Unveiling Insights: Classifying X-ray Images of Lung Pathologies

A Data-Driven Apsproach to Identifying COVID-19, Normal, and Viral Pneumonia Cases

May 5, 2024

Outline

Introduction

- The project proposes the development and experimentation of Machine Learning models for the identification and classification of x-ray images of lung pathologies.
- Aim: to distinguish between COVID-19, normal, and viral pneumonia cases.

Objectives

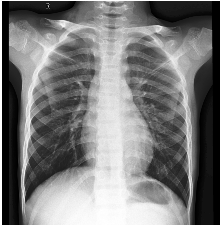
- Explore the methodologies behind X-ray image classification.
- Present insights gained from our data-driven approach.

Data

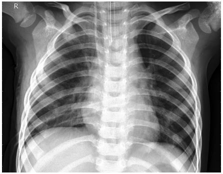
- The images are taken from the database of the University of Montreal which was trying to build a classifier through medical experts.
- Each X-ray image is labelled with one of the following categories:
    - Normal
    - Covid19 infected
    - Viral Pneumonia infected
- We perform classification of the images into these categories using supervised learning techniques.

Data Preparation

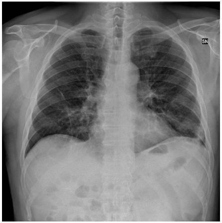
Sample Images



Normal X-ray



Viral Pneumonia



Covid-19

Data Preprocessing

- Normalize pixel values to a standardized range to ensure uniformity across images.
- Augment the dataset with techniques such as rotation, flipping, and scaling to increase model generalization.

Fully Connected Neural Network

Model Architecture:

- Design a fully connected NN using Tensorflow's Keras API.
- 3 layers with 512, 256 and 128 unit, each with ReLU activation.
- Implemented dropout to prevent over-fitting.

Implemented Models- CNN

Model Architecture:

- Designed a CNN architecture comprising convolutional layers for feature extraction followed by fully connected layers for classification.
- The convolutional layer consists of 32 filters of size (3,3), and uses ReLU activation function.
- Each convolutional layer is followed by a max-pooling layer with a pool size of (2, 2).
- Fully connected layers with dropout and ReLU is implemented for classification.

Training and validation

- Trained the DNN and CNN models with Adam and RMSprop optimizers respectively, with categorical cross-entropy loss function.
- Monitored training progress through metrics such as accuracy and loss on the test set.
- Evaluated the trained model on the test set to assess its performance on unseen data.
- Classification performance is analyzed using accuracy.

Utilizing pre-trained models

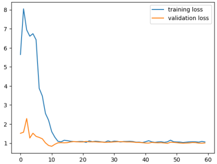
Model Architecture:

- Utilized a pre-trained VGG16 model with ImageNet weights, excluding its fully connected layers.
- Added global average pooling and dense layers for task-specific classification.

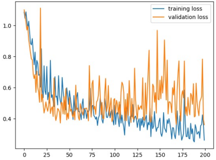
Fine Tuning:

- Fine-tuned the model by freezing the pre-trained weights while adapting the fully connected layers for the specific classification task.
- Sparse categorical cross-entropy loss and Adam optimizer for training.
- Employed early stopping and model check-pointing to prevent over-fitting and save the best performing model.

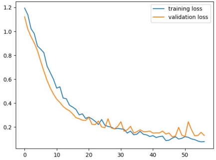
Loss Value Graph



DNN loss

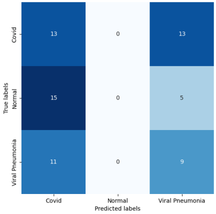


CNN loss

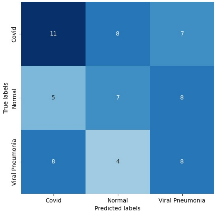


VGG-16 loss

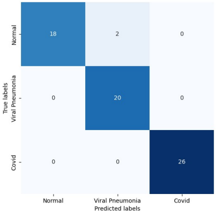
Test set performance



DNN heatmap



CNN heatmap



VGG-16 heatmap

Results

Table: Comparison of the performance of different models

	Training Accuracy	Test Accuracy
DNN	61.7	59.1
CNN	88.8	86.8
VGG-16	97.6	97