# Report on Histopathologic Cancer Detection

Guanbo Bian

## Problem Description

PCam is highly interesting for both its size, simplicity to get started on, and approachability. The dataset comprises numerous small pathology images, each identified by an image ID. The ground truth for the images in the training folder is provided in the train_labels.csv file. The objective is to predict the labels for the images in the test folder. A positive label indicates that the central 32×32 pixel region of a patch contains at least one pixel of tumor tissue. Notably, tumor tissue in the outer region of the patch does not influence the assigned label.

In this Kaggle project, we will employ binary classification to categorize images as cancerous (class label 1) or benign (class label 0). Specifically, the task involves identifying metastatic cancer in small image patches extracted from larger digital pathology scans.

PCam simplifies the clinically-relevant challenge of metastasis detection into a straightforward binary image classification task, which is executed by a Convolutional Neural Network (CNN) model.

# Data Exploration

There are totally 220025 training images and 57458 testing images.

The label files for training and testing are 1 or 0 for each image. The first 10 training sample labels are shown below:

| | id | label |
|---|---|---|
| 0 | f38a6374c348f90b587e046aac6079959adf3835.tif | 0 |
| 1 | c18f2d887b7ae4f6742ee445113fa1aef383ed77.tif | 1 |
| 2 | 755db6279dae599ebb4d39a9123cce439965282d.tif | 0 |
| 3 | bc3f0c64fb968ff4a8bd33af6971ecae77c75e08.tif | 0 |
| 4 | 068aba587a4950175d04c680d38943fd488d6a9d.tif | 0 |
| 5 | acfe80838488fae3c89bd21ade75be5c34e66be7.tif | 0 |
| 6 | a24ce148f6ffa7ef8eefb4efb12ebffe8dd700da.tif | 1 |
| 7 | 7f6ccae485af121e0b6ee733022e226ee6b0c65f.tif | 1 |
| 8 | 559e55a64c9ba828f700e948f6886f4cea919261.tif | 0 |
| 9 | 8eaaa7a400aa79d36c2440a4aa101cc14256cda4.tif | 0 |

Fig. 1 First 10 sample classes for training datasets

Each image can be represented by a array in dimension f by f by $n_c$, where f is the height and width, and $n_c$ is the number of channels. For example, the first image can be represented by 96 by 96 by 3, 3 represents R, G and B.

The table below shows the distribution of the dataset, in terms of different classes.

Table 1. Distribution of images in dataset.

| Magnification | Benign | Malignant | Total |
|---|---|---|---|
| 40x | 652 | 1370 | 1995 |
| 100x | 644 | 1437 | 2081 |
| 200x | 623 | 1390 | 2013 |
| 400x | 588 | 1232 | 1820 |
| Total | 2480 | 5429 | 7909 |

The plots below show how the cancer and non-cancer images look like.

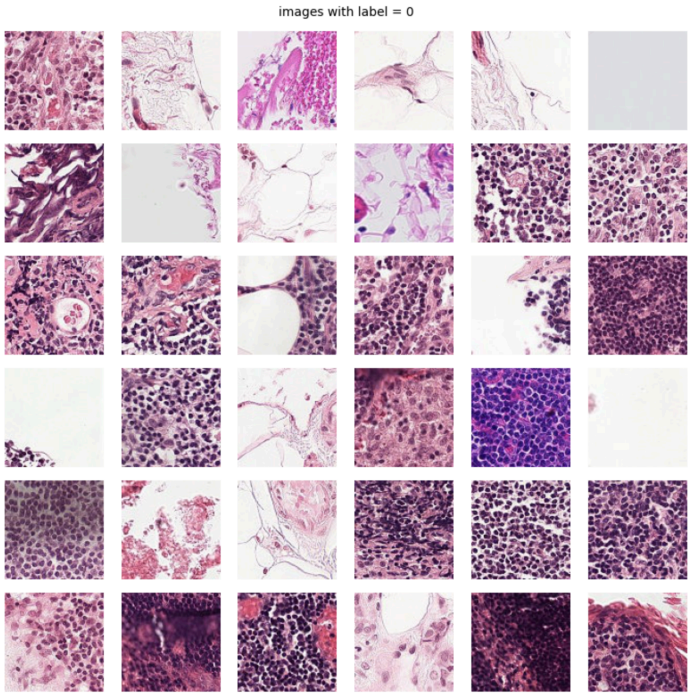



Fig. 2 (a) images without cancer

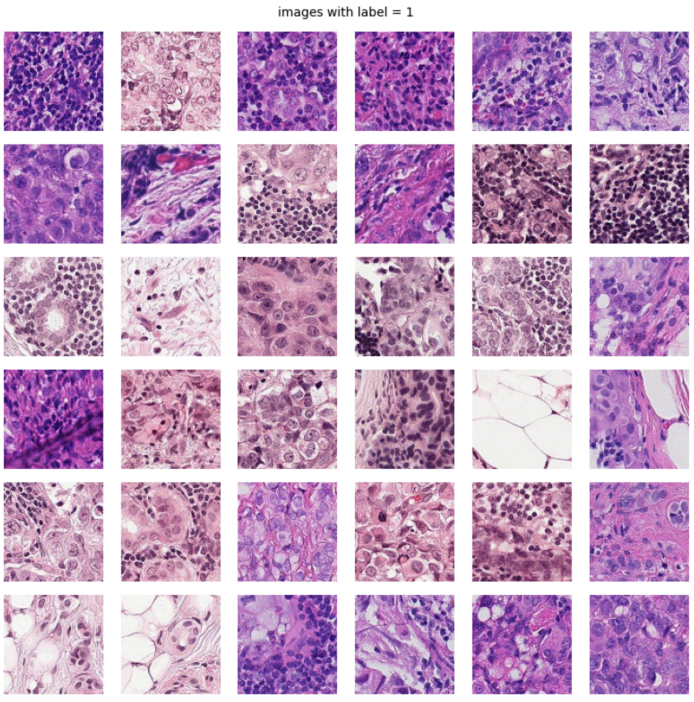



Fig. 2 (b) images with cancer

## Model Architecture

We will implement baseline models using Keras model subclassing, incorporating configurable and reusable blocks to facilitate hyperparameter adjustments. Both model creation approaches, model subclassing with reusable blocks and functional APIs, will be employed.

There are two models which I worked on in this exercise. The first one is the CNN traditional model and the second one is a more advanced version called Resnet50.

The baseline models will follow a traditional CNN architecture, consisting of Convolution/Pooling blocks designed to achieve translation invariance, increase the field of view, and reduce the number of parameters. Subsequently, the models will include flatten, Dense, and a binary classification layer.

The Convolution Block will be implemented through the ConvBlock class, encompassing two Conv2D layers of the same size followed by a MaxPool2D layer. Optionally, a BatchNormalization layer, aimed at reducing internal covariate shift between batches, can be included immediately after each Conv2D layer. The filter size, kernel size, activation function, pooling size, and the presence of batch normalization will all be configurable, with default values, allowing specification during block instantiation.

The next reusable block, termed TopBlock, will comprise a Flatten layer followed by two dense layers, culminating in a classification layer with a single output neuron. The size of the dense layer will be customizable. Default ReLU non-linear activation will be applied to layers, except for the final classifier layer, which will use sigmoid activation to constrain the output to the range [0−1]. This output range can be interpreted as the probability of an image containing tumor cells.

The hyperparameters for batch_size and im_size are adjustable parameters in our model. We will tune and modify these values accordingly. Specifically, a batch size of 256 will be utilized, and all images will undergo resizing to dimensions of 64×64. This resizing is implemented to conserve memory resources during the model training process.

The picture below shows the base model CNN architecture.

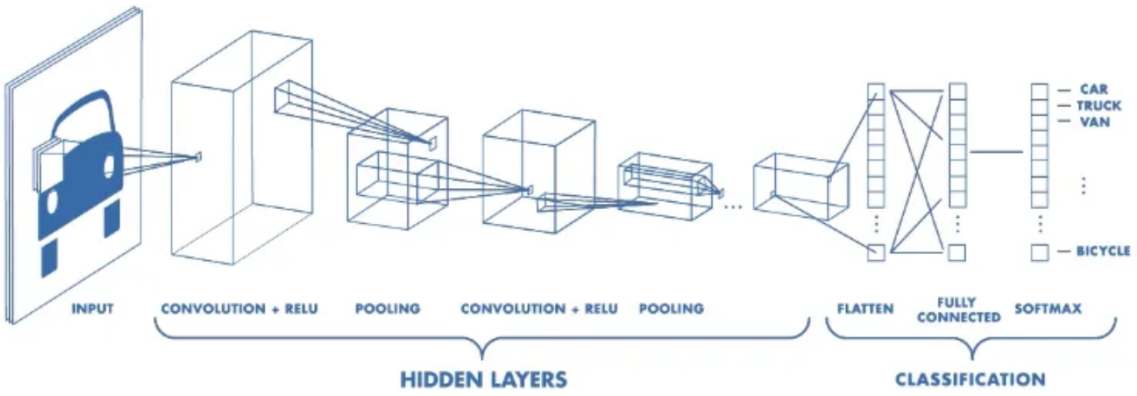



Fig 3 CNN architecture.

Keras is a versatile API widely used in artificial intelligence and machine learning for neural network models. Among its components, Keras Sequential excels in managing the ordering or sequencing of layers within a model. This functionality seamlessly integrates layers associated with neural networks into the Keras API or Keras library. The Keras Sequential model, true to its name, accepts a single input and anticipates a single output. This modeling approach is well-suited for addressing straightforward, layer-based problems.

The Keras Sequential class plays a crucial role within the broader framework of the Keras sequential model. This class facilitates the creation of a coherent structure, forming a cluster where layers of information or data flow in a top-to-bottom manner. It encompasses numerous layers incorporated with tf.Keras, creating a model where many features are trained using algorithms that emphasize sequential patterns within the data.

ResNet, short for Residual Network, represents a distinctive category of convolutional neural networks (CNN) introduced through the influential 2015 paper titled "Deep Residual Learning for Image Recognition" by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. CNNs are widely employed in driving various computer vision applications.

ResNet-50 specifically refers to a 50-layer convolutional neural network, encompassing 48 convolutional layers, one MaxPool layer, and one average pool layer. Residual neural networks, a subset of artificial neural networks (ANN), construct their networks by stacking residual blocks, contributing to their unique architecture and capabilities.

The archetechture of ResNet50 is provided below.

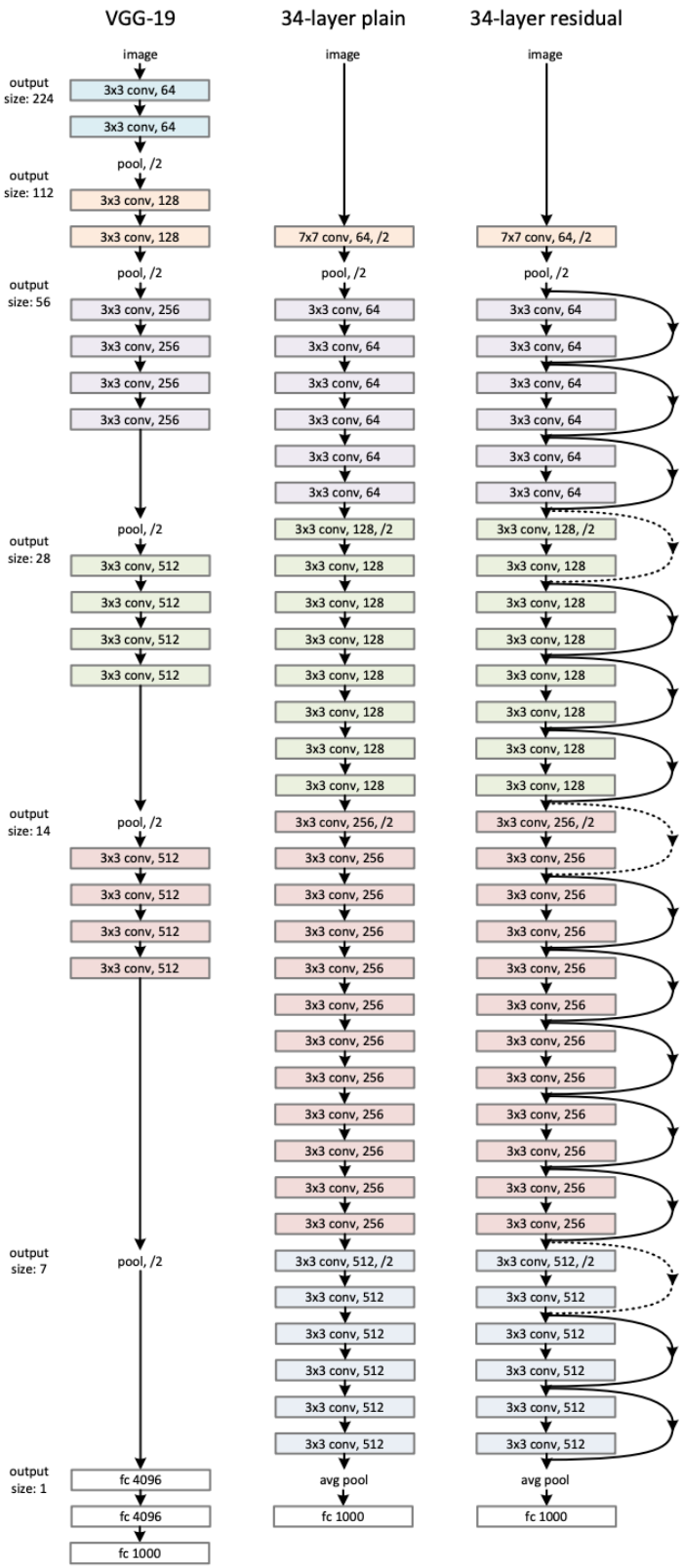



Fig 3 ResNet 50 architecture.

## Results

The plot below shows the accuracy for training and validation dataset. We can see that as the epoch increases from 1 to 10, the model training accuracy increased from around 77% to 82% and the validation accuracy increased from ~80% to 83%.

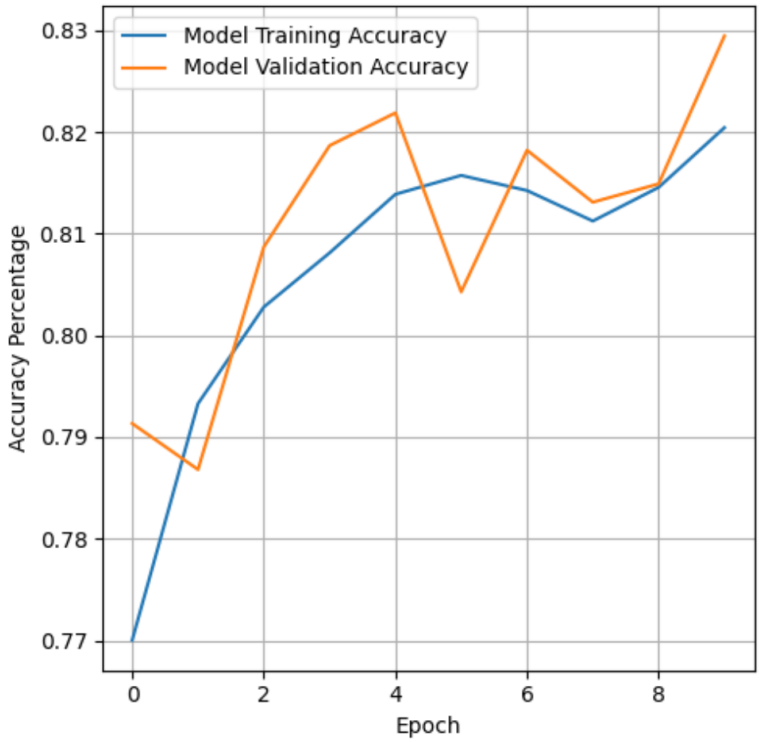



Fig. 3 Model accuracy across training epoch

Fig. 4 shows the percentage of loss for training set and validation set. The training loss deceased from ~50% to 41% and the validation loss decreased from 46% to 39%.

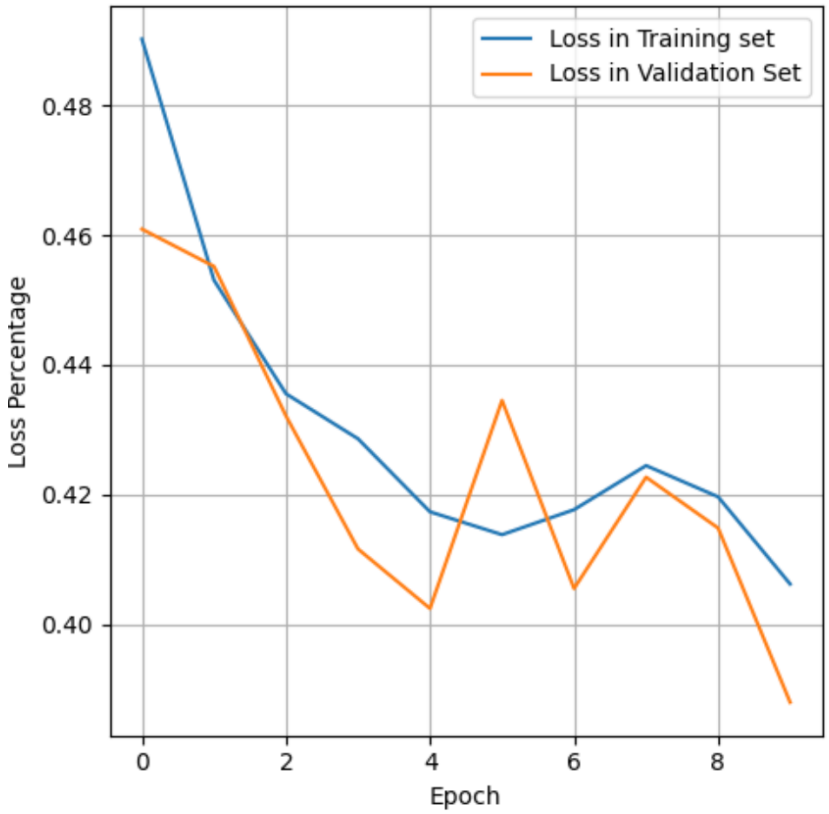
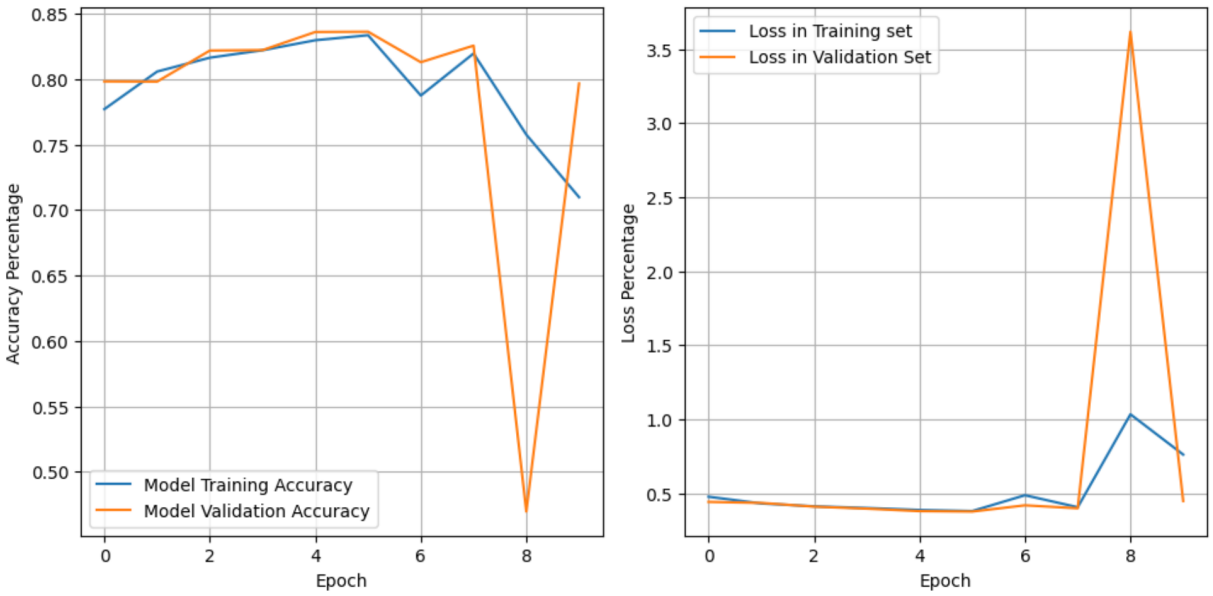


Fig. 4 Model Loss across training epoch

One hyperparameter we are working on is the batch number and image size. The plot below shows the accuracy and loss after the image size being changed from 64 by 64 to 96 by 96. We can see that for the validation set the accuracy loss are not stable and have a sudden change in epoch 9.



The result of ResNet50 is provided in the plots below. Note that due to the laptop memory limit, the training is limited to 5 Epoch. We can see that as the epoch increases from 1 to 5, the model training accuracy increased from around 85% to 91% and the validation accuracy increased from ~60% to ~80% and then drop. If more epochs are used to train the model, the accuracy will get to a stable state.

Fig. 6 shows the percentage of loss for training set and validation set. The training loss deceased from ~40% to 20% and the validation loss is not in a stable state within 5 training epoch.

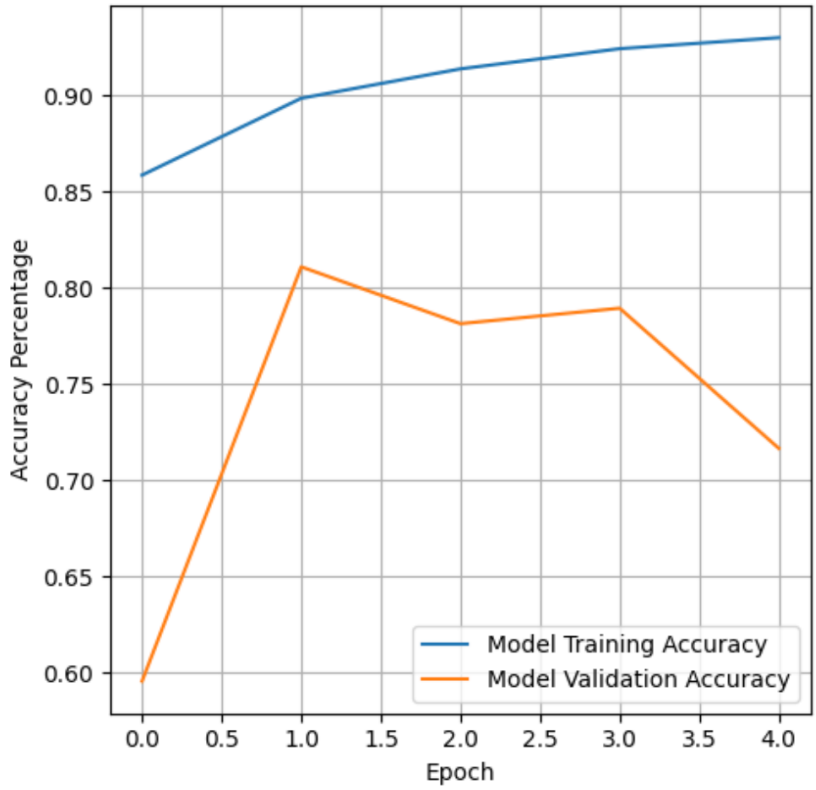



Fig. 5 Model accuracy across training epoch

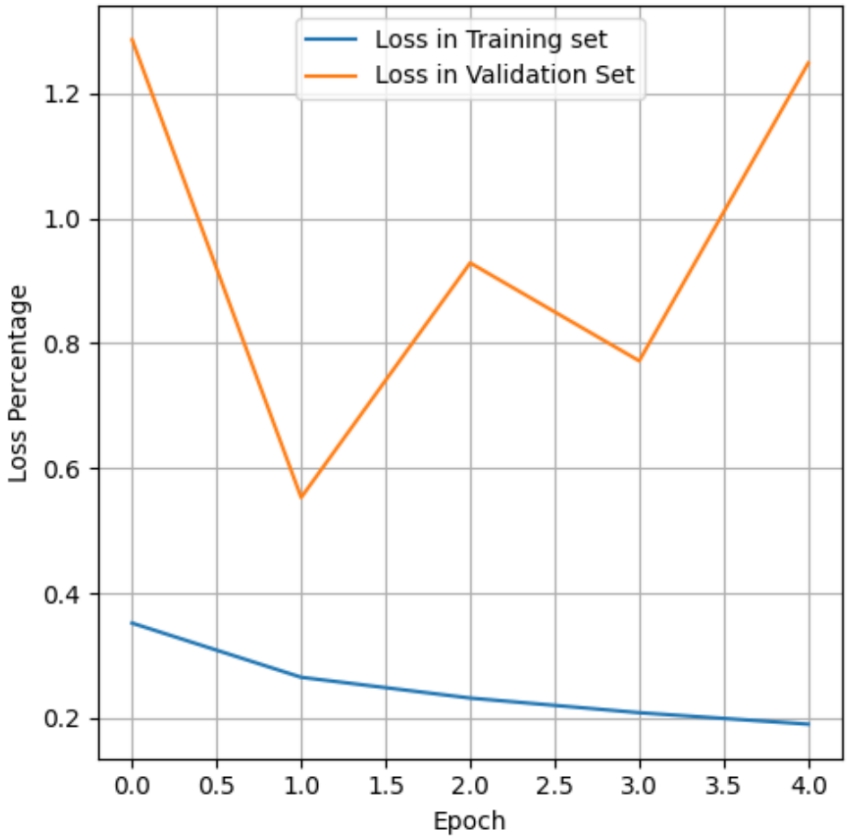


Fig. 6 Model Loss across training epoch

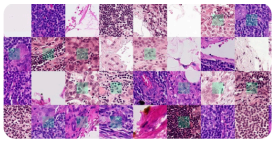
# Conclusion

In this study, we utilize Convolutional Neural Network (CNN) techniques to predict cancer patients based on their images. Notably, the CNN model without BatchNormalization outperforms the ResNet50 model, suggesting a potential influence of insufficient epochs for the latter method. The parameters under consideration for optimization are the batch number and image size. Through experimentation, we observe that the combination of a batch number set at 256 and an image size of 64 yields the highest accuracy and optimal model performance. This insight emphasizes the importance of tuning these parameters to enhance the predictive capabilities of the CNN model in the context of cancer patient classification.

# Submission

## Histopathologic Cancer Detection

Identify metastatic tissue in histopathologic scans of lymph node sections



Overview Data Code Models Discussion Leaderboard Rules Team Submissions

### Submissions

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submission, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

0/2

Submissions evaluated for final score

All Successful Selected Errors Recent

| Submission and Description | Private Score | Public Score | Selected |
|---|---|---|---|
| **submission2.csv** Complete (after deadline) · 1m ago · first try | **0.7442** | **0.8088** | ☐ |

# Reference


[1]
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[2]
https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

[3]
Sumaiya Dabeer, Maha Mohammed Khan, Saiful Islam, Cancer diagnosis in histopathological image: CNN based approach, Informatics in Medicine Unlocked, Volume 16, 2019, 100231, ISSN 2352-9148, https://doi.org/10.1016/j.imu.2019.100231.

[4]
https://huangqinjian.blog.csdn.net/article/details/85017223?spm=1001.2101.3001.6650.4&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-4-85017223-blog-109177206.235%5Ev39%5Epc_relevant_yljh&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-4-85017223-blog-109177206.235%5Ev39%5Epc_relevant_yljh&utm_relevant_index=8

[5]
https://sandipanweb.wordpress.com/2023/03/26/histopathologic-cancer-detection-with-cnn/

[6]
https://developer.apple.com/metal/tensorflow-plugin/