Milestone 1 - Detection of Pneumonia

Team

Name	Macid	Role	Role Breakdown
Akash Santhanakrishnan	santhana	Model	- Implementing Residual Neural Network (Resnet) - Ensuring input, output and hidden layers mesh together - Compiling and training model
Rochan Muralitharan	muralr3	Evaluation	- Identifying and implementing evaluation strategy - Evaluating model on test set and validation set - Reporting and interpreting performance of the model
Dhruv Thakor	thakord	Data	- Identifying data sampling technique - Gathering data and preprocessing data - Applying required data augmentation

Context

Pneumonia is a possibly life-threatening respiratory infection for vulnerable populations like children, the elderly and people with weak immune systems. Pneumonia is the world's leading cause of death among children under the age of five accounting for 16% of all deaths of children under five, which was approximately 2,400 child deaths a day in 2015. The early and accurate diagnosis of this infection is vital for effective treatment, however we currently rely on traditional methods of radiologists interpreting chest X-rays. This makes detection of pneumonia challenging as this process can be time-consuming and relies on the expertise of the radiologist making the diagnosis. We are aiming to tackle this issue by creating quick and accurate methods of detecting pneumonia (bacterial and viral) through the inspection of chest X-rays. Machine learning can aid in the detection of this disease by utilizing Residual Neural Networks (ResNet), to automate and improve the detection of pneumonia from medical imaging data. By training a predictive model using labeled chest X-ray images, the model can help doctors by providing reliable, fast and scalable support for diagnosis, especially in settings where resources like expert radiologists are limited. An accurate model can aid in a quick treatment and possibly save countless lives who contract pneumonia but do not receive a quick diagnosis. The key to lowering this large number is through a quick diagnosis, and that is what we are trying to accomplish with our machine learning model. Pneumonia is a disease that affects many worldwide and by creating this model, our team can help in lowering the amount of serious cases that lead to countless child deaths around the world.

Dataset

The dataset we will be using is from Kaggle : Chest X-Ray Images (Pneumonia). The number of features will be based on the number of pixels in the grayscale images. Although the size for each image is constant, it is unclear what the actual size is, but the number of features will be the image width multiplied by the image height. In total, there are 5856 samples. The model will attempt to use the max number of samples that is computationally possible on our machines. The images will be resized to 224 by 224 pixels so that all images will be a standard size and will fit through the layers of the ResNet.

Proposed Solution

The proposed solution is predictive, as it aims to use labeled chest X-ray images to predict whether a patient has pneumonia and whether it is viral or bacterial. The pixel values of the images will serve as features, and the number of features depends on the image size, which may change if we resize the images. The target variables are {0: Normal, 1: Pneumonia (Viral), 2: Pneumonia (Bacterial)}. The machine learning technique will be Residual Neural Networks (ResNets), which are effective for image analysis and automatically learn features such as edges, textures, and shapes. Although the dataset already includes a training, test, and validation set, we will create randomized sets for unbiased evaluation. Since the dataset categorizes pneumonia types in file names rather than in separate folders, we will need to organize the data into distinguishable sets. We will resize the images to a standard size (150x150) and scale pixel values to [0, 1]. If there is class imbalance, we may employ oversampling to balance the classes. Performance metrics will include precision, recall, and F1-score, as accuracy may be misleading with imbalanced classes. After training, we will evaluate the model on a held-out test set. This task has been extensively explored in medical image analysis, and transfer learning from pre-trained models like ResNet or DenseNet is a common solution. Instead of using transfer learning, we will build a small ResNet model from scratch. Pytorch will also be used for pre-processing/resizing the images and Mathplotlib will be used for visualization. Evaluation metrics will be calculated using scikit-learn.

References

https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis
https://www.thoracic.org/patients/patient-resources/resources/top-pneumonia-facts.pdf
https://www.cell.com/cell/fulltext/S0092-8674(18)30154-5
https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

Milestone 2 - Detection of Pneumonia

Preprocessing

To preprocess the data, the images were loaded into the model and then converted into tensors. The tensors were then normalized using the ImageNet mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]). Finally, tensors were wrapped as in DataLoaders, one each for the train, test and validation set. The train set consists of 5000 images, the validation set has 232 images and the test set has 624 images. The train and validation set were split from the train images in the dataset. No other preprocessing of the data was performed as there once the images were converted to tensors and normalized, our ResNets could process the data. No specific feature selection strategies were implemented, our CNN model learns the features automatically from the pixel data. No explicit feature correlation removal was done, as the CNN will be able to distinguish the pixel-level features. No explicit data augmentation was applied, only resizing and normalization was performed. One-hot encoding was not hard coded into our model, however the data loader and PyTorch cross-entropy loss are able to handle the integer labelling internally. No pretrained models were used to train our model as a custom ResNet model is used and no pretrained weights are used.

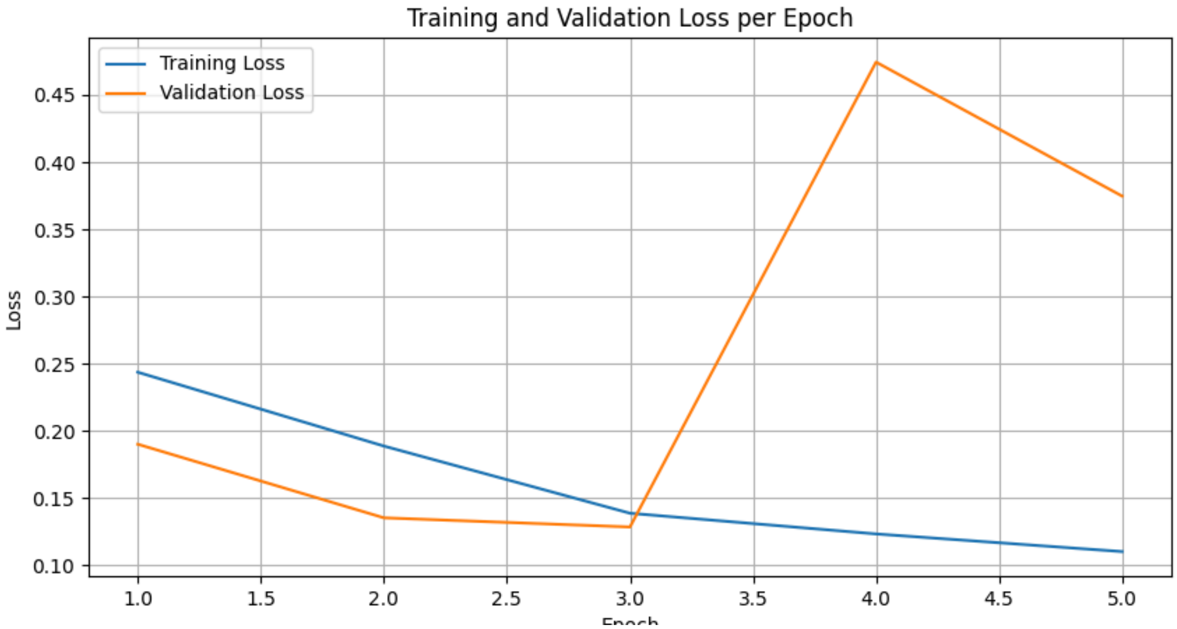
Model Specifications

The model we chose for our project is a Supervised Deep Neural Network which was based on the ResNet architectures. The model configuration consists of the residual block and the ResNet50 architecture. The residual block consists of 3 convolutional layers with skip connections to learn residual mappings. The block expands channels by a factor of 4 by using the final convolutional layer in each block. A batch normalization and ReLU activation function are used after each convolutional layer. The ResNet50 architecture is composed of an initial convolutional layer followed by 4 stages of residual layers with varying numbers of blocks. There are 3, 4, 6 and 3 blocks in each of the 4 layers respectively, creating a total of 50 layers. The initial convolutional layer has a kernel size of 7x7 with a stride of 2. The residual blocks have kernel sizes of 1x1, 3x3, and 1x1 in its 3 convolutional layers respectively. The model implements down-sampling using a stride of 2 for specific layers. An adaptive average pooling function reduces the output of the network to a single value per feature map before the final fully connected layer. The output layer is a fully connected layer with 2 neurons to perform binary classification(Normal vs. Pneumonia). The model's hyperparameters include a learning rate of 0.001, a batch size of 32, and 5 epochs. The Adam optimizer and CrossEntropyLoss function were used for the network as well. We also implemented a progress tracking strategy, which tracked the training loss by computing the cross-entropy loss at every batch interval, and the training and validation loss is recorded after each epoch.

Evaluation

During training, validation loss(CrossEntropyLoss) and accuracy are computed after each epoch to evaluate the training progress. During testing, the average test loss is computed across all batches and the accuracy of the model is also computed.

Preliminary Results



- Test Loss: 0.353
- Test Accuracy: 84.94%
- Test F1-Score: 0.85
- Test Recall: 0.85

Limitations

Our current model focuses on being trained to distinguish between Normal vs Pneumonia input images, however it is not able to distinguish between Viral vs Bacterial which is a current limitation as this was intended in the preliminary implementation. Another limitation is also related to the short training time and limited epochs. We also believe a potential limitation is the possibility of overfitting due to a small validation set, which can be improved by implementing an early stopping algorithm. We believe we can improve our model for further optimization so that it can support the three-class classification and obtain an accuracy higher than 85%. We may also try other ResNet configurations, if computationally possible, such as ResNet101 and ResNet152.

References

https://arxiv.org/pdf/1512.03385
https://www.digitalocean.com/community/tutorials/writing-resnet-from-scratch-in-pytorch
https://www.run.ai/guides/deep-learning-for-computer-vision/pytorch-resnet
https://www.researchgate.net/publication/374812284_International_Journal_of_INTELLIGENT_SYSTEMS_AND_APPLICATIONS_IN_ENGINEERING_Advancements_in_NSFW_Content_Detection_A_Comprehensive_Review_of_ResNet-50_Based_Approaches
https://iopscience.iop.org/article/10.1088/1742-6596/1634/1/012110