BDA800: Business Analytics

COVID-19 Chest X-Ray Classification

By:
Efat Gorji,
Patrick Ellison,
Prince Shamalshibhai Dobariya,
Victoria Villani

COVID-19 Chest X-Ray Image Classification

Problem Statement

Canada's healthcare system, especially during the pandemic, has become overwhelmed, resulting in immense strain on health care providers (Government of Canada, 2024). While Canadian healthcare systems tried to address the surge in COVID-19 cases, backlog inevitably occurred, leading to challenges in providing timely care to both COVID-19 and non-COVID-19 patients (Zeitouny, 2023).

Thus, the ability to quickly prioritize patients is critical to ensure immediate isolation of COVID-19 diagnoses, consequently slowing the spread of the virus, reducing diagnosis wait times, alleviating strain on healthcare workers, and providing better patient care for those who need it urgently.

This project aims to accurately classify chest X-ray images into distinct categories of lung infections, including COVID-19. Implementing such a classification model in healthcare settings not only addresses current challenges, but also offers a valuable resource for enhancing preparedness and adaptability in the face of future pandemics, encouraging resilience in the healthcare systems.

Dataset

The COVID-19 Radiography Database available on Kaggle was utilized. It was created by a team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh, along with collaborators from Pakistan and Malaysia, in collaboration with medical doctors. The dataset can be accessed on the COVID-19 Radiography Dataset website.

The dataset comprises chest X-ray images, categorizing them into four classes, COVID-19 positive cases, Normal cases, Viral Pneumonia cases, and Non-COVID lung infection. The dataset consists of over 20,000 chest X-ray scans, with approximately 3,000 COVID-19 cases, 1,000 non-COVID infections, 6,000 non-COVID lung infections, and 10,000 normal cases.

For each X-Ray class, the dataset includes an *image* directory which explicitly contains X-ray images (See Figure 1.). There is also metadata associated for each image including the file name, format, size, and URL (see Figure 2.).

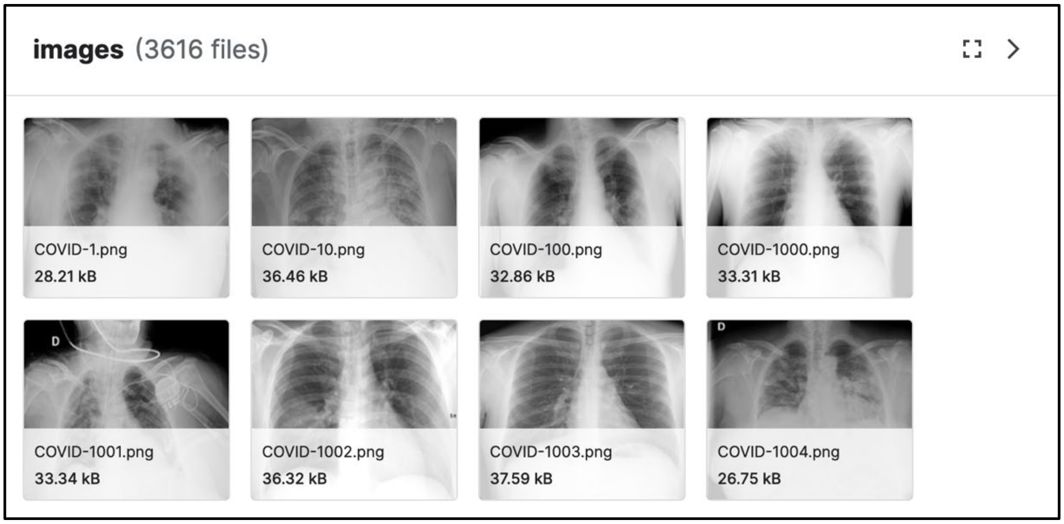

Figure 1. A screen image showing eight X-rays in the COVID-19 images folder.

Sheet1 (3616 rows)

Detail Compact Column

FILE NAME	FORMAT	SIZE	URL
COVID-1	PNG	256*256	https://sirm.org/category/senza-categoria/covid-19/
COVID-2	PNG	256*256	https://sirm.org/category/senza-categoria/covid-19/
COVID-3	PNG	256*256	https://sirm.org/category/senza-categoria/covid-19/

Figure 2. A screen shot showing the metadata provided for the X-ray images in the COVID-19 metadate folder.

Ground Truth

Regarding the specific problem statement, ground truth involves the correct labelling of chest X-ray images (Figure 3.). These initial correct labels provide the foundation for the classification model and serve to evaluate the models' ability to accurately classify unseen X-rays.

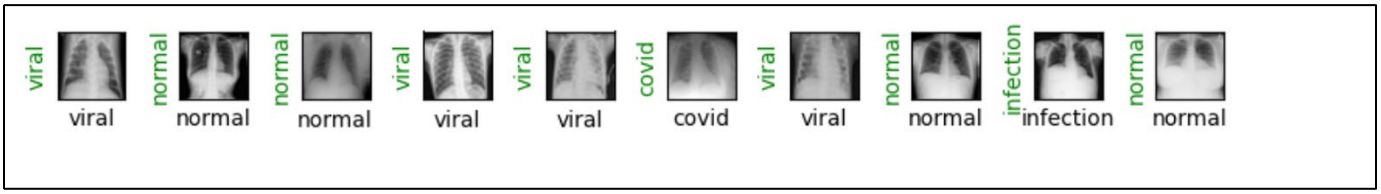

Figure 3. Labelled X-Ray images

Training vs. Testing

To validate our model, a test dataset was created by randomly copying 30 X-Ray images of each class, saving them to a test folder. This folder contained sub-folders for each class, which follows the same structure as the original dataset. This random selection ensured a balanced representation of each class in the test set. Rather than producing a typical 80-20 split, this method helps to avoid bias towards a particular class during the evaluation of the classification model and generalize well when presented with unseen data. The training set size for each class varies; Normal – 10162, COVID-19 – 3586, Infection – 5982, and Viral – 1315.

Methods

Libraries

This project utilizes PyTorch and TorchVision, open-source machine learning libraries. They contain functions that can pull images from directories for classification and recognize patterns in images to perform classification. They also contain functions that can load and preprocess images, such as resizing, cropping, and normalizing, which can help to ensure images are in the correct format and size. PyTorch and TorchVision also contain tools useful for evaluating a model's performance and visualizing the results. The os library was used to help create dictionaries and lists of the X-Ray images, with their respective paths to folders they were located in, helping to identify their label as, normal, Viral Pneumonia cases, Non-COVID lung infections and COVID-19 positive. The random library was used to create a function that can randomly select a class, helping to reduce any bias when selecting images for training our model, which may occur when a model is trained with a class-imbalanced dataset. The Matplotlib library was used for visualizing the X-Ray images.

Data Preprocessing

The X-Ray images are stored in folders that were used identify their label as normal, Viral Pneumonia cases, Non-COVID lung infections or COVID-19 positive.

The pre-trained model ResNet-18 was trained on large datasets containing color images, typically having the three-color channels of red, green, and blue. Since these models are designed to process RGB images, and their architecture and parameters are optimized for this type of input, converting grayscale images to RGB helps ensure our images are compatible with the ResNet-18's input requirements. In our preprocessing class, we added a function that transformed the X-Ray images from black and white to RGB color.

Tensor images are representations of images in the form of multi-dimensional arrays. These are required when implementing classification using ResNet-18 because PyTorch is built around tensor computation, and their neural network modules and layers are designed to operate on tensors. Tensors also allow for efficient computation on both CPUs and GPUs as operations on them that can take advantage of hardware acceleration which can result in faster training.

ResNet-18 was trained with images from ImageNet, which were resized to 224 by 224 pixels. There are also specific mean and standard deviation values that have been adopted as standard normalization parameters when implementing classification using ResNet-18. Therefore, the X-Ray images were resized to match the size of the images ResNet-18 was trained on and normalized the images to the normalized mean and standard deviation values. This also helped to ensure our images are compatible and stabilize the training process by bringing the input data to a similar scale that ResNet-18 was trained. These transformations were coded in an object so they can be applied to both the training and testing datasets consistently.

From PyTorch, Data Loader objects were also utilized to generate custom datasets in batches, allowing us to create the training and testing datasets in batches. A function utilizing Matplotlib was created to visualize the X-Ray images in batches, with the image's ground truth, the class label, displayed on the x-axis, helping us perform exploratory data analysis. This function was also leveraged in visualizing results during model evaluation, allowing us to display predictions on the y-axis, and use colour differentiation to indicate if the prediction was correct or not.

Exploratory Data Analysis

Exploratory data analysis was performed by visualizing the X-Ray images for each class, as well as the examining the frequencies of X-Ray images in each class. These steps helped in understanding the characteristics of the dataset and verifying the correctness of the preprocessing steps.

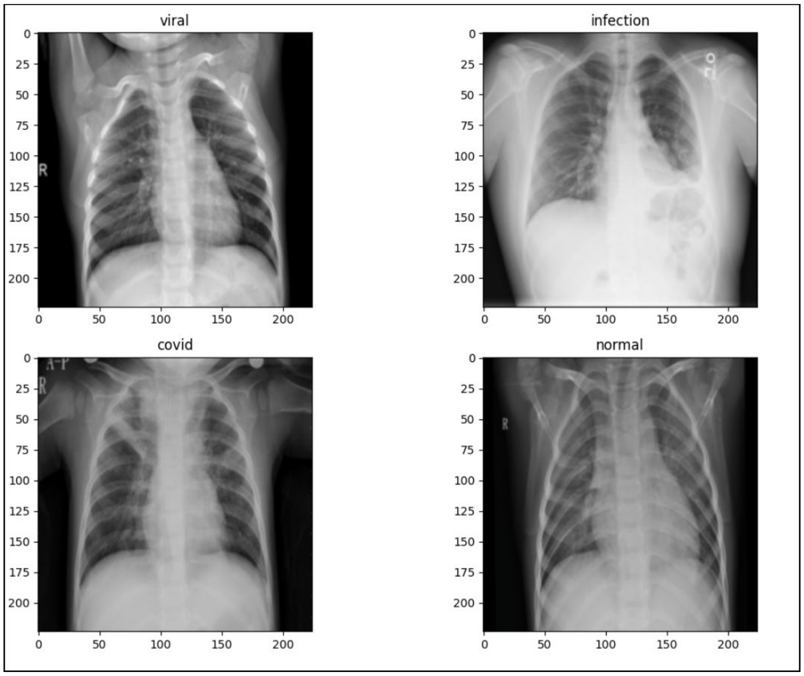

Figure 4. X-Ray images visualized in Python show an example of each class.

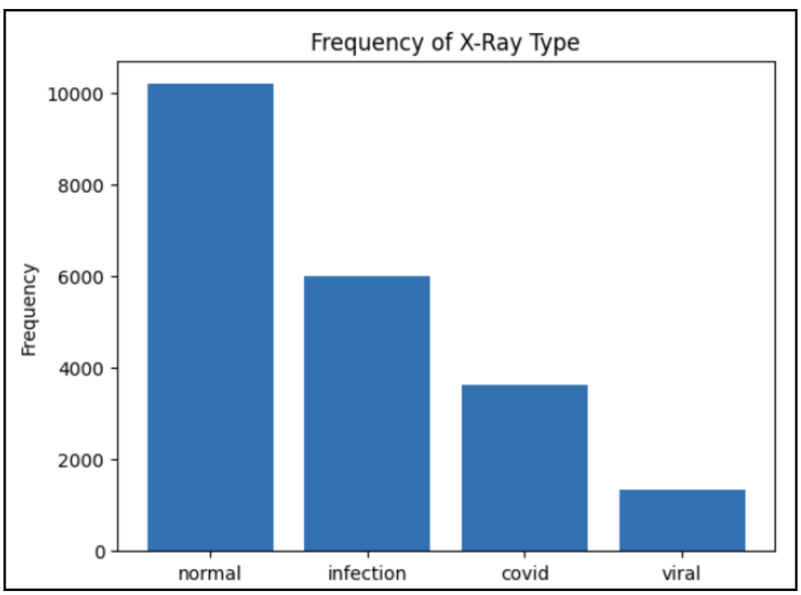

Figure 5. A histogram showing the distribution of the number of X-rays belonging to each class.

While the image in Figure 4, belonging to the infection class was unique, the other images were very similar. This suggests that machine learning techniques can be used to help identify traits not easily identified by the naked eye, consequently helping to diagnose unclassified X-Ray images. The histogram in Figure 5 shows that the number COVID-19 positive and Viral Pneumonia cases are relatively small compared to the other two cases, especially the Normal X-rays. This imbalance of classes in our dataset may create a bias when training our model. To avoid this during training, we created a function in our custom Data Loader class that randomly selects one of the four classes when getting an image.

Data Modeling

A model was created using the pre-trained ResNet-18 model imported from TorchVision. Since ResNet-18 was trained on a dataset that contains 1000 classes, that last layer was modified to have four output features, which corresponds to the four classes of X-ray images. To measure how well the model is performing compared to the ground truth labels, a loss function was defined. Cross-entropy loss was used as the loss function for multi-class classification, and the Adam optimizer was used for model optimization.

The model was trained over multiple epochs using a training loop, allowing the model to be evaluated periodically during training and performance to be monitored so that generalization to unseen data is ensured.

Evaluation

Predictions were visualized by creating a method that utilizes the evaluation mode function in the Resnet 18 model. The method used the test Data Loader to get images and labels from the test dataset, predict its class and display it with the image and actual class.

For each epoch during training, a set of images were passed into the model for training. Each epoch was further divided into 20 steps, where after 20 completed steps, the model was evaluated on the test dataset. At this evaluation step, the trained model was used to make predictions on a set of images from the training set for validation with the ongoing accuracy being calculated. These images were displayed, along with their class, the predicted class in green or red font (green indicating a correct prediction), and the current accuracy. Training was stopped when an accuracy of at least 95% was achieved. Examples of the last three evaluation steps are shown in Figure 6.

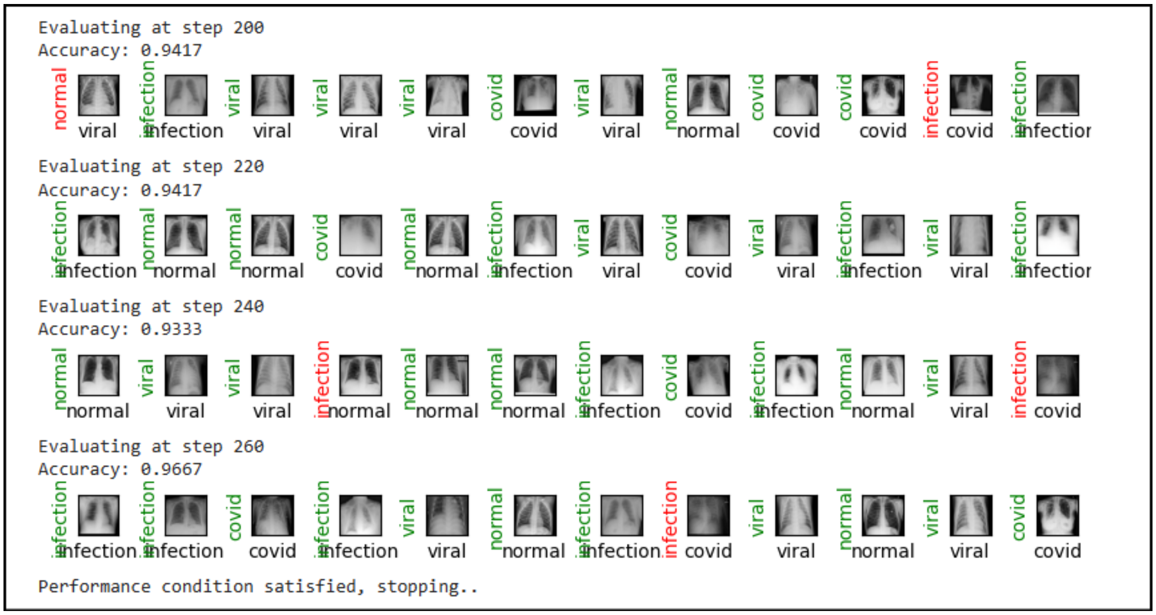

Figure 6. Three steps showing images used in validation, their class, predicted class and the model's accuracy at that set. At step 260, the model made predictions over 95% so training was stopped.

Conclusion

PyTorch was used to create and train a ResNet-18 model using a Chest X-Ray Radiography Dataset. As expected, the more training data used to train the model, the more accurately it became at making predictions on the testing dataset. After 260 steps, the model achieved an accuracy of 95%. These results show that the use of machine learning to classify medical images can help assist medical professionals by enhancing their diagnostic capabilities to detect issues and abnormalities in X-rays, MRIs, and CT scans.

Reflection

Leveraging machine learning techniques to help classify chest X-rays for conditions such as viral pneumonia, lung infections, and Covid-19 can play a significant role in reducing waiting times in hospitals and clinics.

Machine learning algorithms can quickly analyze chest X-ray images to detect patterns associated with different diseases. Instead of waiting for a radiologist to review the images manually, these algorithms can provide results in a matter of minutes. This speed means patients can receive a diagnosis an/or receive treatment sooner, reducing the time they spend waiting for answers and medical care, which can help in emergency situations.

By accurately screening cases of Covid-19, severe lung infections and/or viral pneumonia, healthcare providers can prioritize patients based on conditions that machine learning identifies. Those who are critically ill can be seen and treated promptly, while others with less severe symptoms can wait longer for an official X-Ray reading, helping to ensure that those who need immediate attention get it without delay, and improving outcomes.

Machine learning can help hospitals allocate resources more efficiently by identifying patients who require specialized care or isolation. This information allows healthcare facilities to plan staffing, bed availability, and equipment usage effectively, preventing bottlenecks and reducing waiting times for everyone.

Waiting for test results can be stressful for patients and their families, especially when dealing with a potentially serious illness. Machine learning algorithms can help provide faster and more accurate results, reducing uncertainty and anxiety for patients and enabling healthcare providers to make informed decisions more quickly.

References

Civit-Masot, J., Luna-Perejón, F., Domínguez Morales, M., & Civit, A. (2020). Deep learning system for COVID-19 diagnosis aid using X-ray pulmonary images. Applied Sciences, 10(13), 4640.

Government of Canada. (9, February 2024). *Cutting wait times, delivering better health care in ontario.* https://www.pm.gc.ca/en/news/news-releases/2024/02/09/cutting-wait-times-delivering-better-health-care

Wang, S., Zha, Y., Li, W., Wu, Q., Li, X., Niu, M., Wang, M., Qiu, X., Li, H., Yu, H., Gong, W., Bai, Y., Li, L., Zhu, Y., Wang, L., & Tian, J. (2020). A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. The European respiratory journal, 56(2), 2000775. https://doi.org/10.1183/13993003.00775-2020

Zeitouny, S., Cheung, D. C., Bremner, K. E., Pataky, R. E., Pequeno, P., Matelski, J., ... & Kulkarni, G. S. (2023). The impact of the early COVID-19 pandemic on healthcare system resource use and costs in two provinces in Canada: An interrupted time series analysis. *Plos one*, *18*(9), e0290646.

Declaration

We, Efat Gorji, Patrick Ellison, Prince Shamalshibhai Dobariya, and Victoria Villani declare that the attached milestone is our own working accordance with the Seneca Academic Policy. We have not copied any part of this assignment, manually or electronically, from any other source, including websites, unless specified as references. We have not distributed our work to other students.

All members made significant contributions to writing the Python code and completing each milestone. Below is a chart indicating some specific contributions from each member:

Name	**Tasks**
Efat Gorji	Elevator Pitch (MS2), Project Presentation (MS3), Final Report (MS4)
Patrick Ellison	Python code, Final Report (MS4)
Prince Shamalshibhai Dobariya	Final Report (MS4)
Victoria Villani	Python code, Project Report (MS1), Exploratory Analysis (MS2) and Final Report (MS4)