Data Collection and Preprocessing Phase

Date	29 June 2024
Team ID	SWTID1720084679
Project Title	CovidVision: Advanced COVID-19 Detection from Lung X-rays with Deep Learning
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training.

Section	Description
Data Overview	The dataset contains lung X-ray images categorized into COVID-19 positive, viral pneumonia, and normal cases. The data is sourced from Kaggle under the API "kaggle datasets download -d tawsifurrahman/covid19-radiography-database".
Resizing	Resize images to a standard size (e.g., 299x299 pixels) to ensure uniform input dimensions for the Inception model
Normalization	Normalize pixel values to a range of 0 to 1 to stabilize and accelerate the training process.
Data Augmentation	Apply augmentation techniques such as flipping, rotation, shifting, zooming, and shearing to increase the diversity of the training dataset and reduce overfitting.
Color Space Conversion	Convert images from grayscale to RGB, if necessary, to match the input requirements of the Inception model, which expects 3-channel images.
Image Cropping	Crop images to focus on the lung regions, eliminating irrelevant parts of the X-rays to improve model accuracy.But every image in the dataset is already perfect for use, there is no need of image cropping

Batch Normalization	Apply batch normalization to the input of each layer in the neural network to improve training stability and performance. But, Batch Normalization is not required in Transfer learning based codes.
Data Preprocessing Code Screenshots	

Loading Data	`# Define the path to the unzipped dataset`
	`dataset_path = '/content/COVID-19_Radiography_Dataset/'`
	
	`# Create directories for train and test splits`
	`train_dir = '/content/train'`
	`test_dir = '/content/test'`
	`os.makedirs(train_dir, exist_ok=True)`
	`os.makedirs(test_dir, exist_ok=True)`
	
	`# Create subdirectories for each class in train and test directories`
	`classes = ['COVID', 'NORMAL', 'VIRAL_PNEUMONIA', 'LUNG_OPACITY']`
	`for cls in classes:`
	`os.makedirs(os.path.join(train_dir, cls), exist_ok=True)`
	`os.makedirs(os.path.join(test_dir, cls), exist_ok=True)`
	
	`# Function to get all image file paths from the directory`
	`def get_image_paths(directory):`
	`image_paths = []`
	`for root, _, files in os.walk(directory):`
	`for file in files:`
	`if file.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp', '.tiff')):`
	`image_paths.append(os.path.join(root, file))`
	`return image_paths`

Resizing	`train = train_datagen.flow_from_directory(trainPath,target_size=(224,224),batch_size=16)` `test = test_datagen.flow_from_directory(testPath,target_size=(224,224),batch_size=16)`
Normalization	`train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2)` `test_datagen = ImageDataGenerator(rescale=1./255)`
Data Augmentation	`train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,shear_range=0.2)` `test_datagen = ImageDataGenerator(rescale=1./255)`
Color Space Conversion	There is no need of Color Space Conversion for this dataset, as it is perfetly placed.