X-Ray Segmentation and Classification

Daniyal Ahmed, Vikas Trivedi, Kanishka Dhir
daniyalahmed@hotmail.ca, vtrivedi9770@gmail.com, kdhir3@my.centennialcollege.ca
Student #s: 301152472, 301217554, 301220757

***Abstract*—To address the critical need for novel diagnostic tools during the COVID-19 epidemic, this project uses Deep Learning to create and test three distinct models for COVID-19 detection from chest X-ray images. The dataset includes instances ranging from viral pneumonia to non-COVID lung infections, COVID-19 positive patients, and normal cases. This report digs further into the approaches used, stressing the complexities of model structures and algorithms, as well as a thorough examination of the results.**

I. Introduction

Innovative methods of diagnosis have been spurred by the fight against the COVID-19 pandemic, with artificial intelligence emerging as a powerful tool to combat this fight. The objective of this project was to create and assess three different models for COVID-19 detection from chest X-ray images. The dataset, which was assembled via cooperative efforts, includes a wide variety of cases, such as viral pneumonia, non-COVID lung infection cases, COVID-19 positive cases, and normal cases [1]. This report explores the methods used, the complexities of the model architectures and algorithms, and a thorough examination of the outcomes. The images with their masks can be seen in Fig 1.

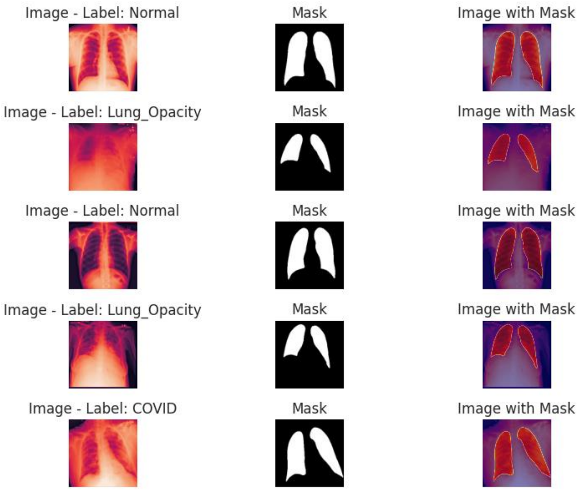

Fig. 1. True images with masks

II. Methodology

A. *Model 1: Simple CNN Classification*

For image classification, the first model used a Simple Convolutional Neural Network (CNN), a fundamental tool in computer vision [3]. This model's architecture included multiple convolutional layers for feature extraction, max-pooling layers for spatial down sampling, and dropout layers for regularization. The Rectified Linear Unit (ReLU) activation function was used to introduce non-linearity. For multi-class classification, the final layer included a softmax activation. The categorical cross-entropy loss function and the Adam optimizer were used to train the model. It is critical to investigate the specifics of each convolutional layer to improve interpretability. The first layers concentrated on detecting basic features such as edges and textures, and subsequent layers abstracted these features into more complex patterns. The max-pooling layers were critical in reducing spatial dimensions, allowing the model to focus on the most important features. Dropout layers served as a regularizing mechanism, preventing overfitting by deactivating a subset of neurons at random during training.

B. *Model 2: U^2-Net for Segementation and Classification*

The U2-Net architecture, shown in Fig. 2, created especially for biomedical image segmentation applications, was adopted by the second model. U2-Net is a formidable deep neural network with a specific emphasis on its use in the identification of COVID-19 from chest X-ray images [4]. The novel Recursive Supervision Units (RSU), shown in Fig. 3 are at the heart of U2-Net. These units work in a recursive fashion, extracting features in a hierarchical fashion and gradually revealing intricate details within the input images. The model's recursive nature allows it to capture nuanced patterns, which is critical for detecting COVID-19 anomalies. The addition of skip connections strengthens U2-Net's design even more. These connections allow for the smooth transmission of information across layers, maintaining critical features from the early phases of feature extraction. Specifically, during training, the model uses a combination of binary cross-entropy loss for segmentation and categorical cross-entropy loss for classification, ensuring a comprehensive approach to both tasks. The images segmented by the U2-Net are then flattened and given as an input to a simple CNN to get the accuracy of predictions of the system on the Labeled images.

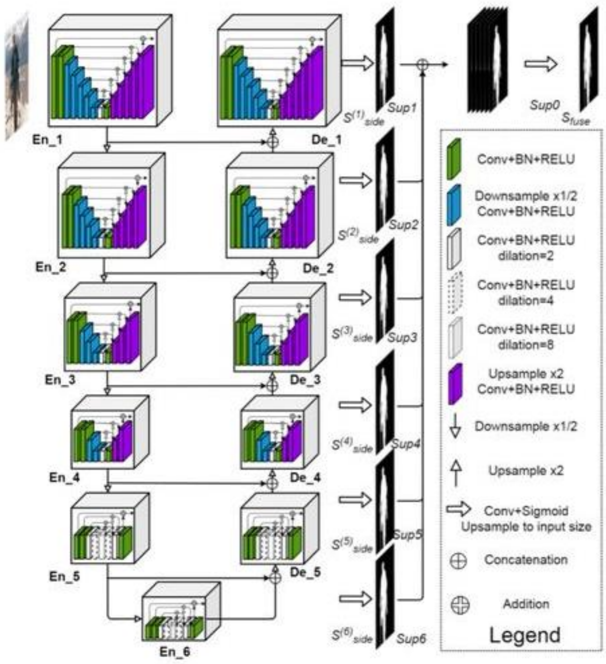

Fig. 2. U^2-Net Architecture

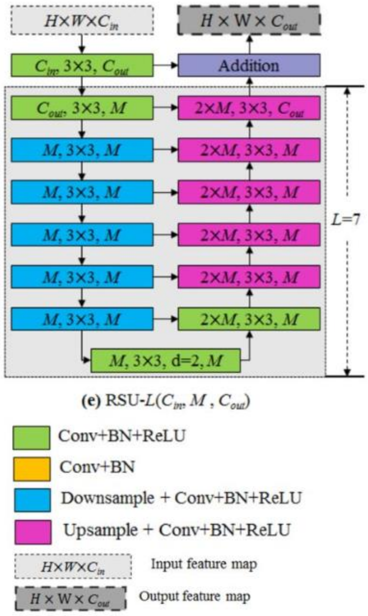

Fig. 3. U²-Net's RSU-L Architecture

C. Model 3: EfficientNetB0 for Classification

EfficientNetB0, a pillar of the EfficientNet family, has proven useful in a variety of computer vision applications, including COVID-19 identification via chest X-ray analysis. The concept of compound scaling, which scales the depth, width, and resolution of the neural network simultaneously, is key to its efficiency [5]. This balanced scaling enables optimal performance across a wide range of applications while managing computational resources wisely, shown in Fig. 5. The architecture makes use of depth wise separable convolution, a breakthrough that breaks down traditional convolutions into depth wise and pointwise convolutions. This reduction in parameters and processing cost is especially beneficial in cases where resources are limited. Furthermore, EfficientNetB0 leverages transfer learning, relying on pre-trained weights from a model that has been exposed to a large dataset. By integrating knowledge gathered from a diverse set of images, this technique improves the model's ability to recognize COVID-19 instances in chest X-ray images, contributing to robust and accurate classifications in the unique diagnostic context. The U2-Net and EfficientNetB0 architectures work in tandem to build a comprehensive method to addressing the issues of COVID-19 diagnosis via AI-driven picture analysis.

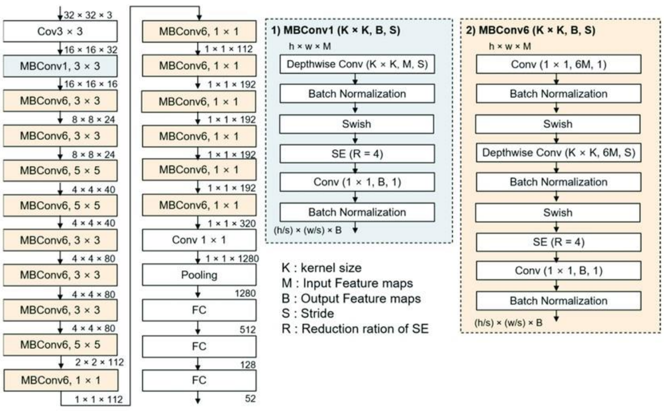

Fig. 4. EfficientNet B0 Architecture

D. Data Pre-processing

Prior to starting the model training phases, the dataset was carefully preprocessed. Resizing the images was essential to maintaining the dataset's uniform dimensionality. To bring photos to a uniform scale, normalization—which involves standardizing pixel values—was used. Furthermore, the dataset was artificially expanded using data augmentation techniques, such as rotation and horizontal flipping, which improved the models' generalization skills.

III. Results

A. Model 1: Simple CNN Classification

The simplicity of the CNN classification model displayed in its sub-par results, achieving a 100% accuracy on both training and validation sets. The low loss values indicated that the model learned some patterns within the dataset. The performance of the simple CNN classification model calls for a closer examination of its generalization capabilities. While the model excelled on both the training and validation sets, it is essential to judge its performance on external datasets to affirm its robustness in real-world scenarios. Due to the 100% accuracy after the first epoch, the conclusion can be made that it is overfitting the data heavily and has not actually learned any of the underlying patterns that should be learned from the dataset. For this reason, this simple CNN although looks like it is performing well, it is overfitting the data thus requiring further exploration using the following models.

B. Model 2: U²-Net for Segmentation and Classification

The U2-Net model, with its dual focus on segmentation and classification, performed well, with an accuracy of 71.88%. The segmentation job allowed the model to pinpoint specific regions of interest, which helped it generate more sophisticated classifications, shown in Fig. 5. However, the results suggested that there was room for improvement, recommending a closer look at optimization tactics to fully realize its potential. The segmentation feature improves the model's interpretability by emphasizing certain regions that contribute to the classification decision. However, the moderate accuracy indicates possible difficulties. Investigating misclassifications and improving the segmentation approach may improve the model's accuracy, making it a more reliable diagnostic tool.

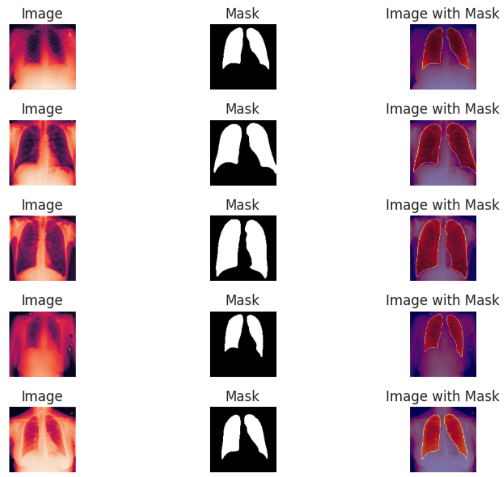

Fig. 5. U²-Net's segmentation/mask results

C. Model 3: EfficientNetB0 for Classification

The EfficientNetB0-based classification model performed admirably, with an accuracy of 97.41% on the training set, 96.77% on the validation set and 96.17% on the testing data. The confusion matrix demonstrated its capacity to distinguish between classes with good precision and recall values. The exceptional classification performance of the EfficientNetB0-based model highlights the importance of efficient architectures in medical image processing. Despite the early stopping, the model displayed an impressive capacity to generalize to new data. Exploring the model's decision-making process using interpretability techniques could reveal insights into the aspects that drive its correct predictions, increasing faith in its diagnostic skills.

IV. Conclusion

Finally, the three models demonstrated distinct techniques of detecting COVID-19 from chest X-ray pictures. The CNN classification model's simplicity, the U2-Net model's dual-task strategy, and the efficiency of the EfficientNetB0-based model all demonstrated the versatility of deep learning in medical imaging. While each model shows strengths, continuous study and iterative modification are required for these models to be translated into useful, trustworthy tools for healthcare providers. The path from algorithmic invention to clinical applicability is complicated, but our findings point to a possible path forward in the aim of utilizing artificial intelligence for better medical diagnoses.

V. Contributions

This project exemplifies teamwork, with each group member bringing unique skills to its accomplishment. Vikas Trivedi fine-tuned and investigated U2-Net Architecture, Kanishka Dhir maintained the EfficientNetB0 model, and Daniyal Ahmed developed the documentation (report, PowerPoint, and ReadMe). The collaboration of varied skills highlights the interdisciplinary nature of using artificial deep neural networks for medical diagnostics.

References

[1] T. Rahman, “Covid-19 radiography database,” Kaggle, https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database (accessed Dec. 2, 2023).

[2] J. Brownlee, “How to save and Load Your Keras Deep Learning Model,” MachineLearningMastery.com, https://machinelearningmastery.com/save-load-keras-deep-learning-models/ (accessed Dec. 4, 2023).

[3] “What are convolutional neural networks?” IBM, https://www.ibm.com/topics/convolutional-neural-networks (accessed Dec. 6, 2023).

[4] “Papers with code - U2-net explained,” Explained | Papers With Code, https://paperswithcode.com/method/u2-net (accessed Dec. 2, 2023).

[5] K. Team, “Keras Documentation: EfficientNet B0 to B7,” Keras, https://keras.io/api/applications/efficientnet/ (accessed Dec. 4, 2023).

[6] K. Team, “Keras Documentation: Models API,” Keras, https://keras.io/api/models/ (accessed Dec. 2023).

[7] “Sklearn.model_selection.train_test_split,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (accessed Dec. 2023).

[8] “3.1. cross-validation: Evaluating estimator performance,” scikit, https://scikit-learn.org/stable/modules/cross_validation.html (accessed Dec. 2023).

[9] “Sklearn.preprocessing.LabelEncoder,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#sklearn.preprocessing.LabelEncoder (accessed Dec. 2023).