Breast Cancer Image Detection Using ANN and CNN

Name: Choeng Veyseng
ID: M060806

Summary of the Report

Using breast images from computed tomography scan, also known as CT scan as the inputs to detect the images whether they are containing the cancer tumors or not is one of the important works to support medical field to reduce the time of detecting the tumor manually. Moreover, deep learning is a trending technology that is help many parts of medical field for doing automation task of identifying and detecting many objects by the machine itself. This project is using breast cancer tumor images with the support from convolutional neural network (CNN or ConvNet) for detecting the images contain benign tumor or malignant tumor or no tumor with three classifications. The highest accuracy from the models using CNN is 71.79 % with testing set and 99.57 % with training set.

Keywords

Breast Cancer, Convoluntional Neural Network (CNN), Artificial Neural Network (ANN)

Objectives

- Understanding the image dataset and how to work with images
- Doing the preprocessing procedure and convert images to pixels data passing to model to train
- Building model from CNN to detect the tumor from images output
- Evaluating the model through technical observations of the training and evaluation flow

Introduction to the Project

I would like formally introduce this project is about breast cancer tumor detection by passing medical images as the input and get the result that predict whether the diagnosis of the image is normal, benign or malignant. With the inspiration of a journal paper [3] studies on the predicting the breast cancer with 2 labels: benign, and malignant, the proposed solution would introduce the effective way of implementing the model with CNN. It is one of the reasons, I would like to take CNN model for building the prediction model with images.

Dataset Introduction

To process the data and assign the data to the models, understanding and knowing the dataset clearly are the fundamental requirements. Breast Ultrasound Images dataset [2] is consisted of 3 classes: normal, benign and malignant. For the convenience of the training and testing process, the labels in string form is converted to integer accordingly such as normal as 0, benign as 1 and malignant as 2. The total data of normal label is 133, data of benign label is 437 and data of malignant is 210.

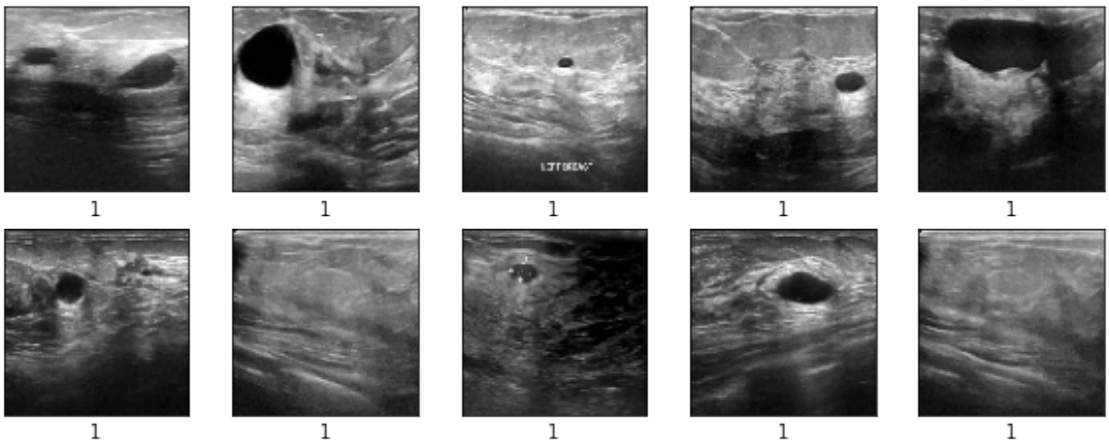


Figure 1: Images of Breast Tumor before Shuffle

Since the dataset is assigned accordingly, there would be obvious that image representation using Matplotlib library of Python showing only one label which 1 that is benign tumor. There would be the process of shuffle in order to make the randomness of the dataset.

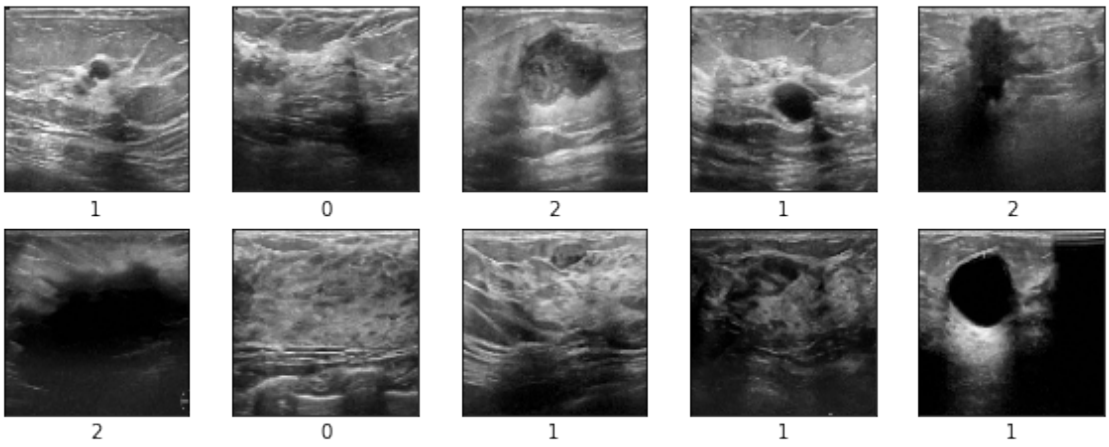


Figure 2: Images of Breast Tumor after Shuffle

After successfully applying the shuffle process, the image representations are consisted of three labels which would be better to assign to the training loop of the models.

Link to Access the Dataset on Google Drive: Breast Ultrasound Images dataset Link

Experiment Setup

The project source code is running on Google CoLab, a product from Google Research in order to execute codes specially Python code for the experiment process. So in order to run effectively on this project, the internet connection is recommended. Tensorflow framework is using in this project to form CNN models in the training phase.

There are the required libraries in order to support the running of the project.

```
import numpy as np
import cv2
import os
import matplotlib.pyplot as plt
```

- Numpy library is the most common library that is used with parameters that are the number types and can convert to numpy array.
- Computer Vision library or cv2 is used for read and work with the image in the project.
- OS library is used for working with directories and paths.
- Matplotlib library is used for plotting the images after reading from directory with specific path and plot the results of the models.

```
from google.colab import drive
drive.mount("/content/drive")
```

- Importing drive to work and mount the personal Google drive with Google CoLab project.

```
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
```

- Shuffle is used to random the data to avoid the imbalance occurence of training and testing dataset.
- train_test_split is used in order to ease the task of splitting the dataset by only indicate the percentage for training and testing datasets from the full dataset.

```
import tensorflow as tf
from tensorflow.keras import layers, models
```

1. TensorFlow library is needed to support the design of CNN as the models
2. Keras is actually a sub-library of TensorFlow that we need to import some features as well to form the models alongside the coding process

After import all important libraries and features to support the project, we can start do the coding and other tasks to reach the training phase and testing phase while also get the evaluation through plotting on the graph at the end of the project.

Convolutional Neural Network

Before go through the methodology, technology for implementing is crucial to understand to adjust the dataset into the successful implementation within the process of training. Convolutional Neural Network or CNN can be defined as specific model of neural network which is consisted of three main layers: Convolutional layers, Pooling layers, and fully-connected layers. In the figure 3, it shows the concepts and key elements of CNN which is started from the input, then pass to convolutional layers and pooling layers that are working together, flatten process, and then passing to fully-connected layers until reaching to the outputs that satisfy the end goal of the model.

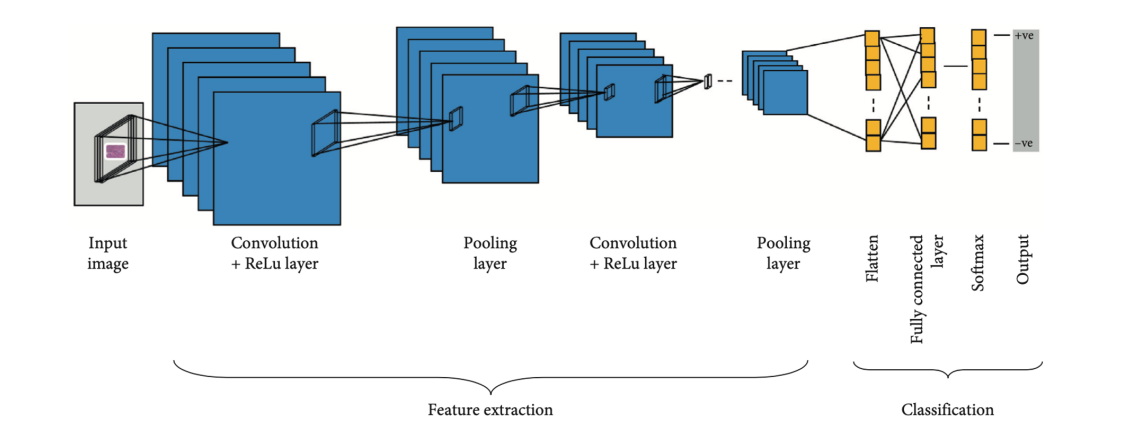


Figure 3: CNN architecture

Convolutional layers are one of the core elements within the CNN model and its feature is to filter the feature and learn along the way through feature selection to generate feature maps as show in figure 4 and Relu is being used as the activation function to avoid the exponential growth within the process.

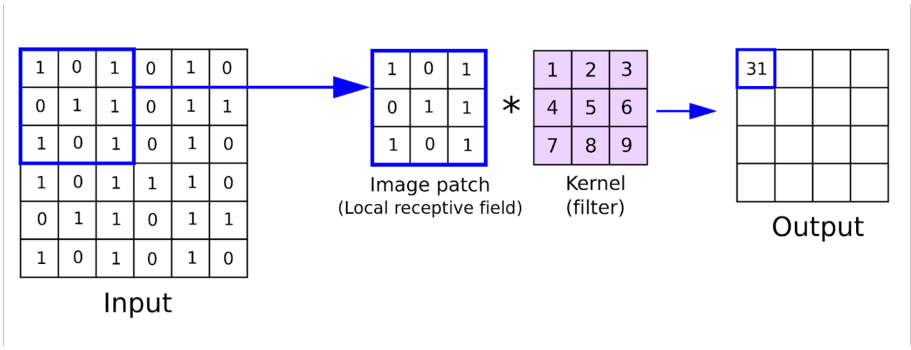


Figure 4: Example of Convolution Operation

Pool layers work like the filter from the convolutional layers where each pooling layer would perform after the convolutional layer to resampling the data by reduce the dimensions through combination of neuron clusters at one layer to single layer for the next layer which is the convolutional layer or flatten process. Mostly the local pooling combined small cluster with 2 x 2 pixels. There are two main type of pooling operation"

1. Maximum pooling is the process of calculating the maximum value of each cluster and filter like the operation in figure 5
2. Average pooling is the task of calculating the average value of each cluster.

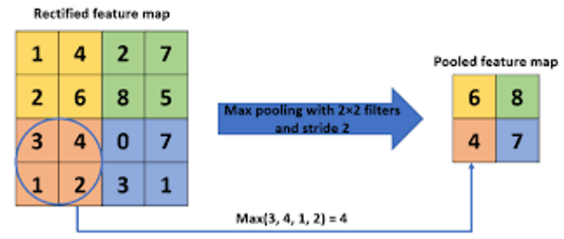


Figure 5: Example of Pooling Operation

Fully connected Layers have the functionality like artificial neural network which is the feed forward neural network. It has full connections of all activation in the previous layer to drop out the layer until it reaches the expected number of outputs. This layers are using Softmax or Sigmoid as the activation function to predict the outcomes. In the figure 6 shows the location of fully connected layers on the CNN model.

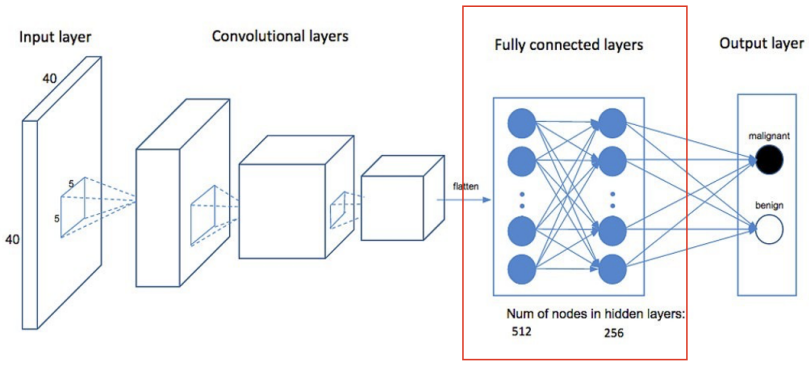


Figure 6: Example of Convolution Operation

Methodology

Deep learning is required the forming of the right learning rate, activation functions and many more features in order to build good models.

1. Visualize image dataset in order to understand the pattern of order
2. Pre-processing on the dataset and scale the image sizes
3. Resize the images into 100 pixels x 100 pixels
4. Create CNN models using TensorFlow framework consisted of 3 convolutional layers, 2 Max Pooling layers, 2 dense layers or fully-connected layers and ensure there are 3 output classes
5. Adam optimizer or Adaptive Momentum Estimation optimizer for the optimization task
6. Cross Entropy loss is used to calculate loss function from the model
7. Accuracy rate is the evaluation metrics for the models
8. the final task is to plot the testing dataset accuracy rate compare with the training dataset accuracy rate among side with number of epoches

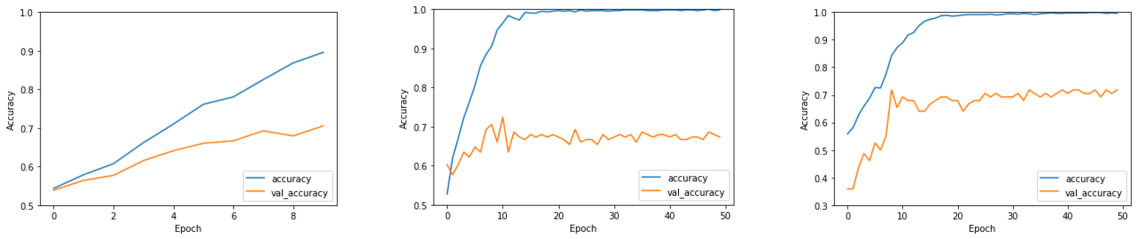
There are 3 models to be used in this project:

1. Model1: Using 10 epoches, and 80%-20% training and testing splitting
2. Model1: Using 50 epoches, and 80%-20% training and testing splitting
3. Model1: Using 50 epoches, and 90%-10% training and testing splitting

For reference of implementation, this three models are the simple CNN model that are customized from TensorFlow official tutorial website [1] that have image classification prediction which is similar to this term project.

Results

After the implementation on coding side, the results are considered to be good as they are better than the previous detection using only feed forward neural network or artificial neural network model by only 66.15% with testing set. The model 3 tends to perform better than other two models which are model 1 and model 2 by reaching to 71.79% with testing set follow by 70.51% and 67.31% accordingly with testing set. With the observation of the three graphs show on figure 7, the performance of training process reach the best result after training with 10 epoches.



(a) Model 1 (b) Model 2 (c) Model 3

Figure 7: Results of three models

Conclusion

After all processes to get the results, it is crucial to understand the theory and practical examples for working with images and data in the context of computer vision and image processing field. Deep learning is one trending technology that would eventually support the prediction of the images more effectively and could apply to many advanced topics which are working with images or videos or many more type of data. One the great things about deep learning is that it could deal with big amount of data input and with this case it would gradually work better than machine learning models which are not using the neural network concept. In the deep learning study, there are many designed to adjust and improve the performance of deep learning use cases that are rapidly growing and convolutional neural network is one of the stand out models that is highly recommended. After finished the project, we would get more understand of how to work with medical images like breast cancer tumor images in this project and how to do with the preproccessing like sampling shuffling and splitting that dataset. Moreover, we can be about to understand the basic concept of convolutional neural network and how it work in the practical use case. The overall result is reaching to more than 70% with testing dataset and consider to work normally for prediction task.

Future Work

The results are fairly good in general but there would be better to improve on the adjustment within the models like increase more epoches and study more on the trending neural network design models that could apply for the better results for our dataset. After this project, there would be more experiments to work in order to dealing with overfitting problem and increase number of dataset in the testing dataset. Furthermore, more works on image processing could be initialized like image segmentation and contour detection. There are more models to test on such as VGG-16, VGG-19, SegNet, U-Net, and many more to work and do the experiments with. More than that doing more literature review on the recent research works in computer vision and deep learning fields would be the ideal task to do in order to get the best results for the projects related to these fields.

References

[1] Convolutional neural network (CNN) | TensorFlow core.

[2] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. 28:104863.

[3] Saad Awadh Alanazi, M. M. Kamruzzaman, Md Nazirul Islam Sarker, Madallah Alruwaili, Yousef Alhwaiti, Nasser Alshammari, and Muhammad Hameed Siddiqi. Boosting breast cancer detection using convolutional neural network. 2021:e5528622.