A Report on Chest Opacity Classification using Deep Convolutional Neural Network

Grace Kwagalakwe
Department of Computer Science
Makerere University
kwagalakwegrace10@gmail.com
2021/HDO5/2302U

Abstract

Chest opacities are a major health problem worldwide; failure of early detection and treatment may lead to significant morbidity and mortality. In addition, chest diseases are a great burden to the individual suffering, their attendants and the society at large. Chest diseases are among the ten leading causes of death, ranking the second in Africa. There is therefore need for models or tools based on Ultrasound images, that helps in the diagnosis and management of chest opacities. while overcoming the limitations of X-ray and CT scan-based methods. This project is undertaken to develop a deep learning model (Convolutional Neural Network Model) for classifying ultra sound Chest images, to evaluate the model performance and as well assess how well the model can generalize on images captured in different environments. I used the Convolutional Neural Networks (CNN) algorithm because CNN-based deep learning classification approaches have the ability to automatically extract high-level representations from data using little preprocessing. The results show that the model performs well in predicting images captured in the same conditions or environments with an accuracy score of 97 percent but performs fairly in predicting images captured in different environments with an accuracy score of 56 percent. The model performance on images captured in different environments can be improved by applying more preprocessing techniques like enhancing the contrast of the images as it was observed that its more likely the images were captured in low lighting environments. The use this model can go a long way towards ensuring proper detection of chest opacities and thus reduce related risks of morbidity and mortality.

Introduction

Chest opacities are a major health problem worldwide; failure of early detection and treatment may lead to significant morbidity and mortality. In addition, chest diseases are a great burden to the individual suffering, their attendants and the society at large. Chest diseases are among the ten leading causes of death, ranking the second in Africa. Chest X-ray is the first line procedure to assess chest conditions but interpretation of radiologic sign such as vascular opacity redistribution and interstitial edema are often questionable and subjective. Moreover, X-ray is 2 dimensional hence one may not accurately tell the nature of these opacities and localization of the same is not definitive; even with established guidelines for interpretation, chest X-ray has demonstrated to be an insensitive method with relatively low accuracy. This delays establishment of the cause of chest opacity hence mismanagement of the patient leading to more costs for hospitalization and even death. CT scan has been determined as the gold standard to characterize, localize and identify the nature of opacities. However, it is expensive, not always available, uses ionizing radiation and transferring critically ill patients to the CT room is complicated.

In this project we use deep learning to train a model, based on Ultrasound images, that helps in the diagnosis and management of chest opacities while overcoming the limitations of X-ray and CT scan-based methods. The method has potential to accurately localize opacities, differentiate between opacities and suggest appropriate further investigations. Ultrasound is easily available, uses no ionizing radiation, is relatively cheaper and portable. This study explores data sets with chest images to be classified whether normal or sick using a deep learning model. The performance the model was evaluated on dataset1. In addition, an evaluation on dataset two having images captured in different environment was also conducted.

The rest of the report is organized as follows, in the next section the objectives of the project, overview on CNN and related works are presented, the materials and methods used to develop the model, results and discussion are then presented and thereafter a conclusion drawn.

Objectives

- To develop a CNN deep learning model capable of detecting and classifying chest images whether normal or sick.
- To evaluate the model performance.
- To use the model to predict the classes of the unknown images

- To test the model on dataset 2 and see how well it generalise with images captured in different conditions.

Convolutional Neural Network Overview

Convolutional neural network (ConvNets or CNNs) is one of the main categories to do image recognition, images classifications, Objects detections, recognition faces etc.

CNN is a special type of Feed-Forward Artificial Neural network, where neurons receive input for transforming to hidden layers. Every hidden layer is fully connected to all neurons of the previous layer. It has four main layers:

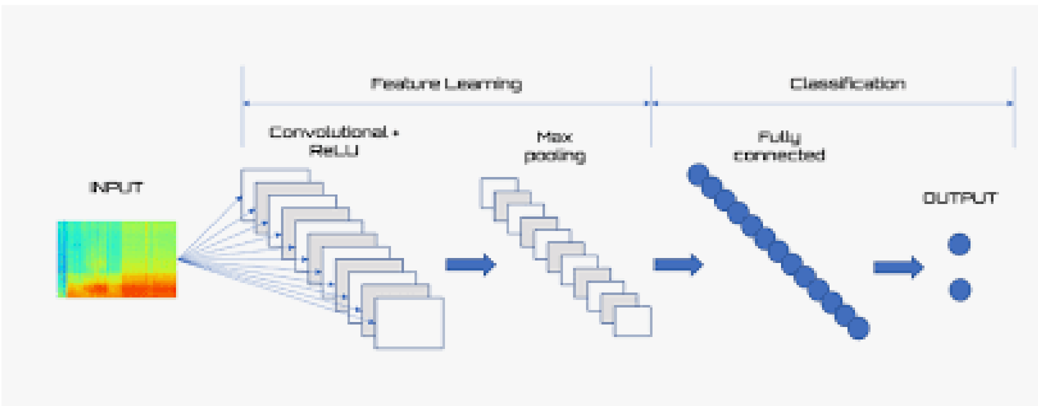

Fig1: Convolutional Neural Network Architecture

1. Convolution Layer: Convolution is the first layer to extract features from an input image. The convolutional layer uses a matrix of filters over an array of image pixels.
2. Rectified Linear Unit: The purpose of the rectifier function is to increase the non-linearity of the network. Like sigmoid function. The rectifier function removes all the black elements, keeping only positive value. Negative pixels to zero.
3. Pooling Layer: It basically reduces the dimensionality of the feature map or reduce the number of parameters when the images are too large. The max pool takes the largest item from the revised feature map.
4. Fully Connected Layer: The output of the last Pooling Layer acts as input for the Fully Connected Layer. The output of both convolution and pooling layers are 3D volumes (Matrix), but a fully connected layer expects a 1D vector of numbers (Vectors). So, smoothing the result of the last vector merge layers.

Related works

From various researchers and papers, related studies have been made over the years to do with the diagnosis of opacities using deep learning models, In the early years, a CAD system was built using traditional machine learning methods. This would select and extract functions from the input manually and it was based on the criteria considered. This was basically preliminary deep learning. Advancements in the method were done to bring a significant change in the way the model interacts and learning to autonomously detect complex patterns and functions directly from input data. (Antor, 2020)

Later on, in the field of computer vision, an important milestone was achieved when, the ImageNet deep learning model for large-scale visual recognition, ImageNet, based on convolutional networks, and surpassed the performance achieved with other machine learning methods.
Recent improvements in deep learning models and having huge data sets helped algorithms outperform medical staff in many medical imaging tasks to do with classification. This makes the automated diagnoses to trend highly. [1]

Deep convolutional neural networks (CNNs) have proven to be powerful tools for a wide range of computer vision tasks, predominantly driven by the emergence of large-scale labeled datasets and more powerful computational capabilities.(Johnson et al., 2021). Pioneering work in computer-aided diagnosis on chest radiographs mainly focused on a specific disease (e.g., pulmonary tuberculosis classification13, lung nodule detection). The recent release of the large-scale datasets, such as "NIH ChestX-ray (which is an extension of the eight common disease patterns in "NIH ChestX-ray 8"), "CheXpert" 17 and "MIMIC-CXR" 18, have enabled many studies using deep learning for automated chest radiograph diagnosis. [3]

Materials

Google colab notebook was utilized to develop the CNN model and all the necessary libraries imported in the notebook.
Two datasets were used for the project, i.e. dataset1 and Dataset2 and these were provided by the Instructor for the course. The two datasets consisted of images for two classes, normal and sick. Dataset1 contains 715 known images that were organised in two classes (normal or sick) and 50 image files whose class is unknown. Dataset two consists of 224 image files that were also organised in two classes (normal or sick).

Methodology

Data Pre-processing

Data pre-processing refers to manipulation or transformation of raw data before it is used for further training in order to ensure or enhance performance. The pre-processing step is an important element in the image analysis schema. It can enhance the original image and reduce noise or unwanted details.

Firstly, the known images of dataset one were organised in two different folders (training, testing) and each of these folders contained two subfolders each belonging to the image classes being examined. a train-test split ratio of 0.8:0.2 was used to populate the folders with images.

Dataset two was also organised in two folders each belonging to the image classes.

The rest of the pre-processing techniques were done on google co-laboratory using colab Notebook, so the organised folders were uploaded onto the google drive. The google drive mounted so that the content can be accessed from the colab notebook.

In the colab notebook, all the necessary libraries needed for pre-processing and training were imported. Classes of images were set and also a uniform size set for all images. A method/function for loading the images and then pre-processing them was defined. In the method; images and labels were loaded in different sections and mainly two pre-processing techniques were done with iterations made through each image in the folders.

1) Resizing the images to a uniform size of (150,150),
2) Converting the images to gray scale (using cv2).

Images were then appended to their corresponding labels, converted images and labels to numpy arrays to make it easier to process them. A directory path for loading the images from the folder in the drive was set and the previously defined method for loading and pre-processing them was called. Both the training and testing set of images were loaded with their labels. The images were then shuffled using a random state of 25 and a sample of them displayed as shown in fig 2.

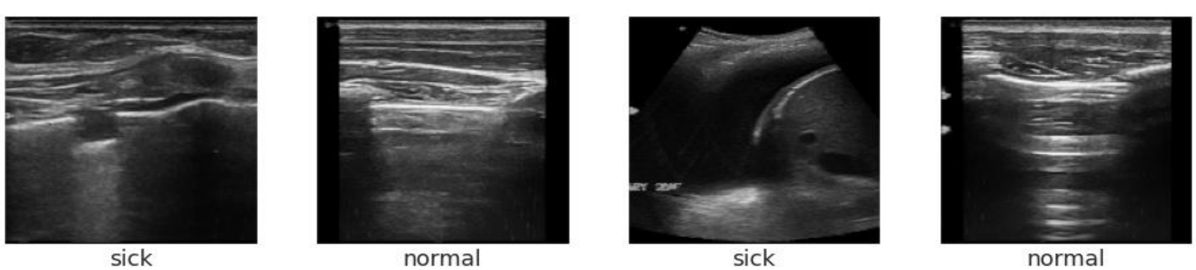

Figure 2: Sample of the loaded images.

Model Training

A multi-layered CNN model was developed. The model has three convolution2D layers each with a kernel size of 3*3 and a Relu activation that introduces non-linearity. The three layers extract different number of features where the first layer extracts 16 features, the second extracts 32 features and the third extracts 64 features.

A maxpool kernel size of 2*2 is added on each Conv2D layer to reduce Dimensionality and preserve spatial invariance on the output of the conv2D layer.

The output is then flattened into a single vector. A dense layer is then applied to classify the output of the single vector.

A final Dense layer with a softmax activation function is applied to classify two classes.

The model has a total of trainable parameters of 9,495,074 and zero non-trainable parameters. The summary of the model architecture is as shown in figure 3.

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 16)      448

 max_pooling2d (MaxPooling2D  (None, 74, 74, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 64)       0
 2D)

 flatten (Flatten)           (None, 18496)             0

 dense (Dense)               (None, 512)               9470464

 dense_1 (Dense)             (None, 2)                 1026

=================================================================
Total params: 9,495,074
Trainable params: 9,495,074
```

Figure 3: Model summary

The model was then compiled using the sparse-categorical-cross entropy for loss, RMSprop

Optimizer with a learning rate of 0.001 and accuracy metrics. The model was fit using a validation split of 0.2 and 30 epochs.

Evaluation of the model

On train set

Accuracy, validation accuracy, loss and validation loss were used to evaluate the performance of the model on the training set of images. The train results were visualized in a graph plot that shows the accuracy and loss during the training at the different epochs. The plot was as shown in figure 4.

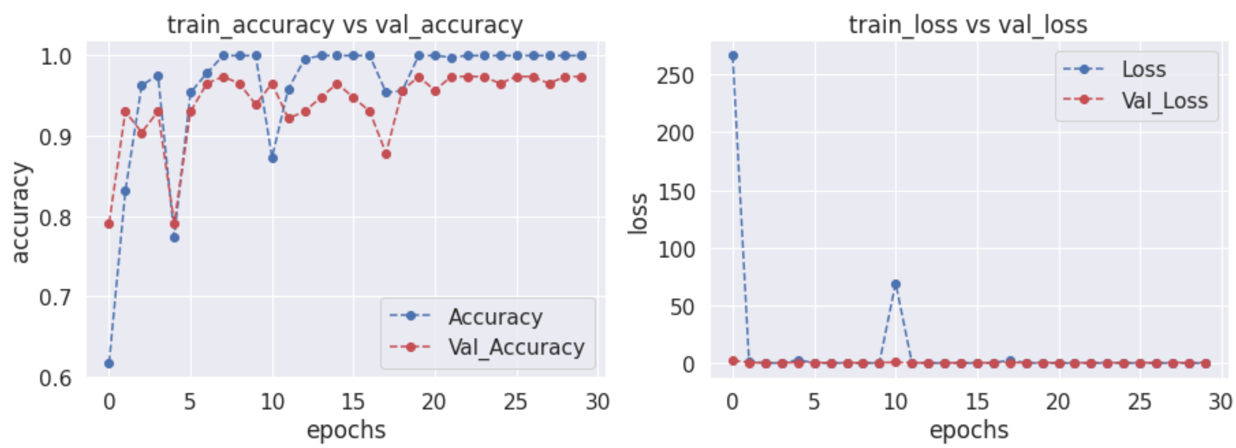

Figure 4: Accuracy_loss plot.

On test set

The model was also evaluated on unseen data using the test split. An accuracy score of 97 percent was attained. A classification report showing more metrics such as precision, recall and f1 score was also printed and the details were as shown in figure 5.

	precision	recall	f1-score	support
0	0.96	0.99	0.97	74
1	0.99	0.96	0.97	69
accuracy			0.97	143
macro avg	0.97	0.97	0.97	143
weighted avg	0.97	0.97	0.97	143

Figure 5: Classification report for dataset1 performance evaluation.

A confusion matrix showing more details of true positives, true negatives, false positives and false negatives was also visualized. The model was saved as a .h5 file.

Predictions on unknown images.

The directory path for the unknown images was set. A function that reads each image in the folder was defined and each image read and resized to the previously set image size.

The images were converted to an array and their dimension set using the expand dims function from the numpy array library.

The classes of the images were then predicted basing on the probabilities assigned to each image. belonging to normal or sick class. The probabilities were rounded off to two decimal places.

Each image was assigned two probabilities in a two-dimension array where the first row is for the probability of belonging to the normal class and second row is for probability of belonging to the sick class. The image assumed the class for the row with the highest probability.

The images were then imported to a csv file with their predicted classes.

Evaluation on dataset2

The images in dataset2 were also organized in two folders one for the sick and the other for the normal. The images were uploaded onto the google drive.

A method for loading and preprocessing the image that was defined before was then called to load the images for dataset two via their directory path on the drive.

A sample of the images was also displayed using the previously defined function use on dataset one.

The performance of the model on dataset two images was evaluated and an accuracy of 0.56 was obtained. The scores of other metrics like precision and recall are as shown in the classification report in figure 6.

A confusion matrix that indicates the true positives, false positives, true negatives and false negatives was also generated.

	precision	recall	f1-score	support
0	0.93	0.12	0.22	112
1	0.53	0.99	0.69	112
accuracy			0.56	224
macro avg	0.73	0.56	0.46	224
weighted avg	0.73	0.56	0.46	224

Figure 6: Classification report for dataset2 performance evaluation.

Discussion of results

The performance of the model was mainly evaluated using four metrics; Accuracy, precision, recall and f1 score and these were printed in the classification reports as seen from figure 4 and figure 5.

Accuracy is the ratio of correctly predicted observation to the total observations. It generally describes how well the model performs across all classes.

$$\text{Accuracy} = \frac{TruePositive+TrueNegative}{TruePositive+TrueNegative+FalsePositive+FalseNegative}$$

Precision is the ratio of the number of positive sample correctly classified to the total number of sample classified as positive. It specifies the proportion of positive identifications that are actually correct.

$$\text{Precision} = \frac{TruePositive}{TruePositive+FalsePositive}$$

Recall is the ratio of the number of positive samples correctly classified to the total number of positive samples. It specifies the proportion of actual positives identified correctly.

$$\text{Recall} = \frac{TruePositive}{TruePositive+FalseNegative}$$

The F1 score is a weighted harmonic mean of precision and recall.

Support is the number of actual occurrences of the class in the specified dataset.

Performance on dataset 1

The model performs generally well on dataset1 with an accuracy score of 98% as represented in the classification report for dataset1(figure 4). The model classifies well both the normal class and the sick class.

Figure 7 represents a confusion matrix obtained from the performance of the model on dataset1. The model wrongly predicted 4 images and correctly predicted 139 images which is generally a good prediction with less errors.

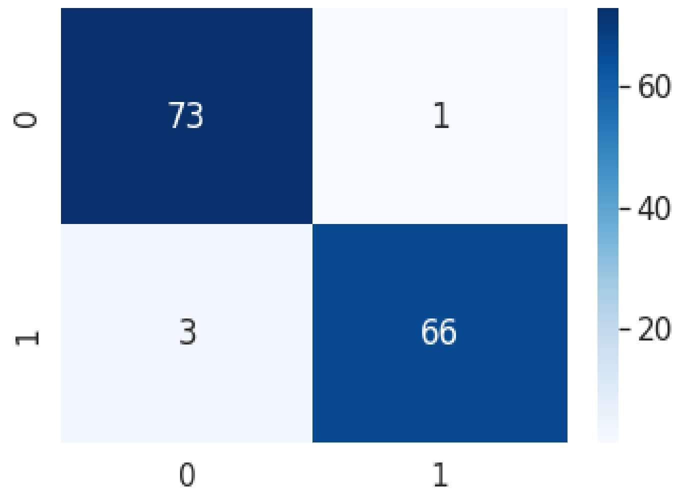

Figure 7: Confusion matrix for performance results of dataset 1.

The confusion matrix values were used to calculate for precision, recall and accuracy and results compared with that generated in the classification report.

Accuracy = $\frac{73+66}{73+66+1+3}$ = 0.9720

Precision for normal class(0) = $\frac{73}{73+3}$ = 0.9605 ≈ 0.96

Recall for normal class(0) = $\frac{73}{73+1}$ = 0.9894≈ 0.99

Precision for sick class(1) = $\frac{66}{66+1}$ = 0.9850 ≈ 0.99

Recall for sick class() =$\frac{66}{66+3}$ = 0.9565≈ 0.96

The computed results and that generated in the classification report are equivalent.

Performance on dataset 2

The model performs fairly on dataset two with an accuracy score of 56% classifying the normal class better than the sick class.

Figure 8 represents a confusion matrix obtained from the performance of the model on dataset2. The model wrongly predicted 99 images and correctly predicted 125 images which is fairly a good prediction.

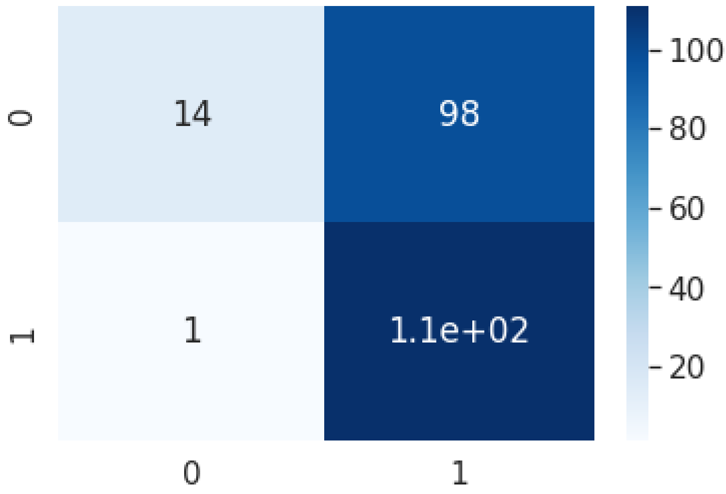

Figure 8: Confusion matrix for performance results of dataset 2.

The confusion matrix values were used to calculate for precision, recall and accuracy and results compared with that generated in the classification report.

Accuracy = $\frac{14+111}{111+98+1+14}$

= 0.5580

Precision for normal class(0) = $\frac{14}{14+1}$ = 0.9333 ≈ 0.93

Recall for normal class(0) = $\frac{14}{14+98}$ = 0.125

Precision for sick class(1) = $\frac{111}{111+98}$ = 0.5311≈ 0.53

Recall for sick class(1) = $\frac{111}{111+1}$ = 0.9910≈ 0.99

The computed results and that generated in the classification report are equivalent.

The model performance is fair due to the fact that the images might have been captured in low lighting conditions and so features are hard to be extracted, detected and be predicted.

Conclusion and Recommendations

Chest opacities are a major health problem worldwide. In this project, we have proved that deep learning models for example Convolution Neural networks can be applied in the diagnosis and management of chest opacities to improve on the existing methods and also to overcome the limitations of X-ray and CT scan-based methods.

Experimentation was done using RMSprop optimizer and gave an accuracy of 97 percent. We recommend use of other optimizers like Adam and then compare the results.

We also recommend applying more image preprocessing techniques like augmentation, balancing the dataset and also compare and contrast their effect on the performance of the model.

For dataset two, we recommend applying more preprocessing techniques like enhancing the brightness of the images to increase chances of detecting, extracting and learning the features

References

[1] Antor, M. H. (2020). Lung Opacity Identification Using Mathematical Model Based on Deep Learning. *International Journal of Engineering Applied Sciences and Technology*, *5*(5), 25–29. https://doi.org/10.33564/ijeast.2020.v05i05.005

[2] Hao, Y. (2021). Convolutional Neural Networks for Image Classification. *Proceedings - 2021 2nd International Conference on Artificial Intelligence and Computer Engineering, ICAICE 2021*, 342–345. https://doi.org/10.1109/ICAICE54393.2021.00073

[3] Johnson, B. K., Eden, A., Reinecke, L., & Hartmann, T. (2021). Self-control and need satisfaction in primetime: Television, social media, and friends can enhance regulatory resources via perceived autonomy and competence. *Psychology of Popular Media*, *10*(2), 212.

[4] Classification: Precision and Recall: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall. Accessed on 2nd September, 2022.

[5] Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall by Ahmed Fawzy Gad: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/. Accessed on 2nd September, 2022.

[6] Understanding a Classification Report for Your Machine Learning Model by Shivam Kohli: https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397 . Accessed on 2nd September. 2022.