieved AUCs of 0.72 and 0.60 on the validate and test data, respectively. This variation uses an embedding layer for each tile followed by a MinMax pooling layer to select 4 tiles and finally a sigmoid layer to make the prediction. This model uses trade-offs between the logistic regression baselines, as it only uses a sigmoid layer without an MLP, and the CHOWDER model, as the embeddings are trainable and MinMax pooling is used. While the AUC scores show potential, more work into model development must be carried out because the predictions have high stakes. That is, these predictions would determine the type of cancer therapy that may save a person's life, so detecting the mutation must be performed with a high degree of certainty and trustworthiness, which this model does not provide.

Further work into developing the model should look to acquiring larger datasets for model development as the small dataset has posed limitations and liabilities to training and overfitting. Implementing a unique feature extraction algorithm for this dataset, rather than using transfer learning from models trained on other types of data, could also improve performance. Finally, looking into the differences in data acquisition differences across centers and understanding the properties of the mutation from a biological point of view could also provide domain-specific knowledge for model development insights.

In all, the models developed in this project have been largely underwhelming because this is a challenging machine learning task that is weakly-supervised and uses a small dataset. However, as the field of deep computer vision models has a lot of potential and automating this process would have a significant impact on patient's lives, there is optimism that better models for this task will be developed.

References

CHEN, X., FAN, H., GIRSHICK, R. B., AND HE, K. 2020. Improved baselines with momentum contrastive learning. *CoRR abs/2003.04297*.

COURTIOL, P. 2018. Classification and disease localization in histopathology using only global labels: A weakly-supervised approach.

GLOROT, X., AND BENGIO, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 249–256.

OTSU, N. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics 9*, 1, 62–66.

OWKIN, 2023. Detecting PIK3CA mutation in breast cancer. https://challengedata.ens.fr/participants/challenges/98/.

PASZKE, A., GROSS, S., CHINTALA, S., CHANAN, G., YANG, E., DEVITO, Z., LIN, Z., DESMAISON, A., ANTIGA, L., AND LERER, A. 2017. Automatic differentiation in pytorch.

ZAGORUYKO, S., AND KOMODAKIS, N. 2016. Wide residual networks. *CoRR abs/1605.07146*.

A Exploratory Data Analysis

Figures 6-8 show distributions of metadata in the training and test sets. Figure 6 reveals there is a fairly even distribution of samples from each center and that the samples in the test set come from different centers than those in the training set. Figure 7 shows that the fraction of samples with the genetic mutation is fairly consistent across the centers in the training set. Figure 8 shows that most samples in the training and test sets are taken at zoom level 16, with a few also taken at 14, 15 and 17.

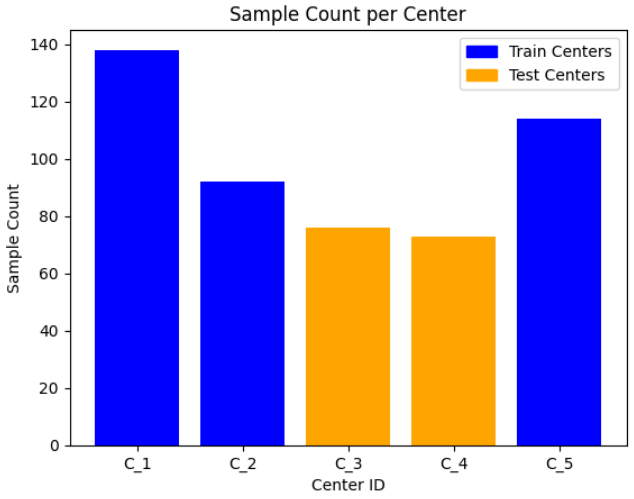

Figure 6: *The distribution of samples from each center. The centers used in the training data are different from those used in the test data.*

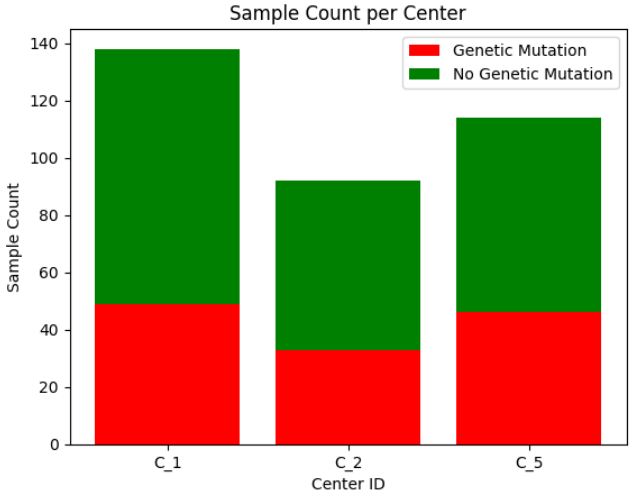

Figure 7: *The distribution of samples per center in the training dataset with breakdowns of the number of samples that have the PIK3CA genetic mutation.*

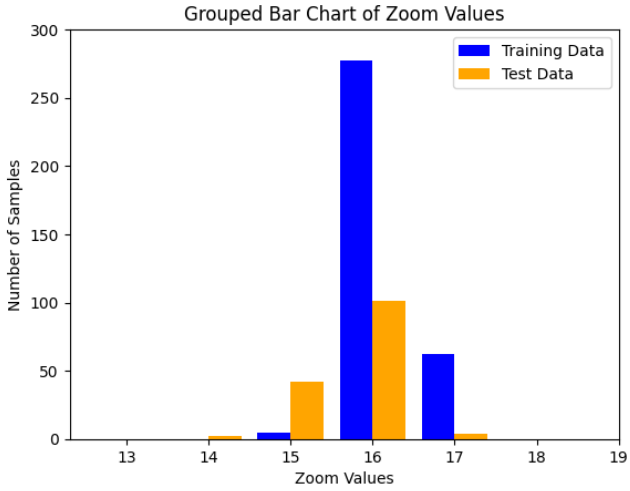

Figure 8: *The distribution of zooms in the training and test data.*

B Pooling and Architecture Results

Different modifications to the CHOWDER architecture are tested to explore whether performance would improve.

Model Architectures:

- MinMax Layer returns the top 5 and bottom 5 tiles followed by the base MLP classifier.
- A max-pooling layer chooses the tile with the largest feature embedding and passes it to a logistic regression classifier.
- A max-pooling layer chooses the 5 tiles with the largest feature embeddings and passes them to a logistic regression classifier.
- The base MinMax layer is used followed by logistic regression classification.
- The base MinMax layer is used followed by an MLP with 1 hidden layer that has 4 nodes.
- The base MinMax layer is used followed by an MLP with 2 hidden layers that have 4 nodes in each layer.

Pooling	Classifier	Val AUC
MinMax 5	Base MLP	0.43
Max	Logistic Regression	0.51
Max 5	Logistic Regression	0.51
Base MinMax	**Logistic Regression**	**0.66**
Base MinMax	MLP 1 hidden layer	0.53
Base MinMax	MLP 2 hidden layers	0.51

Table 4: *Validation AUC results from architectural modifications to the CHOWDER model. In these models, the weights are initialized using the random normal initialization. Apart from the MinMax 5 + Base MLP, these models use 12 epochs (as they are simpler than the CHOWDER, so 30 epochs leads to overfitting) and the L2-regularization parameter is dropped to 0.1. For the models with base MinMax and modified hidden layers, the probability of dropout is reduced to 0.1.*

The best-performing model on the validation set uses a MinMax layer that picks the top 2 and bottom 2 tiles from the embedding layer and passes these through a logistic regression classifier. This model is used on the test set, see Table 2 for results. This model is also re-evaluated on the validation set with 20 epochs, instead of 12. This change leads to the validation AUC increasing to 0.72.

C Hyperparameter Tuning Results

16 models are trained and evaluated on the validation set with different hyperparameter combinations. The best set of values uses a learning rate of 0.001, batch size of 10, 20 epochs and an L2-regularization parameter of 0.1. This model has been also evaluated on the test set.

Learning Rate	Batch Size	Epochs	L2-param	Val AUC
0.001	**10**	**20**	**0.1**	**0.72**
0.001	10	20	0.5	0.57
0.001	10	30	0.1	0.42
0.001	16	20	0.1	0.57
0.001	10	30	0.5	0.52
0.001	16	20	0.5	0.47
0.001	16	30	0.1	0.56
0.001	16	30	0.5	0.47
0.0001	10	20	0.1	0.63
0.0001	10	20	0.5	0.58
0.0001	10	30	0.1	0.55
0.0001	16	20	0.1	0.57
0.0001	10	30	0.5	0.53
0.0001	16	20	0.5	0.43
0.0001	16	30	0.1	0.35
0.0001	16	30	0.5	0.33

Table 5: *Modified CHOWDER model performance on the validation data when tuning the learning rates, batch size, number of epochs and L2-regularization parameter.*

D Hybrid Model Results

For the hybrid model, the performance when different metadata is included is evaluated on the validation set. Table 6 shows the results when basic metadata is included, such as microscope zoom, all tile coordinates and top instance tile coordinates.

The best-performing model uses all coordinates (without zoom). Therefore, hyperparameter tuning is carried out on the validation set and the results are shown in Table 7.

Zoom	All Coordinates	Top Coordinates	Val AUC
✓	X	X	0.63
X	✓	**X**	**0.68**
X	X	✓	0.60
✓	✓	X	0.60
✓	X	✓	0.48

Table 6: *AUC scores on the validation data for different combinations of zoom and coordinate features. The All Coordinates column refers to tests where coordinates from all tiles from the Min-Max layer are included. The Top Coordinates column refers to tests where only coordinates from the top instances are included.*

Learning Rate	Epochs	L2 Conv	L2 Linear	Val AUC
0.0001	20	0.01	0.0001	0.42
0.0001	20	0.01	0.001	0.55
0.0001	20	0.01	0.01	0.56
0.0001	**20**	**0.1**	**0.0001**	**0.66**
0.0001	20	0.1	0.001	0.40
0.0001	20	0.1	0.01	0.62
0.0001	30	0.01	0.0001	0.52
0.0001	30	0.01	0.001	0.63
0.0001	30	0.01	0.01	0.46
0.0001	30	0.1	0.0001	0.49
0.0001	30	0.1	0.001	0.28
0.0001	30	0.1	0.01	0.58
0.001	20	0.01	0.0001	0.38
0.001	20	0.01	0.001	0.41
0.001	20	0.01	0.01	0.58
0.001	20	0.1	0.0001	0.63
0.001	20	0.1	0.001	0.45
0.001	20	0.1	0.01	0.53
0.001	30	0.01	0.0001	0.64
0.001	30	0.01	0.001	0.60
0.001	30	0.01	0.01	0.48
0.001	30	0.1	0.0001	0.52
0.001	30	0.1	0.001	0.46
0.001	30	0.1	0.01	0.48

Table 7: *Hybrid model performance on the validation data when tuning the learning rates, number of epochs and L2-regularization parameters (for the convolution and linear layers).*