A Technical Study on Pneumonia Detection using CNN and InceptionV3

Gauri Vikram Purandare (U56067938)

***Abstract*—Pneumonia is an infection that particularly affects the lungs. It can start from very basic cold or cough but can convert into a malignant form. Thus, the early detection of the same becomes of prime importance. The technical study focuses on the healthcare domain in the field of cyber-physical systems. Taking into consideration the dataset provided by the US National Institutes of Health Clinical Center, the goal of this project is to pre-process the images using data augmentation, execute the neural network frameworks such as convolutional neural networks and InceptionV3 and present results in the form of accuracies.**

***Index Terms*—Convolutional neural network, InceptionV3**

I. Introduction and Motivation

PNEUMONIA is an infection that is caused when foreign materials or fluids enter the lungs. The infection can start from a normal cough or cold and can prove to be fatal. It can be caused due to a variety of bacteria and fungi. As per the official records of the (CDC) Centers for Disease Control and Prevention [1], the number of deaths caused due to pneumonia totals up to a value of approximately 48,000. Also, in 2017, according to the records of the World Health Organization (WHO) [2], more than 800,000 children under the age of 5 were killed due to pneumonia.

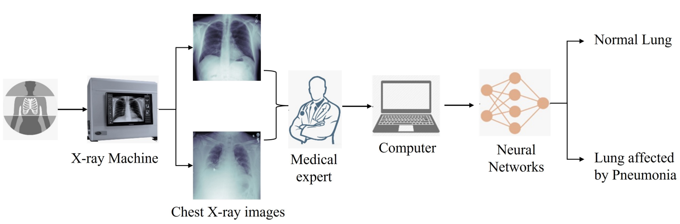

Fig. 1. The figure shows the block diagram of the cyber physical system involved in this project. The X-ray machine equipped with X-ray tubes, detector, sensors and cameras constitute the physical components of the system. On the other hand, the transfer of images and implementation of neural networks using a network fabric and computer constitute the cyber component of the system.

This project is a cyber-physical system from the healthcare domain. Generally, a cyber-physical system consists of physical and cyber components. The physical part of the system comprises the mechanical parts or human operators. The cyber part of the system is a combination of the computational platforms such actuators or any computers and the network fabric. Here, as seen in figure 1, the X-ray machine consisting of X-ray tubes, detectors, sensors and cameras form the physical part of the system. On the other hand, the transfer of captured images and the implementation of neural network using a computer form the cyber part of the system.

II. Background

Wang and his team [3] presented a study where they derived a new database namely ”ChestX-ray8”. A total of 108,948 chest X-ray images of 32,717 patients were collected. These images were then assigned 8 different labels according to the severity in them. The labels were namely Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia and Pneumothorax. If none of these categories were found, the images were termed as normal. The dataset was then fed into a multi-label CNN model. The authors implemented many models such as ResNet, GoogleNet, VGGNet and AlexNet. The models included pre-trained ImageNet weights. The performance metrics were measured in terms of specificity and AUC curve. Amongst all the algorithms that were implemented, the authors concluded that ResNet50 demonstrates the best results.The results were displayed in the form of accuracies.

Chouhan along with his teammates [4] utilized the dataset provided by the Guangzhou Women and Children’s Medical Center. The data was passed through multiple transfer learning approaches such as AlexNet, ResNet8, InceptionV3, DenseNet21 and GoogleNet and classified into three main categories or classes. The classes were “Normal”, “Bacterial Pneumonia” and “Virus Pneumonia”. After the prediction of each algorithm is done, the authors used an ensemble classification technique. The ensemble classification method takes the predictions of all algorithms in the form of a prediction vector, conducts a voting and finally gives a result. The authors used an Intel computer for the implementation. Finally, the authors determined the accuracies, AUC values and execution times. ResNet18 achieved a test accuracy of 94.23% and AUC value of 99.36%. Also, the execution time of all models using the GPU was the lowest at 0.043 seconds while that on CPU took 0.332 seconds. On the other hand, the ensemble technique took a computational method of 0.161 seconds.

Another technical study was presented by Varshni and her team[5]. The authors of this paper used a DenseNet169 approach for feature extraction and a SVM algorithm for the purpose of classification. The images were passed through a convolutional, single pooling layer, three transition layers and four dense blocks to extract the features. Then the support vector machine algorithm was implemented to classify the images into whether the presence of pneumonia is seen or not. One of the limitations that the authors stated was the need of high computational power since multiple convolutional layers are deployed in this model.

III. Technical Approach and Results

This section comprises various subsections including the discussion about the dataset that was taken into consideration, the image pre-processing techniques that were deployed, implementation of CNN and InceptionV3 algorithms and display of results.

A. Dataset

For the purpose of this project, the dataset was derived from the US National Institutes of Health Clinical Center[6]. The NIH put together the chest X-ray records of more than 30,000 patients and made it available to the common public for research purposes. The identity of the patients were kept anonymous. The patients who had contributed towards this dataset had voluntarily participated in this program. To be specific, the dataset consists of 112,120 chest X-ray images of 30,085 patients.

Normal Lung | Not Normal Lung but not affected with Pneumonia | Lung affected with Pneumonia

Fig. 2. The figure shows the images of chest X-rays. Image (a) depicts a normal lung. Figure (b) shows a lung that is not normal but the patient is not affected by pneumonia while (c) shows a lung affected by pneumonia.

When the images are captured through the X-ray machine, the X-ray tube emits the rays that pass through the lungs. The X-rays pass through the lungs and reach the detector on the other side. As seen in the figure 2, soft tissues such as lungs with air do not absorb the X-rays and thus appear to be black in the image. The black part of the image depicts bones and the grey material shows the fluids or any other foreign material inside the lung.

B. Technical Approaches

Before the implementation of neural networks, data augmentation techniques were applied on the dataset. Data augmentation is a technique of creating artificial training data from the existing one. The data augmentations include transformations such as shifts, denoising, random flips, zoom, brightness range, rotations and many more.

As shown in the image, the project deployed many data augmentation techniques such as rotations, zoom, shifts, flips, ZCA whitening and normalization. For rotation, a source image is randomly rotated clockwise or counterclockwise by some number of degrees. A rotation augmentation randomly rotates the image clockwise by a given number of degrees from 0 to 360. A zoom augmentation randomly zooms the

```
train_datagen = ImageDataGenerator(validation_split=0.2,
                                   rotation_range=2,
                                   width_shift_range=0.1,
                                   height_shift_range=0.1,
                                   zca_whitening = False,
                                   zoom_range=0.2,
                                   horizontal_flip=True,
                                   samplewise_center=True,
                                   samplewise_std_normalization=True)
```

Fig. 3. The figure depicts the data augmentation methods that were applied on the original dataset to create new training data.

image in and either adds new pixel values around the image or interpolates pixel values respectively. Since the float value specified for this augmentation is less than 1, it will zoom in the image. The width_shift_range and height_shift_range arguments control the amount of horizontal and vertical shift respectively. The horizontal flip reverses the pixel values column-wise. Also, the ZCA whitening is set to a Boolean value of False and normalization is done. Following the data augmentation, three approaches were utilized to obtain the target variables that are ”Nomral Lung”, ”Not Normal Lung but not affected with Pneumonia” and ”Lung affected with Pneumonia”.

1) CNN without transfer learning:

The convolutional neural network implemented in this project consists of 16 layers. There are 6 convolutional layers. The convolutional layer takes an input from the user. It consists of parameters such as kernel size, number of filters, and the activation function. Here, the Rectified linear unit activation function with a kernel size of 5 and padding was used. In the figure 4, we can see that four pooling layers were used. The pooling layers are utilized to reduce of size of feature maps to increase the efficiency of the model. The dropout layers are used to prevent the model from overfitting while the dense layers is the final or output layer of the model.

We know that the ReLu activation function is used to reduce the gradient problem while the softmax activation layer is used to return the original outputs into a form of probabilities. In the output or the final layer here, the softmax activation function was used.

2) CNN with transfer learning model VGG16:

The VGG16 model is a 16 layer convolutional neural network. Here, the project deploys a VGG16 model in addition to the existing CNN network. In this model, the pre-trained ImageNet weights are applied. The pretrained ImageNet weights are defined by the Keras.

In the figure 5 we can see that ImageNet weights have been applied to the model. In addition to this, two hidden layers and one softmax layers have been added. The model was iterated for 20 epochs along with the Adam optimizer. Early stopping mechanism was also implemented in this project. Basically, early stopping is a technique that prevents the model from training itself even though there is no change in the parameters

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 128, 128, 32)      2432

 conv2d_1 (Conv2D)           (None, 128, 128, 32)      25632

 max_pooling2d (MaxPooling2D  (None, 64, 64, 32)       0
 )

 dropout (Dropout)           (None, 64, 64, 32)        0

 conv2d_2 (Conv2D)           (None, 64, 64, 64)        18496

 conv2d_3 (Conv2D)           (None, 64, 64, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 32, 32, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 32, 32, 64)        0

 conv2d_4 (Conv2D)           (None, 32, 32, 128)       73856

 conv2d_5 (Conv2D)           (None, 32, 32, 128)       147584

 max_pooling2d_2 (MaxPooling  (None, 16, 16, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 16, 16, 128)       0

 global_max_pooling2d (Globa  (None, 128)              0
 lMaxPooling2D)

 dense (Dense)               (None, 256)               33024

 dropout_3 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 3)                 771

=================================================================
Total params: 338,723
Trainable params: 338,723
Non-trainable params: 0
_________________________________________________________________
```

Fig. 4. The figure shows the model summary of the CNN model without transfer learning. It is a 16 layer model which consists of convolutional, pooling, dropout and dense layers.

```
from tensorflow.keras import layers, models

flatten_layer = layers.Flatten()
dense_layer_1 = layers.Dense(50, activation='relu')
dense_layer_2 = layers.Dense(20, activation='relu')
prediction_layer = layers.Dense(3, activation='softmax')


cnn_VGG16_model = models.Sequential([
    base_model,
    flatten_layer,
    dense_layer_1,
    dense_layer_2,
    prediction_layer
])

cnn_VGG16_model.summary()
```

```
from tensorflow.keras.callbacks import EarlyStopping

cnn_VGG16_model.compile(
    optimizer='Adam',
    loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
    metrics=['accuracy'],
)

## Early stopping whe validation accuracy does not change for 7 iteration
es = EarlyStopping(monitor='val_accuracy', mode='auto', patience=7,  restore_best_weights=True)

#Trainign the model
history = cnn_VGG16_model.fit(train_ds, y_train, epochs=20, validation_data=(train_val_df,y_val) ,callbacks=es)
```

Fig. 5. The figure depicts the snippets of the code showing the ImageNet weights being applied. In addition to that, the early stopping technique was deployed.

of the model such as validation accuracy. This parameter can be adjusted by the user.

3) InceptionV3:

Traditionally, Inception is an algorithm that has three versions. The versions are InceptionV1, V2 and V3. The main disadvantage of the InceptionV1 was the high level of computational complexity. To tackle this issue, V2 and V3 were introduced. The InceptionV2 algorithm highly complex layers into smaller and less complex layer to reduce the computational cost. However, this led to decrease in the accuracy of the model. Furthermore, InceptionV3 was developed with some changes.

InceptionV3 is a advanced convolutional neural network algorithm that is used in image classification. It is a 42 layer network that deploys multiple convolutional and pooling layers of different sizes. The main advantages of using this approach is to increase the accuracy of the model, and prevent overfitting of the model.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 300, 300, 3)	0	
conv2d (Conv2D)	(None, 149, 149, 32)	864	input_1[0][0]
batch_normalization (BatchNorma	(None, 149, 149, 32)	96	conv2d[0][0]
activation (Activation)	(None, 149, 149, 32)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 147, 147, 32)	9216	activation[0][0]
batch_normalization_1 (BatchNor	(None, 147, 147, 32)	96	conv2d_1[0][0]
activation_1 (Activation)	(None, 147, 147, 32)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 147, 147, 64)	18432	activation_1[0][0]
batch_normalization_2 (BatchNor	(None, 147, 147, 64)	192	conv2d_2[0][0]
activation_2 (Activation)	(None, 147, 147, 64)	0	batch_normalization_2[0][0]
max_pooling2d (MaxPooling2D)	(None, 73, 73, 64)	0	activation_2[0][0]
conv2d_3 (Conv2D)	(None, 73, 73, 80)	5120	max_pooling2d[0][0]
batch_normalization_3 (BatchNor	(None, 73, 73, 80)	240	conv2d_3[0][0]
activation_3 (Activation)	(None, 73, 73, 80)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 71, 71, 192)	138240	activation_3[0][0]

Fig. 6. The figure shows the model summary of InceptionV3 algorithm. This algorithm consists of multiple convolutional as well as pooling layers. It also consists of multiple dropout layers and a dense layer.

The figure 6 shows the model summary of this algorithm. The model was iterated for 50 epochs with a batch size of 32. The model is compiled with a RMSProp optimizer and batch normalization is also done.To prevent overfitting, the label smoothing parameter was included. After the implementation of this algorithm, results were obtained in the form of accuracy.

C. Results

After the implementation of the above-mentioned three algorithms, the model was evaluated and the accuracy and loss graph was plotted. In addition to this, the test accuracy was also determined.

The figure 7 shows the graphs for accuracies and loss. The test accuracy was also computed. We can see that the test accuracy of the CNN without transfer learning model is quite low at a value of approximately 45%.

Moving ahead with the introduction of a transfer learning approach VGG16 along with CNN we can see in the figure 8 that the test accuracy has increased to approximately 65%. This shows that the introduction of a transfer learning approach increases the efficiency of the model in classification of data into the required categories.

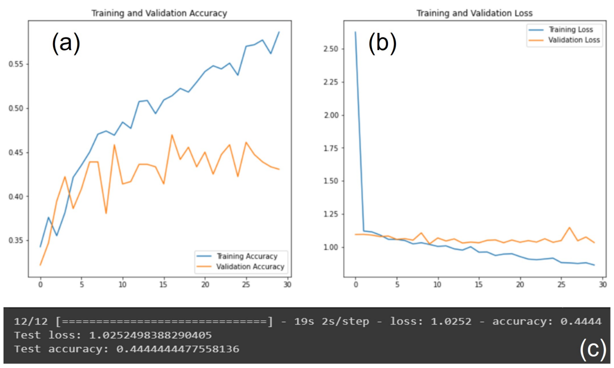

Fig. 7. The graphs shows the train and validation accuracies, losses and the test accuracy respectively. (a) This graphs plots the accuracy with respect to the number of epochs. (b) This graph shows the train and validation loss. (c) The image shows the test loss and test accuracy of the CNN model without transfer learning.

```
12/12 [==============================] - 71s 6s/step - loss: 0.7708 - accuracy: 0.6472
Test loss: 0.7708384394645691
Test accuracy: 0.6472222208976746
```

Fig. 8. The test loss and accuracy was computed for the CNN with transfer learning VGG16 approach.

Furthermore, the implementation of InceptionV3 algorithm generated the train, validation and test accuracies. The test accuracy for this algorithm was found to be approximately 89% as shown in figure 9.

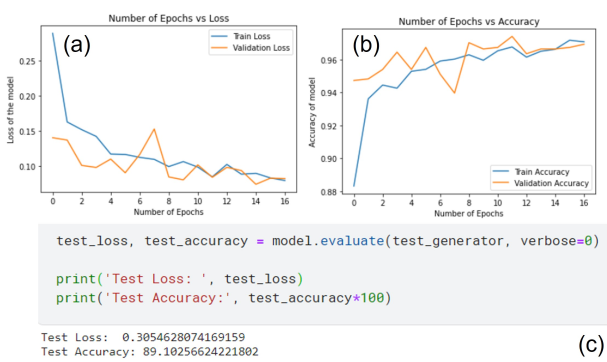

Fig. 9. The figure (a) shows the trade-off between number of epochs and accuracies while (b) shows the trade-off between loss and number of epochs. (c)The test accuracy and loss is computed and displayed.

References

[1] "Centers for disease control and prevention." [Online]. Available: https://www.cdc.gov/nchs/fastats/pneumonia.htm

[2] "World health organization." [Online]. Available: https://www.who.int/health-topics/pneumoniatab=tab$_1$

[3] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2097–2106.

[4] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. De Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest x-ray images," *Applied Sciences*, vol. 10, no. 2, p. 559, 2020.

[5] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, and A. Mittal, "Pneumonia detection using cnn based feature extraction," in *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*. IEEE, 2019, pp. 1–7.

[6] "United states national institutes of health clinical centers pneumonia dataset." [Online]. Available: https://nihcc.app.box.com/v/ChestXray-NIHCC