

The background of the slide features several abstract, organic shapes in various shades of purple and blue. These shapes are layered and wavy, resembling topographical map lines or perhaps stylized representations of cellular structures. They are positioned around the central text, with some shapes appearing more prominent than others.

Histopathologic Cancer Detection Using Deep Learning

Table of contents

01

About Problem

Describing the primary problem being targeted

02

Objective and Motivation

Primary objective and motivation behind the project.

03

Dataset Description

Detailed description of the dataset used

04

Architecture

CNN models and tools used

05

Results

Training and testing results of the models

06

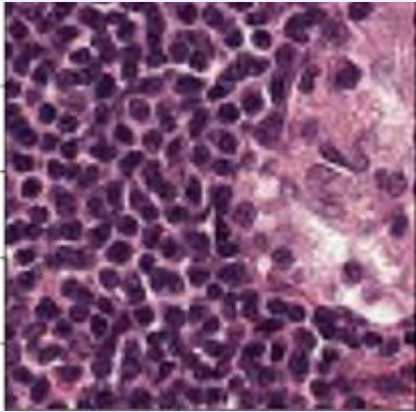
Conclusion

Conclusions drawn from the project

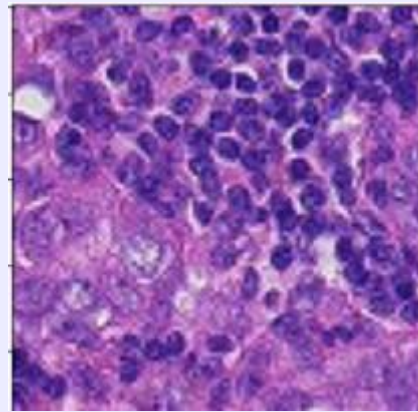
About Problem

- A lot of people lost their live to cancer annually. Majority of deaths due to cancer could be prevented if it is detected in its early phases.
- In cancer diagnosis, radiologists must decide as to whether a possible abnormality in a given MRI/X-Ray image(s) is cancer and dismiss anything that may be mistaken for cancer cells. Moreover, there is always a chance of cancer passing unnoticed even through experienced eyes.

About Problem (continued)



Benign Cell



Malignant Cell

Objective and Motivation

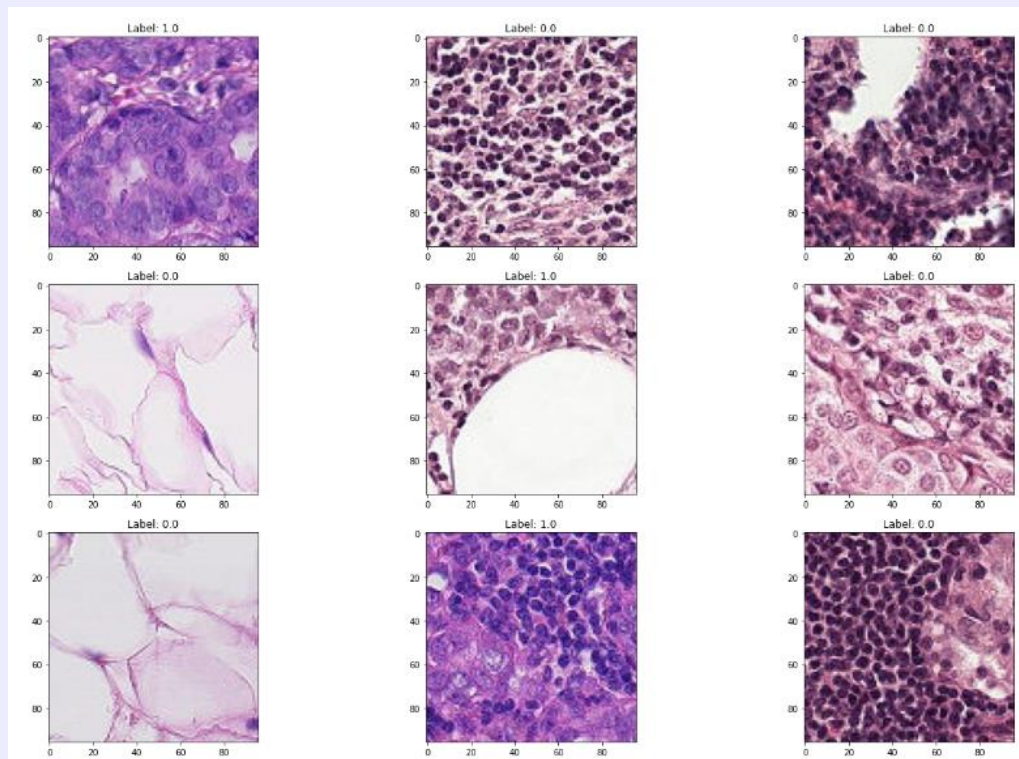
- The ability of minimizing imperishable human error in detecting cancer cells is this primary motivation behind this project.
- The primary objective of this project is to prepare a Deep Learning model that analyses the small patches of pathology scans of metastatic cancer in early phases using Convolutional Neural Nets (CNN).

Dataset Description

- The dataset contains **96*96** (malignant/benign) images of pathology scans in '.tif' format. It contains **220K** labelled training images and **57.5K** unlabelled testing images. The images are taken from various sources, mostly cancer hospitals and labs across the United States.
- This dataset was originally hosted on Kaggle and it was provided by Bas Veeling , with additional inputs from Babak Etheshami Bejnordi, Geert Litjens, and Jeroen van der Laak.

Dataset Description (continued)

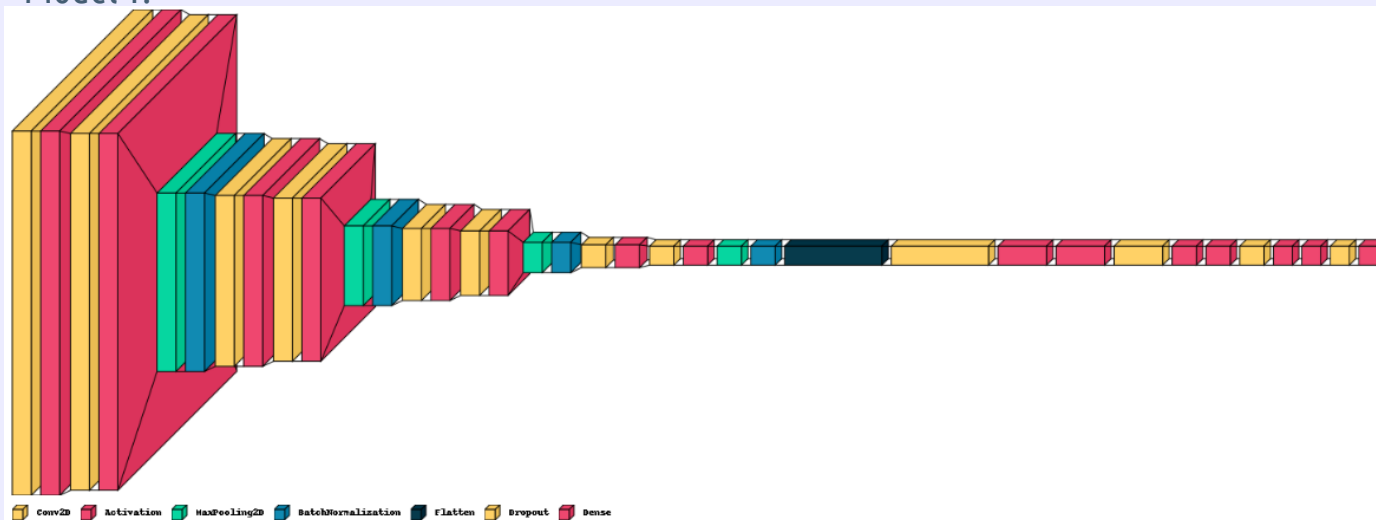
Sample Images:



Architecture

The different CNN architectures are being used in this project, as described below:

Model 1:



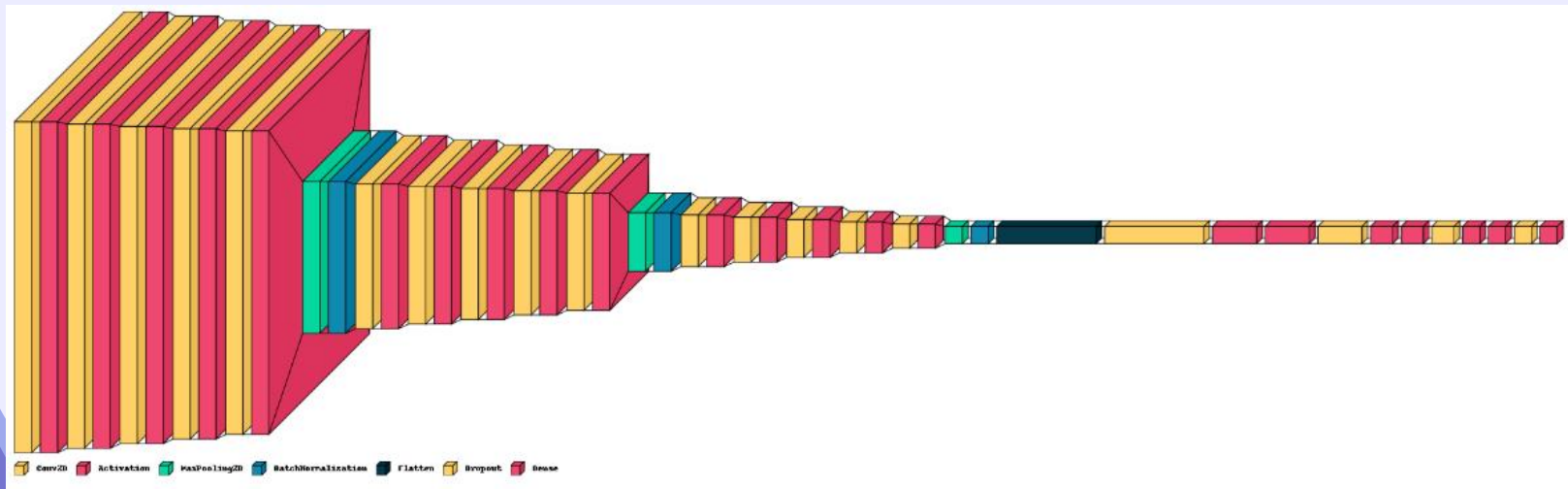
Filters: 32
Kernel Size: 3*3
Padding: "same"

Activation: "ReLu"
Max Pooling(Poolsize): 2*2
Dropout: 0.25

Optimizer: Adam(learning_rate=0.001)
Epochs: 30
Loss: Binary Crossentropy

Architecture (continued)

Model 2:



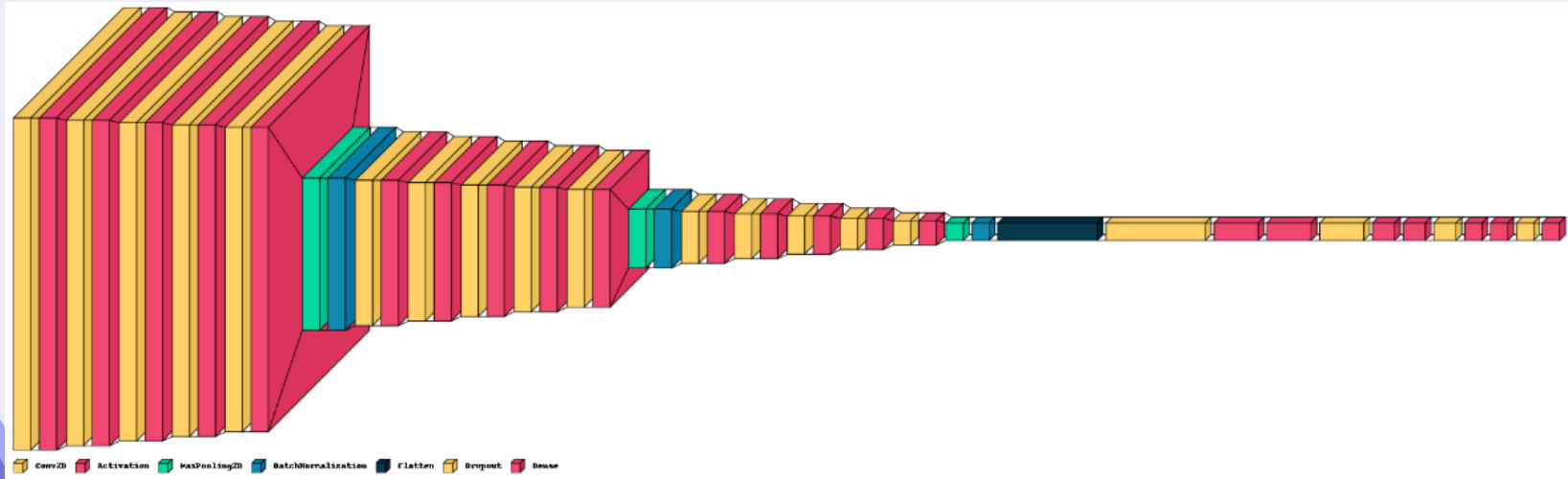
Filters: 32
Kernel Size: 3*3
Padding: "same"

Activation: "ReLu"
Max Pooling(Poolsize): 2*2
Dropout: 0.25

Optimizer: Adam(learning_rate=0.02)
Epochs: 30
Loss: Binary Crossentropy

Architecture (continued)

Model 3:



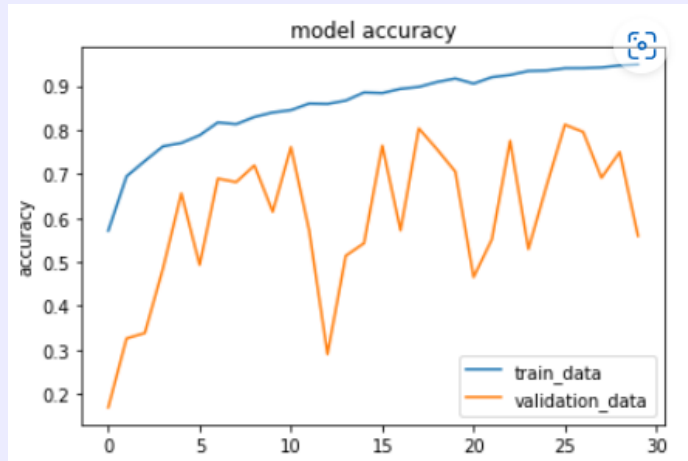
Filters: 32
Kernel Size: 3*3
Padding: "same"

Activation: "ReLU"
Max Pooling(Poolsize): 2*2
Dropout: 0.25

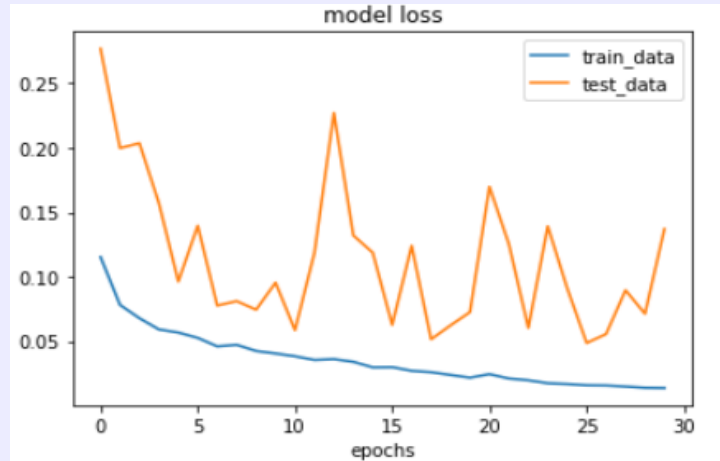
Optimizer: RMSprop(learning_rate=0.001)
Epochs: 30
Loss: Binary Crossentropy

Results

Model 1:



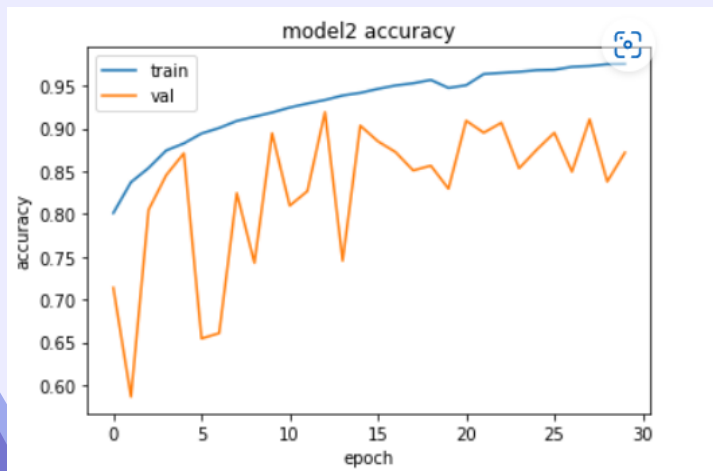
Model Accuracy



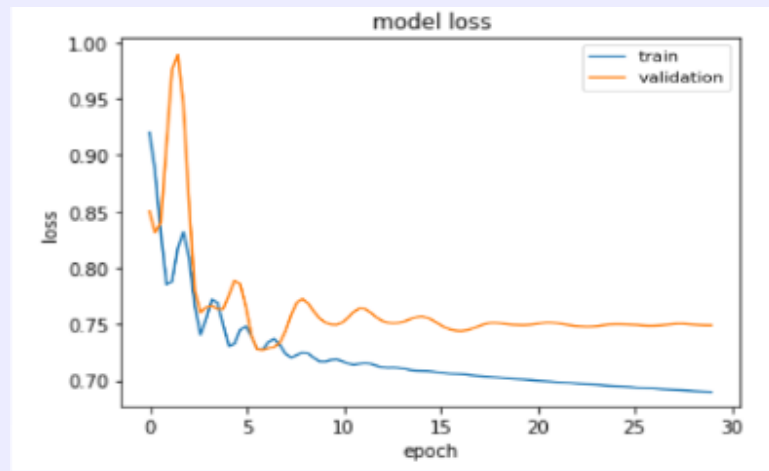
Model Loss

Results (continued)

Model 2:



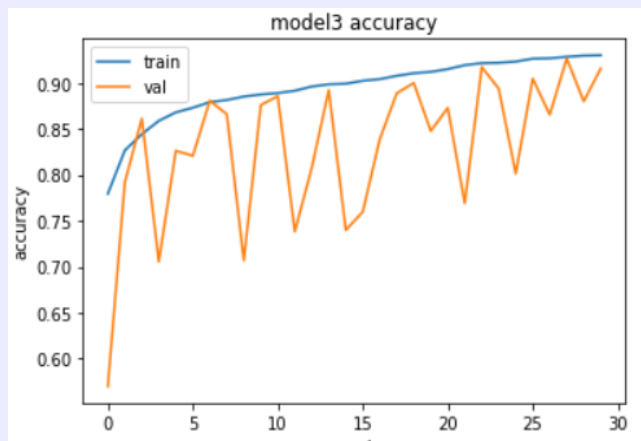
Model Accuracy



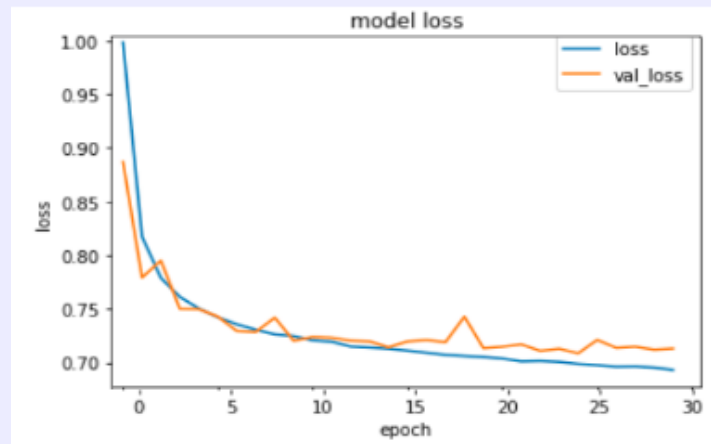
Model Loss

Results (continued)

Model 3:



Model Accuracy



Model Loss

Conclusions

- In this project, we were able to achieve high testing accuracy. However, this use case is highly critical and requires continuous improvements.
- **Model 3** has the highest testing accuracy (**approx. 90%**).
- **RMSprop(learning_rate=0.001)** seems to be the better optimizer for selected architectures.
- The use of some other CNN architectures and **transfer learning** could improve the performance at the expense of higher computational complexity and versatility.