Lung Cancer Detection: Support Vector Machine vs VGG16 Transfer learning Model

Basic Machine Learning: Final Project

Full Name: Kalonga Mabuti

Affiliation: Urban Engineering M1 Student

Student No.: 37-236913

Email: mabuti-kalonga@g.ecc.u-tokyo.ac.jp

1. Background

Lung Cancer is the leading cause of cancer related deaths worldwide. (Lung Cancer Research Foundation, 2023)There are 2 main types of Lung Cancer. Non-Small Cell Lung Cancer (NSCLC) and Small Cell Lung Cancer (SCLC). There are other lung cancer tumours, however of all the lung cancer tumours, these 2 categories make up the largest number of Diagnoses. (The American Cancer Society medical and editorial team, 2023) Furthermore, in the case of Non-Small Cell Lung Cancer (NSCLC), 80% - 85% of lung cancer diagnoses are of this category, making it the most common type of Lung Cancer.

In terms of detection, Non-Small Cell Lung Cancer (NSCLC) progresses at a slow rate in comparison to Small Cell Lung Cancer (SCLC), however before it has been successfully diagnosed, often, it has spread to other parts of the body. (Cleveland Clinic medical Professional, 2022) As a result of the above, the focus of this report will be on the detection and diagnosis of Non-Small Cell Lung Cancer tumours.

2. Project Description

The aim of the report is to create 2 machine learning models and compare their performance with regards to Non-Small Cell Lung Cancer (NSCLC) detection. With regards to NSCLC, there are 3 main types of tumours associated with NSCLC. These include Adenocarcinoma, Squamous cell Carcinoma and Large Cell Carcinoma.

- Adenocarcinoma: Develops in the outer portions of the lungs, predominantly in the cells that secrete mucous and other substances. This type of tumour usually affects former and current smokers but can also develop in non-smokers. Finally, this type of tumour is more likely to occur at a younger age compared to other types of cancer.
- Squamous Cell Carcinoma: This type of tumour usually develops in the central part of your lungs, typically in the squamous cells. These tumours are often associated with a background of smoking.
- Large Cell Carcinoma: This type of tumour develops and progresses throughout the body at a rapid rate and can form in any portion of the lungs. This type of tumour is also known as an undifferentiated carcinoma.

2.1. Datasets

The dataset made use of, is a set of chest CT-Scan images retrieved from Kaggle. The dataset, when downloaded is made up of 3 folders, a test folder for testing data, a train folder for training data, and valid folder for validation data (Figure 1). Within each of the folders, there are 4 further folders: 3 folders for the 3 main Non-Small Cell Lung Cancer tumours, Adenocarcinoma, Squamous cell carcinoma and large cell carcinoma, and 1 folder for CT-scans showing non-infected or 'Normal' chest cells (Figure 2).

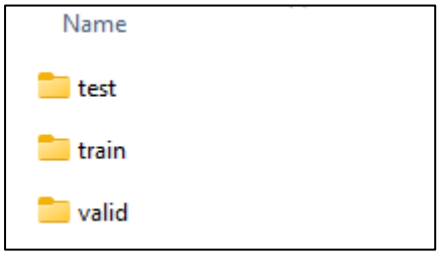


Figure 1: Train, Test, Validation

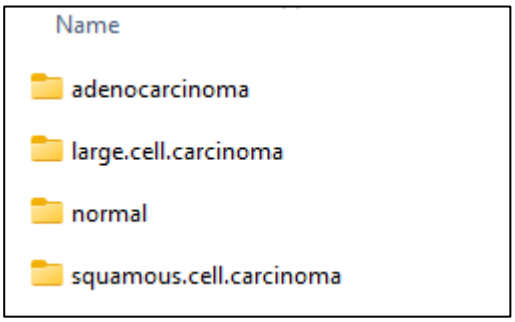
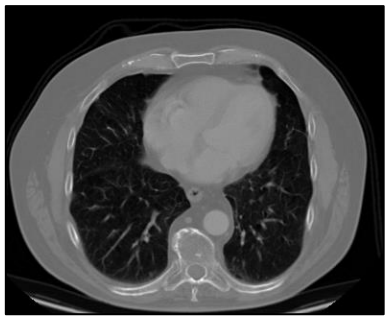
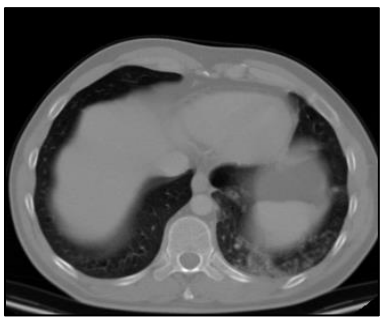


Figure 2: NSCLC data folders

Each of the non-small cell lung cancer (NSCLC) folders have CT-scan images displaying each of the above cases (Figure 3). In terms of training data and test data, for training data, we have 613 images, 115 normal cell, 195 adenocarcinoma, 115 large cell carcinoma, and 155 squamous cell images. In terms of test data, we have 315 images. 54 normal cell, 120 adenocarcinoma, 51 large cell carcinoma, 90 squamous cell images.

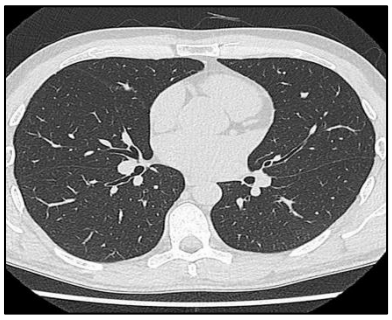


Adenocarcinoma CT-Scan



Squamous Cell Carcinoma CT-Scan

Large Cell Carcinoma



Normal Lung Cells CT-Scan

Figure 3: Dataset CT-Scans

2.2. Models

The Models that will be used include the Support Vector Machine and a Convolutional Neural Network making use of transfer learning techniques to transfer the weights from the VGG16 model to the model we will be building for Non-Small Cell Lung Cancer detection.

2.2.1. Support Vector Machine (SVM)

The support vector machine was selected for this task as among classification tasks, it exhibits significantly high accuracy in comparison to some other techniques. Additionally, in terms of the data we were processing, SVM's are exceptionally capable at processing multidimensional data, whilst being less prone to overfitting, in comparison to other modelling techniques.

2.2.2. Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) was selected as this specific type of neural network, cater specifically to the processing and analysis of image related data. Additionally, one of their main purposes is the classification of images. Rather than build a Convolutional Neural Network from scratch, the author made use of a readily available CNN model, the VGG16 model. This was done to see if the previously learnt weights and parameters would be able to assist in improving the models image classification abilities.

3. Method

In the following sections, specifics related to the models and how the experiment was conducted will be given. Due varied nature of the 2 models, significantly different data pre-processing methods as well as model testing methods were used.

3.1. Support Vector Machine (SVM)

In terms of the Support Vector Machine algorithm, various steps had to be taken prior to modelling due to the requested format of the input data required by the model, and the format of the data at hand. The data pre-processing phase followed the following steps:

3.1.1. Data Pre-processing

1. Import the datasets
    - 4 empty lists were initially created for X_train, Y_train, X_test, Y_test
    - 2 for loops were used to retrieve the data.
    - The first for loop, loops through the different Tumour files.
    - The second for loop, loops through the images and reads the image in grayscale format. This is done to reduce complexity with regards to computation. After reading the images in this format, it resizes the image to standardize the size of all the images to a width and heigh of 200.
    - Lastly the Image is appended to the X_test or X_train list, and the corresponding Tumour class is appended to the Y_test or Y_train list.

```
classes_train = {'Normal_Cell':0, 'Adenocarcinoma':1, 'Large_Cell_Carcinoma':2, 'Squamous_Cell_Carcinoma':3}
classes_test = {'Normal_Cell':0, 'Adenocarcinoma':1, 'Large_Cell_Carcinoma':2, 'Squamous_Cell_Carcinoma':3}

['Adenocarcinoma', 'Large_Cell_Carcinoma', 'Squamous_Cell_Carcinoma', 'Normal_Cell']
['Adenocarcinoma', 'Normal_Cell', 'Large_Cell_Carcinoma', 'Squamous_Cell_Carcinoma']
```

Figure 4: Tumour Classes

```
# Prepare data (Convert into greyscale and resize all images)
X_train = []
Y_train = []

X_test = []
Y_test = []

for cls in classes_train:
  pth_train = "/content/drive/MyDrive/Basic Machine Learning/Week 13 - Final Year Project/Dataset/Data/train/"+cls
  for i in os.listdir(pth_train):
    img_train = cv2.imread(pth_train+'/'+i, 0)
    img_train = cv2.resize(img_train, (200, 200))
    X_train.append(img_train)
    Y_train.append(classes_train[cls])

for cls in classes_test:
  pth_test = "/content/drive/MyDrive/Basic Machine Learning/Week 13 - Final Year Project/Dataset/Data/test/"+cls
  for j in os.listdir(pth_test):
    img_test = cv2.imread(pth_test+'/'+j, 0)
    img_test = cv2.resize(img_test, (200, 200))
    X_test.append(img_test)
    Y_test.append(classes_test[cls])
```

Figure 5: For Loops Importing the Data

2. Convert data values (X_train, X_test, Y_train, Y_test) to NumPy Array

```
# Convert to Array
X_train = np.array(X_train)
Y_train = np.array(Y_train)

X_test = np.array(X_test)
Y_test = np.array(Y_test)
```

Figure 6: Convert to NumPy Array

3. Convert the 3-Dimensional X_train and X_test data into 2-Dimensional Data

```
# Convert the data into a 2-Dimensional data rather than 3-Dimensional
x_train_updated = X_train.reshape(len(X_train), -1)
x_train_updated.shape
```

Figure 7: Convert to 2-Dimensional Data

4. Normalize the images pixel values to keep them between 1 and 0.

```
# Feature Scaling
print(xtrain.max(), xtrain.min())
print(xtest.max(), xtest.min())

xtrain = xtrain/255
xtest = xtest/255

print(xtrain.max(), xtrain.min())
print(xtest.max(), xtest.min())
```

Figure 8: Normalize the images Pixel Values

3.1.2. SVM Model Initialization

Finally, after pre-processing the data the SVM model can be initialized as follows:

```
# Train the Model
svm_model = SVC(kernel ='rbf', C = 100)
svm_model.fit(pca_train, ytrain)
```

Figure 9: Initialize the SVM model

The author had attempted to make use of the GridSearchCV function to tune the SVM to apply the best possible hyperparameters for the model and improve the results, however, this was not possible due to the lack of computational resources.

3.2. Convolutional Neural Network (via Transfer Learning)

With regards to the Convolutional Neural Network algorithm, in terms of the data, although the data we have is image data, some pre-processing steps were still required before any kind of predictive analytics could take place. A Base model was imported to transfer valuable knowledge previously learnt, in the form of weights and parameters.

3.2.1. Data Pre-processing

1. Import the dataset
    - This was done in much the same as was done with the Support Vector Machine, however, with 3 noticeable differences.
    - The first difference is that the images that were read from the source file, were not read in grayscale format, but however were read in BRG format. Due to this occurrence, it was required that after importing the images, they be converted from BRG to RGB as is the standard colour channel format.
    - The images, as was previously done in the SVM pre-processing, were resized to a width and heigh dimension of 224. This was done as the Base model used, the VGG16 base model, requires the input images to be of at least 224 or less.
    - The image pixel values were normalized once again. However, a different normalization technique was used. The normalization technique used came from the cv2 library. Here, an zeros matrix of the same dimensions (224,224,3) was created. After which the cv2.normalize function was called, and the result was transferred into the zeros matrix. Following this step, as was previously done, the X_train, X_test and corresponding Y_train and Y_test values were appended to the previously empty lists.

```
# Prepare data (Convert into greyscale and resize all images)
classes_train = {'Normal_Cell':0, 'Adenocarcinoma':1, 'Large_Cell_Carcinoma':2, 'Squamous_Cell_Carcinoma':3}
classes_test = {'Normal_Cell':0, 'Adenocarcinoma':1, 'Large_Cell_Carcinoma':2, 'Squamous_Cell_Carcinoma':3}

X_train = []
Y_train = []

X_test = []
Y_test = []

for cls in classes_train:
  pth_train = "/content/drive/MyDrive/Basic Machine Learning/Week 13 - Final Year Project/Dataset/Data/train/"+cls
  for i in os.listdir(pth_train):
    # Convert BGR image to
    brg_img_train = cv2.imread(pth_train+'/'+i)
    rbg_img_train = cv2.cvtColor(brg_img_train, cv2.COLOR_BGR2RGB)
    Resize_img_train = cv2.resize(rbg_img_train, (224, 224))

    #Normalize (cv2.normalize())
    norm_img_train = np.zeros((244, 244, 3))
    normalized_img_train = cv2.normalize(Resize_img_train, norm_img_train, 0, 1.0, cv2.NORM_MINMAX, dtype=cv2.CV_32F)
    img_train = normalized_img_train

    X_train.append(img_train)
    Y_train.append(classes_train[cls])
```

Figure 10: Data Import Process

3.2.2. CNN Model Initialization

With regards to the base model, the VGG Net 16 model, the last 6 layers were unfrozen. This was done to allow some of the input image features to be learnt by the last 3 Convolutional Neural Network layers, slightly shift the weights, to increase the accuracy of the model when examining CT-Scan images. Following the unfreezing of the base model layers, and additional 3 fully connected layers were added to the model, each with a 20% dropout rate, to increase the accuracy in terms of classification, whilst preventing overfitting. This model can be seen in Figure 11. Additional to this, the following model summary with trainable parameters can be seen in Figure 12.

```
base_model.trainable = False

for layer in base_model.layers[-6:]:
  layer.trainable = True

new_model = models.Sequential([
    base_model,
    layers.Flatten(),
    layers.Dense(516, activation='relu'),
    layers.Dropout(0.2),
    layers.Dense(516, activation ='relu'),
    layers.Dropout(0.2),
    layers.Dense(516, activation='relu'),
    layers.Dense(4, activation='softmax')
])

new_model.compile(
    optimizer = 'adam',
    loss = 'sparse_categorical_crossentropy',
    metrics = ['accuracy']
)
```

Figure 12: Transfer Learning CNN Model

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14714688
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 516)	12945924
dropout (Dropout)	(None, 516)	0
dense_1 (Dense)	(None, 516)	266772
dropout_1 (Dropout)	(None, 516)	0
dense_2 (Dense)	(None, 516)	266772
dense_3 (Dense)	(None, 4)	2068

Total params: 28,196,224
Trainable params: 13,481,536
Non-trainable params: 14,714,688

Figure 11: CNN Model Summary

4. Results

The following section will display the results from the experiments.

4.1. Support Vector Machine (SVM) Results

After training and fitting the model, to accurately view the performance of the model, a Classification report was produced. (Figure 11)

```
print('Classification Report')
print(classification_report(ytest, y_predict, target_names=['Normal_Cell', 'Adenocarcinoma',
                                                            'Large_Cell_Carcinoma', 'Squamous_Cell_Carcinoma']))
```

```
Classification Report
                        precision    recall  f1-score   support

           Normal_Cell       1.00      1.00      1.00        54
        Adenocarcinoma       0.57      0.57      0.57       120
   Large_Cell_Carcinoma       0.42      0.82      0.55        51
Squamous_Cell_Carcinoma       0.68      0.30      0.42        90

              accuracy                           0.61       315
             macro avg       0.66      0.67      0.63       315
          weighted avg       0.65      0.61      0.60       315
```

Figure 13: SVM Classification Report

From the classification report, we can see that the model is producing a 61% accuracy rate, but what does this mean? The most notable findings can be described as follows:

- Normal Cell Class
  - *Precision* – 100% of all predictions, were in fact correct predictions
  - *Recall* – 100% identification when predicting other tumours
  - *F1-score* –Normal cells have a 100% accuracy in terms of overall prediction.
- Adenocarcinoma
  - *Precision* – 57% accuracy in identifying adenocarcinoma when looking for adenocarcinoma tumours
  - *Recall* – 57% accuracy in identifying adenocarcinoma when predicting other classes of tumours
  - *F1-score* – 57% accuracy overall related to identifying Adenocarcinoma tumours.
- Large Cell Carcinoma
  - *Precision* – 42% accuracy in identifying large cell carcinoma when looking for large cell carcinoma tumours, however,
  - *Recall* – An astonishing 82% accuracy in identifying large cell carcinoma when predicting other classes
  - *F1-score* – Overall we have a 51% accuracy for predicting large cell carcinoma tumours
- Squamous Cell Carcinoma
  - *Precision* – 68% accuracy in accurately identifying squamous cell carcinoma, when searching for squamous cell carcinoma amongst the datasets.
  - *Recall* – 30% accuracy in identifying squamous cell carcinoma tumours when identifying other types of tumours
  - *F1-score* – Ultimately the overall accuracy related to the prediction of squamous cell carcinoma tumours is 42%

These values are not satisfactory at all. It can be assumed that, had more computational resources been available, with the use of the GridSearchCV function the results may have yielded better results.

As such the author believes that this is not an accurate representation of the capabilities of the Support Vector Machine in such a classification task.

4.2. Convolutional Neural Network (via Transfer Learning)

With a Batch size of 50, running through each iteration over 10 epochs, the results were as follows:

- The final 3 epochs can be seen in Figure 14. Whilst the model accuracy rests around 98.69% Accuracy, the Validation accuracy lags at a value of 82.22% which is a large disparity between the 2 values in terms of accuracy. The extent of this disparity can be further examined in Figure 15 were we see the gap between the model performance against the training data, and that of the validation data. It can be assumed that this disparity can be attributed to the rate at which the validation loss function converges towards 0 in comparison to the training loss function.
- With regards to loss, the disparity between the training loss function and the validation loss function, the training loss function converges towards 0 at a much higher rate than that of the validation function. This can be seen in figure 16.
- With more tuning and training of the functions model, it is very likely to achieve a higher validation accuracy for the model.

```
Epoch 8/10
13/13 [==============================] - 543s 43s/step - loss: 0.0303 - accuracy: 0.9918 - val_loss: 0.5966 - val_accuracy: 0.7714
Epoch 9/10
13/13 [==============================] - 512s 40s/step - loss: 0.0384 - accuracy: 0.9853 - val_loss: 0.5066 - val_accuracy: 0.8127
Epoch 10/10
13/13 [==============================] - 540s 43s/step - loss: 0.0468 - accuracy: 0.9869 - val_loss: 0.4763 - val_accuracy: 0.8222
```

Figure 14: Final 3 Epoch results

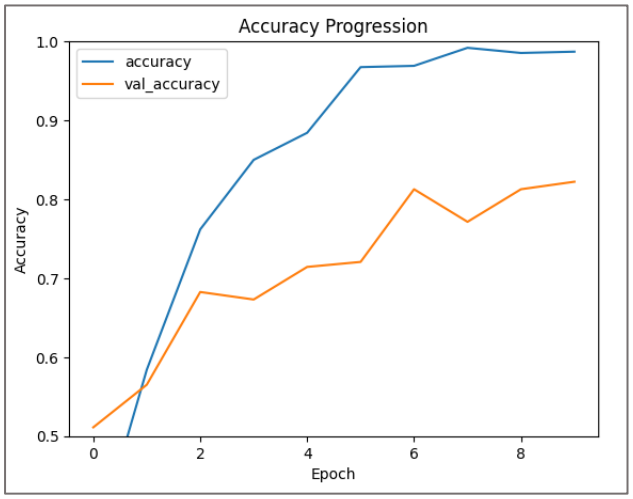


Figure 15: CNN Accuracy Progression

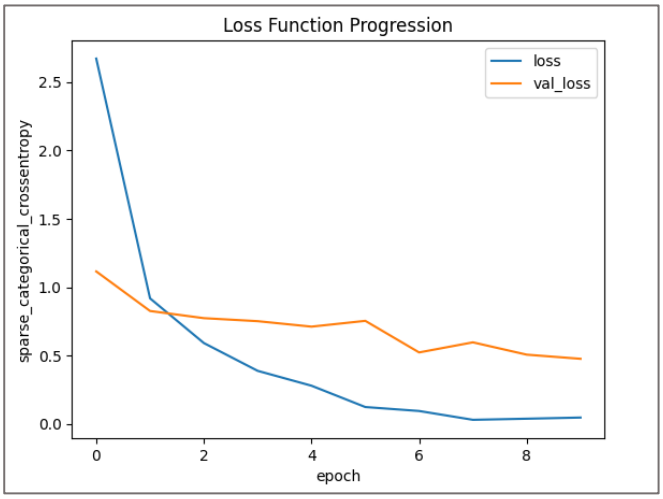


Figure 16: CNN Loss Function Progression

5. Conclusion

Both models could benefit from more parameter tuning to achieve more favourable results. However, this does not take away from the value of the experiment conducted. Perhaps from the beginning it was more likely to that the Convolutional Neural Network would outperform the Support Vector Machine classification model. However, it is almost guaranteed that with more tuning of the SVM, the accuracy of the model can be increased. At its full potential, it would be interesting to assess the results of such a model. Other implementations making use of both transfer learning, providing a base Convolutional Neural Network model to be used together with a SVM model pose additional interesting solutions to solving the problem. In the case of this project, perhaps the dataset was not

substantial enough as well and could benefit from the generation of new sample data or transformation of existing data, to increase the sample size. These would necessary to further the project.

References

Cleveland Clinic medical Professional, 2022. *Non-Small Cell Lung Cancer.* [Online]
Available at: https://my.clevelandclinic.org/health/articles/6203-non-small-cell-lung-cancer
[Accessed 28 July 2023].

Lung Cancer Research Foundation, 2023. *Lung Cancer Facts 2023.* [Online]
Available at: https://www.lungcancerresearchfoundation.org/lung-cancer-facts/#:~:text=Facts%20About%20Lung%20Cancer,cause%20of%20cancer%20death%20worldwide.&text=AN%20ESTIMATED%20238%2C340%20PEOPLE%20will,in%202023%20in%20the%20U.S.&text=1%20IN%2016%20PEOPLE%20will,and
[Accessed 28 07 2023].

The American Cancer Society medical and editorial team, 2023. *About Lung Cancer.* [Online]
Available at: https://www.cancer.org/cancer/types/lung-cancer/about/what-is.html#:~:text=About%2080%25%20to%2085%25%20of,(outlook)%20are%20often%20similar.
[Accessed 28 July 2023].