CAI 4104/6108: Machine Learning Engineering

Project Report: COVID-19 Detection in Chest X-Rays Using Convolutional Neural Networks

Kondreddy, Rohith Sai Reddy
(Point of Contact)
r.kondreddy@ufl.edu

Athota, Guna Teja
gunateja.athota@ufl.edu

Kogila, Anirudh
anirudhkogila@ufl.edu

Mundada, Harshith
harshith.mundada@ufl.edu

Pasham, Kushi Vardhan Reddy
kushivardhpasham@ufl.edu

April 25,2024

1 Introduction

In the below work we aim to develop a highly accurate system to detect the presence of COVID-19 in Chest X-rays. The task is a supervised binary classification problem, where the goal is to determine whether a given CXR image corresponds to a COVID-19 positive or negative case.

2 Approach: Dataset(s) & Pipeline(s)

Dataset: We are using the COVID-19 Radiography Database dataset created by academics from Qatar University.[1] dataset contains a number of 33,920 chest X-ray (CXR) images, with 10,701 normal images, 11,956 COVID-19 positive images, and 11,263 non-COVID infection, viral or the bacterial pneumonia images. All images are in the Portable Network Graphics (PNG) format with a resolution of 299x299 pixels.

CV/Split: Indeed, our dataset is currently divided into three categories: train, test, and val. Each has a count of 3728, 1166, and 932, in that order.

Data Preprocessing: The dataset is already split into train, validation, and test sets. We will perform standard data preprocessing steps such as resizing the images to a consistent size, normalizing the pixel values.

Model Architecture: Model uses the VGG19 Convolutional Neural Network (CNN) as its base[2], and is customized for a specific multi-class classification task.

- The VGG19 has 19 layers in its backbone; 16 layers are convolutional layers, and 5 layers are the max-pooling stage. The convolutional layers are equipped with ReLU activation and 3x3 kernels to aid in the extraction of the features. The following max-pooling layers reduce the spatial dimensions (2x2 kernel with stride 2). Set the include_top parameter to False so that you can adjust the fully connected layers.
- Dense layer is added on top of the VGG19 base, using a softmax activation function for multi-class classification. The number of neurons matches the output classes in the dataset. The model concludes with a global max-pooling layer, converting feature maps into a manageable size for the dense layer.
- The model is compiled with the Adamax optimizer, with a learning rate(LR) 0.001. Loss function is categorical cross-entropy, and accuracy is being used as the performance metric.
- The training procedure is managed by a custom callback class, which also allows for early halting and adaptive learning rate adjustments. During training, the callback enables user interaction.
- The model is trained with the `fit()` method, with 30 epochs and a batch size of 16. The training data is provided by the `train_gen` generator.

[1] Kaggle link: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database/data

- The architecture has 20,026,436 trainable parameters, indicating that the entire VGG19 base is fine-tuned along with the dense layer.

Pipeline To develop the COVID-19 radiography classification model, we followed the below pipeline.

- We constructed a custom model with VGG19[3] as the base. The input shape was set to 224x224 pixels with 3 color channels (RGB). A dense layer with softmax activation was added for classification, with number of neurons equal to the number of classes in the dataset. Here we used the create_gens function to convert the image data to tensors.

- The Adamax optimizer was used to construct the model with a learning rate of 0.001. The key statistic used to evaluate the model was accuracy, with the categorical cross entropy serving as the loss function.

- The model was trained for 30 epochs with a custom callback for early stopping and learning rate adjustments. The training process involved validation with a separate validation set, and the training history was recorded for later analysis.

- Finally, the model was assessed on the training, validation, and test sets after training with loss_value and accuracy of the modal. Also, the confusion matrix and classification report were generated based on the trained model in labeling the various radiography images

- The trained model was saved to a specified location, along with its weights. A CSV file containing class indices and image sizes was also generated for reference.

3 Evaluation Methodology

We have used the following metrics and plots to develop the model with better performance"

3.1 Training and Validation accuracy and loss tracking

Throughout the model development process, we diligently tracked the training and validation accuracy and loss. This enabled us to get the highlights of the model's learning dynamics and identify potential areas for improvement. By monitoring these metrics, we could assess the efficacy of different architectural choices, optimization algorithms, and training strategies.

Accuracy: The percentage of samples that are correctly classified out of all the samples. **Loss:** The variation between the values that were predicted and the actual values, usually reduced throughout training.

3.2 Custom callback for Dynamic Learning Rate Adjustment

A custom callback is implemented to dynamically adjust the learning rate [1] [4] based on training and validation performance.

Key Components and functionality of the custom callback:

Monitoring Metrics: The callback continuously monitors specific metrics during the training process. Common metrics include training accuracy, along with validation loss, and the validation accuracy.

Thresholds and Parameters: Predefined thresholds and parameters are set to determine when and how the learning rate should be adjusted. These parameters typically include a patience level (the number of epochs to wait before adjusting the learning rate), a stop patience level (the number of consecutive epochs without any of the improvement to stop learning or training), and a threshold for metric improvement.

Dynamic Learning Rate Adjustment: During the training, this callback updates the learning rate dynamically, depending on the monitored metric and predefined thresholds. On the other hand, if a predefined number of epochs (patience) does not see an improvement in the monitored metric (e.g., validation loss), then it will update the learning rate accordingly by some predefined factor. This contributes further to fine-tuning in the optimization process of the model, ensuring not to allow overfitting.

Adaptive Optimization: By adapting the learning rate in response to changes in model performance, the callback helps optimize the training process and make the model's convergence speed faster. This adaptive optimization strategy enhances the model's ability to generalize to unseen data and achieve better overall performance.

Integration with Training Loop: The custom callback is seamlessly integrated into the training loop of the deep learning model. It operates in conjunction with other training components, such as loss functions, optimizers, and model checkpoints, to collectively optimize the model's performance.

This callback continuously monitored key performance metrics such as training accuracy, validation loss, and validation accuracy. Based on predefined thresholds, patience levels, and stop patience parameters, the learning rate was dynamically adjusted to optimize training progress and prevent overfitting. This adaptive learning rate strategy helped improve convergence speed and overall model performance

3.3 Evaluation Metrics

The typical assessment criteria of accuracy, loss, precision, recall, F1-score, and area under the ROC curve are used to assess this architecture. The confusion matrix describes the model's performance in terms of true positives, true negatives, false positives, and false negatives.
Accuracy: The proportion of correctly classified samples among the total number of samples.
Loss: The difference between predicted and actual values, typically minimized during training.
Precision: A measure of a model's ability to prevent false positives, expressed as the ratio of true positive predictions to all positive predictions.
Recall: A measure of the model's capacity to capture every positive case, calculated as the ratio of true positive predictions to the total number of actual positive instances.

F1-Score: The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
Area Under the ROC Curve (ROC-AUC): A measure of the model's ability to distinguish between classes, particularly useful for binary classification tasks.
Confusion Matrix: a tabular display that allows for the understanding of true positives, true negatives, false positives, and false negatives based on the model's predictions versus the actual labels. The produced model's aforementioned metrics were contrasted with those of the baseline model.

3.4 Plotting training history

Training and validation loss/accuracy graphs are plotted to visualize the model's learning progress. The best epoch based on validation loss and accuracy is identified and highlighted in the plots.

We compared the above metrics of model being developed with the findings of implementation of basic VGG19 model.

Since the data was already split into train, test, and validation, we used those splits directly in both the scenarios(the basic model and the efficient model).

4 Results

RESULTS OBTAINED:

With training and testing accuracies exceeding 93%, the machine learning model in our study performed well in classifying X-ray pictures. The model demonstrated strong recall and precision values, indicating its dependability in this regard. Below are all the metrics we have for the model after optimized:

- Train Loss: 0.1534
- Train Accuracy: 94.18%
- Validation Loss: 0.1792
- Validation Accuracy: 93.75%
- Test Loss: 0.1812
- Test Accuracy: 93.58%
- Precision: COVID (0.97), Lung Opacity (0.91), Normal (0.94), Viral Pneumonia (0.98)
- Recall: COVID (0.93), Lung Opacity (0.91), Normal (0.95), Viral Pneumonia (0.95)
- F1-Score: COVID (0.95), Lung Opacity (0.91), Normal (0.94), Viral Pneumonia (0.96)

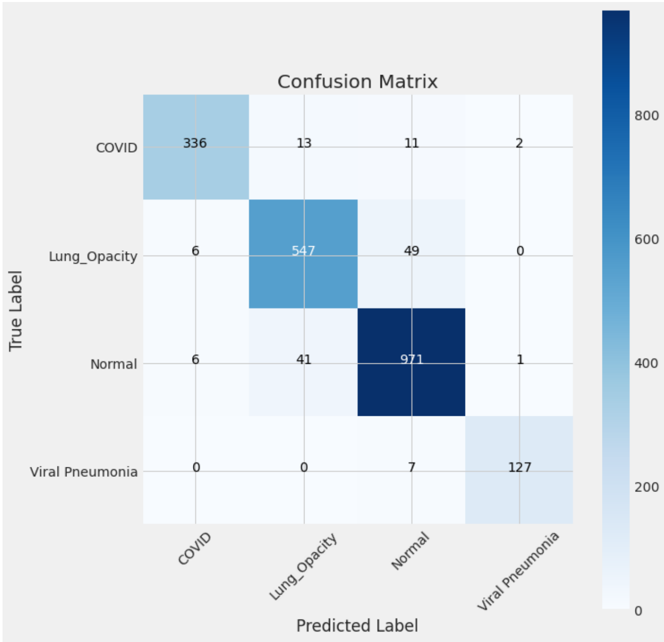

Figure 1: Confusion Matrix after optimization

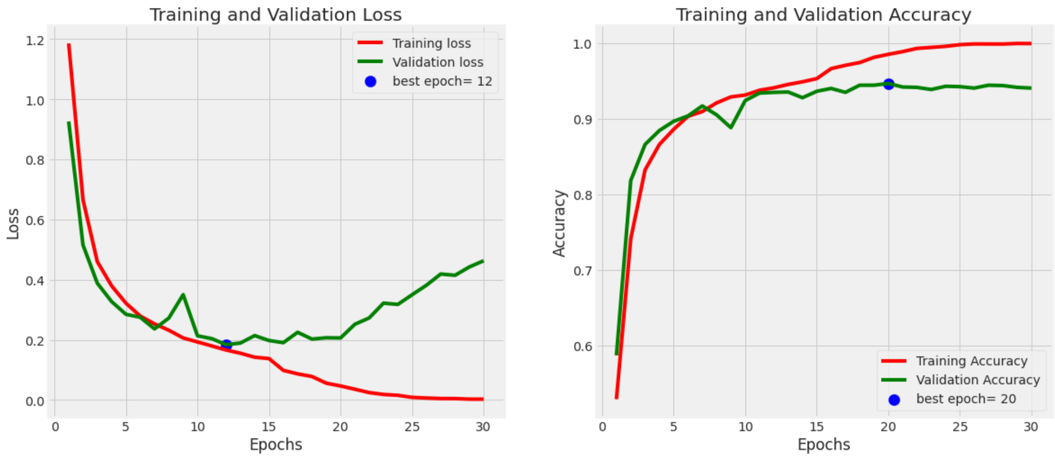

Figure 2: Training, Validation Loss and Accuracy Graph

COMPARISON TO BASELINES:

Our model's capacity to categorize X-ray pictures was greatly enhanced by optimization; test, validation, and training accuracies all dramatically increased from 45–51% to over 93%. Following optimization, the COVID, lung pneumonia, and viral pneumonia and recall scores were remarkably high, highlighting the improved predictive quality of the model. The performance metrics of the model before and after optimization are compared as follows:

Metric	Before Optimization	After Optimization
Train Loss	1.2115	0.1534
Train Accuracy	45.04%	94.18%
Validation Loss	1.1766	0.1792
Validation Accuracy	51.08%	93.75%
Test Loss	1.1866	0.1812
Test Accuracy	48.13%	93.58%
Precision (COVID)	-	0.97
Recall (COVID)	-	0.93
F1-Score (COVID)	-	0.95
Precision (Lung Opacity)	-	0.91
Recall (Lung Opacity)	-	0.91
F1-Score (Lung Opacity)	-	0.91
Precision (Normal)	0.48	0.94
Recall (Normal)	1.00	0.95
F1-Score (Normal)	0.65	0.94
Precision (Viral Pneumonia)	-	0.98
Recall (Viral Pneumonia)	-	0.95
F1-Score (Viral Pneumonia)	-	0.96

Table 1: Model performance comparison between baseline and developed model

5 Conclusions

The advanced model outperforms the basic VGG19 model, owing to a combination of factors such as tailored data preprocessing techniques encompassing image resizing and normalization, alongside the integration of dynamic learning rate adjustment. This dynamic adaptation of learning rates empowers the model to efficiently traverse the training landscape, leading to faster convergence, enhanced generalization capabilities, and increased adaptability to evolving data distributions. Moreover, the comprehensive evaluation, which includes an array of metrics as like accuracy, loss, precision, recall, F1-score, and area under the ROC curve, provides a complete knowledge of the model's performance.

Looking ahead, future endeavors may explore the integration of more advanced methodologies like attention mechanisms or capsule networks, specifically tailored to the intricacies of medical image analysis tasks. By delving into these cutting-edge techniques, we aim to further augment the model's proficiency in handling complex medical imaging data, thus advancing its applicability and efficacy in real-world scenarios.

References

[1] SATISH HIROL. Mastering optimization: Dynamic learning rates unveiled. https://statusneo.com/mastering-optimization-dynamic-learning-rates-unveiled/#:~:text=Dynamic%20learning%20rate%20methods%20adaptively,converge%20to%20a%20good%20solution.

[2] Srikanth Tammina. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, 9(10):143–150, 2019.

[3] Keras Team. Keras vgg documentation. https://keras.io/api/applications/vgg/, 2023. Accessed: 26 April 2024.

[4] Xiao-Hu Yu, Guo-An Chen, and Shi-Xin Cheng. Dynamic learning rate optimization of the backpropagation algorithm. *IEEE Transactions on Neural Networks*, 6(3):669–677, 1995.