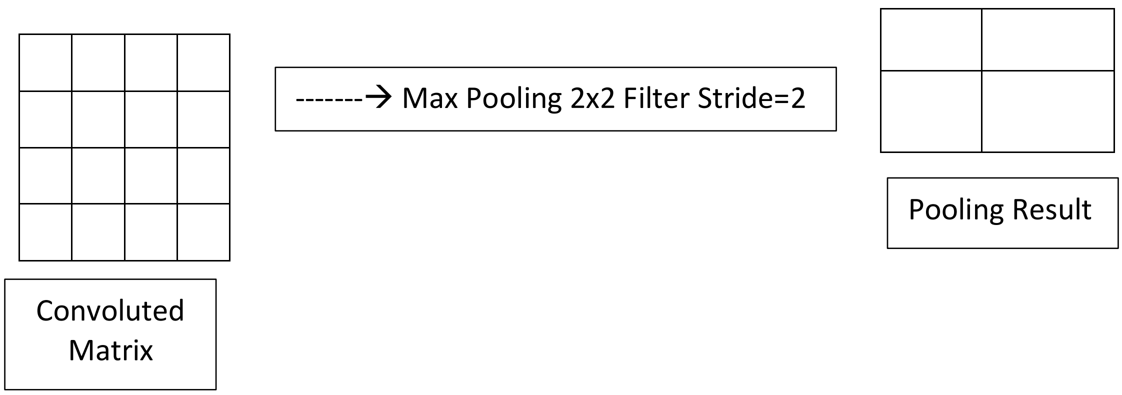
Neural Network Architecture Design through Vgg16

Pneumonia is a common a global epidemic in today's society. The injection of oxygen into the blood vessels enhances the individual's breathing cooperation. The standard course of treatment involves the administration of antibiotics or antiviral medications. Deaths among children aged 5 represent 16 percent of the total. In 2015, it resulted in the death of 920,136 children under the age of 5 years old. (Wubie, A.M.,2024) This phenomenon is mostly observed in less developed nations such as South Africa. The primary factors contributing to this issue are insufficient availability of medication and problems in promptly identifying the illness. Furthermore, it is a lethal condition for those aged 65 and above who have pre-existing chronic illnesses.

There are many kinds of variations of pneumonia. The primary categories include viral, bacterial, and fungal infections. Every one of these variations can be classified into several categories. Hence, the presence of specialists and suitable equipment is essential for the accurate identification of the disease. The diagnosis of pneumonia can be made using two methods. The initial method involves a chest X-ray (EXR), whereas the subsequent method utilises computerised tomography (CT). In certain instances, X-rays may possess invisibility to the human visual system. The weakness identified here is not attributable to the expert's insufficient knowledge. Therefore, the development of automated segmentation algorithms for the comprehensive depiction of medical pictures is both beneficial and essential from a clinical perspective. For several decades, convolutional neural networks (CNNs) have established valuable applications in the domain of computer vision. Due to this explanation, computer vision technology, specifically designed for the detection of visually challenging diseases, is employed. As this technology has advanced over time, it has been noted that it currently achieves extraordinarily high rates of accurate outcomes. The detection rate of pneumonia by CXRs has risen from 67 percent to 96 percent. (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10518240/)

A Convolutional Neural Network (CNN) is a specialised deep learning method developed for tasks that need precise object recognition, such as image categorization, detection, and segmentation. Convolutional Neural Network (CNN) model techniques have shown significant success in computer vision technology, particularly in the medical domain. The reason why CNNs have revolutionized the healthcare field is that they can detect images that were difficult to detect manually and some patterns that were previously extracted manually more accurately and quickly. Hence, Convolutional Neural Networks (CNNs) have made significant improvements to empirical investigations in the domain of disease. These models went through processing of both low and high level images, having often been pre-trained for image classification tasks. Currently, Tensorflow or Pytorch applications are the preferred choice due to their ability to generate enormously adaptable CNN architectures. They also have special libraries such as Keras. Thanks to these libraries, features can be assigned to the layers created in the CNN.

CNN architecture mimics the structure of neurons in the human visual system. In fact, it was inspired by the structure of the visual cortex in animals. The use of CNN, which began in the 1980s, was followed in the 1990s by Le Cun's Le-Net5 model, the first neuron network model. The great success of AlexNet in the competition organized by ImageNet in 2012 has almost revolutionized the perception of visual images and has begun to be used in many real-life areas. When it comes to the structure of CNNs, there are 4 main layers. These are; convolutional layers, ReLU, pooling layers and fully connected layers. The convolutional layers are the core building block of the CNN architecture and where the majority computation occurs. It requires input data, filter and feature map. The aim of the pooling layer is to extract the most important features from the convolutional matrix. It reduces the size of the feature map. Thus, it makes it purer and simpler, and on the other hand, it frees up space in the memory. Thus, it ensures that the model is faster and does not take up space. Pooling is also relevant for mitigating overfitting. There are 3 different types; max pooling, sum pooling and average pooling.



Fully connected layers are the last layer. It is a flattened one-dimensional matrix created by the last pooling. ReLU activation function is applied to prevent non-linearity. Finally, a softmax prediction layer is created. Probability values are generated for each of the possible output labels. Then, the highest probability result is found.

Tensorflow is an open-source library developed as an artificial intelligence application by the Google Brain Team in 2015. It is currently one of the most used artificial intelligence applications. (2020, Jun. 8).

Some of the models developed in Tensorflow are VGG and ResNet. Resnet has achieved state-of-the-art results in image classification, object detection, and semantic segmentation. It has been trained with quite complex images with its multi-layered structure. It has 101 layers and has been trained with 14 million images and 1000 classes on ImageNet dataset. (Madhur Jain) On the other hand, many objectives of VGG is its architecture performs for image classification tasks. Its structure is simple. Pre-trained models such as ResNet and Vgg are quite useful for limited data sets. Changes can be made to the model if desired.

Code Implementation

First of all, the necessary libraries were added. Files were loaded with the help of "os". Tensorflow library was loaded. "Sequentially" was added for the model to have sequential layers. In addition, the building blocks of other CNNs were entered. The necessary layers to import the necessary layers to create a neural network are taken from Tensorflow's Keras library. It was decided to use vgg16 in the problem, so the vgg16 model was imported. "Scikitlearn" library was loaded to measure the performance of the model and a command was entered to ignore any warnings because the system gave a warning.

Furthermore, the files downloaded to the computer were loaded with train, test and validation datasets with the os.path.join method and the number of them was printed. In the preprocessing step, the train and validation sets take the images and create the sets as necessary. 20 percent of the training data is reserved for validation. Train and validation sets are set to suit the neural network format for the vgg model. The images are set to 224x224 pixels. The batch number is set to 32.

In the model creating code, an object suitable for Vgg format is set. The pre-trained vgg model was preferred because it was successful in image classification tasks. Here the photo dimensions are set to 224x224. Imagenet weights are used and the last layer is excluded. The model creation step is done sequentially. The purpose of choosing this model is that it can give more specific and more accurate results. The model performs binary classification. There are 64 filters in the first layer and it applies a 3x3 kernel. 2x2 maxpooling was applied to reduce the dimensions. Fully connected layers were reduced by 1/2 in each layer. Then, the flatten method was applied to make the multidimensional data single-dimensional. Since binary classification was used, a "sigmoid" activation layer was used.

Adamax was selected as the optimizer function. This ensures that the gradients are more stable during learning. Then, early-stopping callback was applied. It was added to prevent overfitting and allow the model to be trained faster. It aims to stop the system if it does not improve for 3 epochs.

Finally, the model is trained for 2 epochs with the training and validation data sets and precision, f1-score, accuracy and recall scores are printed. These metrics are vital in the medical field.

```python
import os
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, MaxPooling2D, BatchNormalization
from tensorflow.keras.applications import VGG16
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import precision_score, recall_score, f1_score
from tensorflow.keras.optimizers import Adamax
import warnings
warnings.filterwarnings('ignore')

#Loading dataset for train,test and validation through os
train_dir = os.path.join('/Users', 'mustafaakgul', 'Downloads', 'chest_xray 2', 'train')
test_dir = os.path.join('/Users', 'mustafaakgul', 'Downloads', 'chest_xray 2', 'test')
val_dir = os.path.join('/Users', 'mustafaakgul', 'Downloads', 'chest_xray 2', 'val')

#Loading structure of train and validation dataset
train_df = tf.keras.preprocessing.image_dataset_from_directory(
    train_dir,
    seed=42,
    validation_split=0.2,
    subset='training',
    image_size=(224, 224),
    batch_size=32
)

val_df = tf.keras.preprocessing.image_dataset_from_directory(
    train_dir,
    seed=42,
    validation_split=0.3,
    subset='validation',
    image_size=(224, 224),
    batch_size=32
)
```

```python
#Loading vgg16 model with correct shape
vgg = VGG16(input_shape=(224, 224, 3), weights='imagenet', include_top=False)

for layer in vgg.layers:
    layer.trainable = False

#Creating model
model = Sequential()
model.add(vgg)
model.add(Flatten())
model.add(Dense( units: 128, activation='relu'))
model.add(BatchNormalization())
model.add(Dense( units: 32, activation='relu'))
model.add(BatchNormalization())
model.add(Dense( units: 16, activation='relu'))
model.add(BatchNormalization())
model.add(Dense( units: 1, activation='sigmoid'))

#Optimization algorithm and loss function
optimizer = Adamax(learning_rate=0.002)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

#Adding earlystopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)

# Model fitting
history = model.fit(
    train_df,
    validation_data=val_df,
    epochs=2,
    callbacks=[early_stopping]
)
```

```python
#collecting predictions
val_labels = []
val_predictions = []

for images, labels in val_df:
    predictions = model.predict(images)
    val_labels.extend(labels.numpy())
    val_predictions.extend(predictions)


# Evaluating model precision,recall and f1-score
precision = precision_score(val_labels, val_predictions)
recall = recall_score(val_labels, val_predictions)
f1 = f1_score(val_labels, val_predictions)

print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1-Score: {f1}")
```

```
Epoch 1/2
131/131 ━━━━━━━━━━━━━━━━━━━━ 492s 4s/step - accuracy: 0.8985 - loss: 0.2707 - val_accuracy: 0.9508 - val_loss: 0.1602
Epoch 2/2
131/131 ━━━━━━━━━━━━━━━━━━━━ 525s 4s/step - accuracy: 0.9906 - loss: 0.0657 - val_accuracy: 0.9904 - val_loss: 0.0447
```

```
Precision: 0.992274678111588
Recall: 0.9948364888123924
F1-Score: 0.9935539321014182
```

This artificial neural network, which is operated with the Vgg16 model and used to detect pneumonia, proves how successful it is in this field with a high accuracy score of 0.9906. If we look at other metrics, the fact that the precision score is 99 percent is inferred to how accurately the model detects pneumonia cases. In addition, the fact that the recall value is 0.99 shows the accuracy of the cases that are positive for pneumonia. The F1 score shows that the model is quite successful in detecting both positive patients and false positive patients. Finally, in line with the results of this metric, it turns out that the model is a very effective tool in pneumonia case study.

CONCLUSION

The project achieved excellent evaluation scores with the artificial neural network model using vgg16 architecture for classifying chest x-ray images and detecting pneumonia. First of all, the project uses 3 datasets; train, test and validation. The vgg16 architecture, which is quite successful in image classification problems, was used for the model architecture. The layers were adjusted sequentially. Adamax was used as the optimizer and binary classification was used as the loss function. Then, early stopping callback was added to make the model more specific. Finally, the model was trained and the results were obtained. With an accuracy score of 99 percent, it was able to correctly predict patients with and without pneumonia. Similarly, having high precision and recall scores, it managed to detect very few false positive and false negative cases. Finally, with an F1-score of 99.35 percent, the model generally gave a close result in detecting pneumonia cases and it was shown that the model was reliable.

In recent years, particularly in 2015, the initiation of digital health data collecting has enabled artificial intelligence to significantly enhance the healthcare sector using machine learning and deep learning algorithms. It has attained exceptionally high scores, particularly in identifying patterns, structures, and anomalies in medical imagery. This achievement in picture classification proves beneficial not only in pneumonia identification, as demonstrated in this project, but also in the categorization of medical images, including those of the brain, MRI, spine, and others. This technique not only identifies but also yields great accuracy in future disease forecasts. The positive outcomes of these models, created using deep learning networks and machine learning, significantly impact public health. Subsequent to these achievements, ample expenditures are being made by governments and corporations, resulting in the continuous advancement of artificial intelligence in the healthcare sector. NLP, in particular, allows healthcare providers to analyze patient records and make more accurate diagnoses by processing human language.

In addition to these benefits, artificial intelligence has several drawbacks. This related to the rapid integration into IT systems and the ability to establish trust among individuals or organizations providing healthcare services. Nonetheless, despite these challenges, artificial intelligence offers accelerated, more accurate, and customized services by transforming decision-making processes within the healthcare business. It is anticipated that it will enhance efficiency, accelerate processes, and fundamentally transform patient care in the future.

References

1. Jain, M., Bora, M. S., Chandnani, S., Grover, S. & Sadwal, S., Comparison of VGG-16, VGG-19 and ResNet-101 CNN Models. Available at: https://file:///Users/mustafaakgul/Downloads/Comparison_of_VGG-16_VGG-19_and_ResNet-101_CNN_Mod.pdf (Accessed: 21 October 2024).
2. Madhur, J., Bora, M.S., Chandnani, S., Grover, S., and Sadwal, S., *Comparison of VGG-16, VGG-19 and ResNet-101 CNN Models*.
3. *10 Most Popular Machine Learning Software Tools in 2020 (Updated)* (2020). Available at: https://towardsdatascience.com/10-most-popular-machinelearning-software-tools-in-2019-678b80643ceb (Accessed: 21 October 2024).
4. The Science and Information (SAI) Organization. (n.d.). *Convolutional Neural Networks with Transfer Learning for the Detection of Pneumonia in Chest X-rays*. Available at: https://thesai.org/Downloads/Volume13No9/Paper_63-Convolutional_Neural_Networks_with_Transfer_Learning.pdf (Accessed: 21 October 2024).
5. Centers for Disease Control and Prevention (2023). *Pneumonia Can Be Prevented - Vaccines Can Help*. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10518240/ (Accessed: 21 October 2024).
6. *Time to recovery from severe pneumonia and its predictors among children under-five years*. *Pediatric Health, Medicine and Therapeutics*, 15, pp. 101-115. (2024)