**Capstone 2: Milestone Report 2**

**Problem Statement**

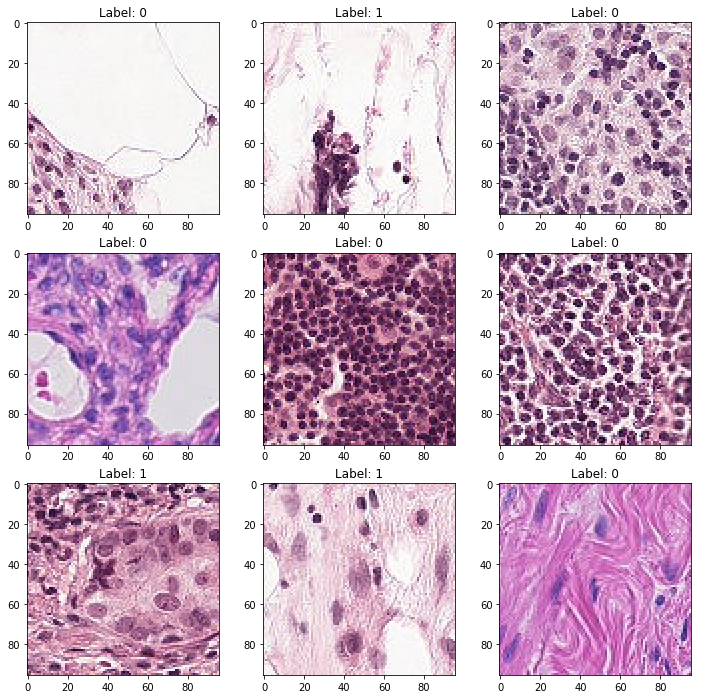
With the increases in cases of cancer, early detection and treatment is of vital importance as it relates to reduction in mortality rate among certain groups in the population. This project focuses on identifying metastatic tissue in histopathologic scans of lymph node sections.

**Dataset Overview**

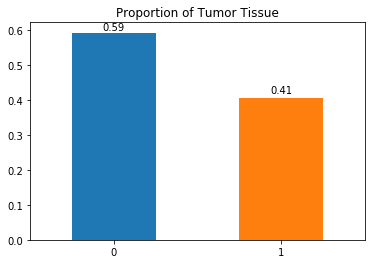
Data was obtained from the Kaggle competition, Histopathologic Cancer Detection. The dataset can be obtained here, www.kaggle.com/c/histopathologic-cancer-detection

The folder contains 277,483 images (7.41 GB) of small pathology images of which 220,025 are for training and 57,458 for testing. Each image has a resolution of 96 x 96 px, however only the center 32x32 px region is used for this binary classification as having at least one pixel of tumor tissue.

Each image varies in cell density, shape, size, and staining.



Along with the images the folder contains a csv file that contains the labels for the training data. Simple visualization reveals that 90,210 (41%) of the training images contain tumor cells in the center 32x32 px region.



**Methodology**

This project presents a large dataset of images which will results in a long training time. Great effort must be expended in pre-processing and system design to feed the images to the network.

The aim of the project is to classify images into two classes, therefore this task requires the use of Deep Convolutional Neural Networks.

To effectively train the model we will use a large number of epochs using GPU to increase the processing speed. Different hyperparameter values will be applied to the model along with the addition of multiple layers. The train set images will be divided into a training set (75%) and a validation set (25%).

The images are all stored in one folder with a csv file containing the names and labels. The images will be feed to the model using the flow\_from\_dataframe() function.

**Model**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv2d\_1 (Conv2D) (None, 94, 94, 32) 896

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 92, 92, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 46, 46, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_3 (Conv2D) (None, 44, 44, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_4 (Conv2D) (None, 42, 42, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_5 (Conv2D) (None, 40, 40, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 20, 20, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 20, 20, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 51200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 128) 6553728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_2 (Dropout) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pred (Dense) (None, 1) 129

=================================================================

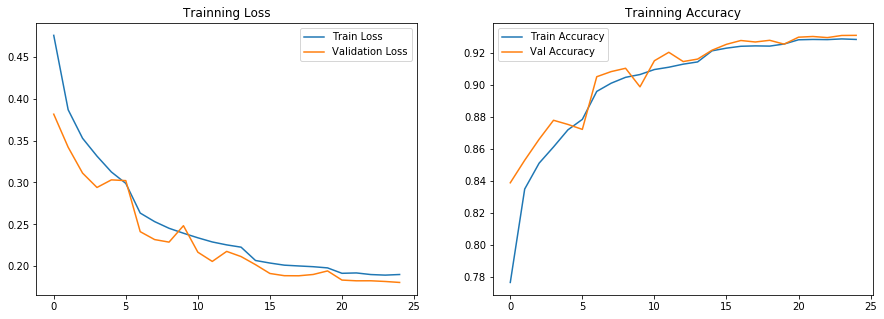
Total params: 6,831,617

Trainable params: 6,831,617

Non-trainable params: 0

**Results**

In the image augmentation step multiple image transformations were applied to the training set images to effectively identify the target feature.



**Discussion**

At the beginning of the process CPU was used to train and evaluate the images and each epoch took approximately 5 hrs to complete. After tapping into the gaming GPU on the computer the time was reduced to ~30 mins per epoch. Callbacks where effect in increasing the accuracy and decreasing the loss, if there was no improvement in two successive epoch the learning rate was adjusted, and the subsequent epoch showed improvement. It would be interesting to see how additional image manipulations affect accuracy.