MA 305: Data Science Project
Group 4

Members:
Anmol Jain - 230008009
Nidarsana M - 230004031
Salaj Bansal - 230002063

The GitHub link for the dataset is mentioned below in Section 3.

1. Computing Environment

- **Platform used:** Google Colab
- **Runtime:** T4 GPU
- **Python version:** 3.x
- **Mixed Precision:** Enabled (`mixed_float16`) for faster training

2. Required Libraries & Versions

The following libraries are required (latest version on Colab):

- **TensorFlow:** 2.x (with Keras integrated)
- **NumPy:** latest Colab version
- **Pandas:** latest Colab version
- **OpenCV (cv2):** latest Colab version
- **Seaborn:** latest Colab version
- **Matplotlib:** latest Colab version
- **scikit-learn:** latest Colab version
- **Git (for cloning the dataset)**

- **Mixed Precision API:** TensorFlow's `mixed_precision`
- **InceptionV3** (from `tf.keras.applications`)

All these libraries come pre-installed in Colab. No manual installation is required.

3. Dataset Information (Component 3)

We use the **COVID-19 Chest X-ray Dataset** publicly available at:

https://github.com/ieee8023/covid-chestxray-dataset

Contents used:

- Images from: `covid-chestxray-dataset/images/`
- Metadata from: `covid-chestxray-dataset/metadata.csv`

The dataset is downloaded automatically using `git clone` inside the notebook.

4. Data Preparation Summary

- Metadata is filtered to include **only X-ray images** with valid findings and views.
- Labels used:
    - `"Pneumonia/Viral/COVID-19"` → **COVID-19**
    - All other findings → **Non-Covid**
- Images are matched with labels and stored in a DataFrame.
- The dataset is **balanced through aggressive oversampling** so both classes have equal samples.
- Final split:
    - **80% Training**
    - **10% Validation**
    - **10% Testing**

5. Model Description

The model is based on **InceptionV3 (ImageNet weights, without the top layer)** along with:

- Dual feature pooling:
    - Global Average Pooling
    - Global Max Pooling
- Deep fully-connected classification head with:
    - Batch Normalization
    - Dense layers: 1536 → 768 → 384 → 192
    - L1–L2 regularization
    - Dropout layers (0.6, 0.5, 0.4, 0.3)
- Final Softmax output for **2 classes**

The base model weights remain frozen during training.

6. Training Configuration

- **Optimizer:** Nadam
- **Learning Rate:** 0.001
- **Loss:** Categorical Crossentropy
- **Metrics:**
    - Accuracy
    - AUC
    - Precision
    - Recall

- **Batch Size:** 12
- **Image Size:** 299 × 299
- **Callbacks:**
    - Early Stopping
    - ReduceLROnPlateau

7. Data Augmentation Used

During training, the following augmentations are applied:

- Rotation
- Width/height shifting
- Zoom
- Brightness variations
- Shear
- Horizontal flips
- Channel shifts
- Image preprocessing using InceptionV3's preprocessing function

Validation and test sets only use preprocessing, without augmentations.

8. Test-Time Augmentation (TTA)

To improve prediction robustness, inference is performed using several augmented versions of each test image, and predictions are averaged.

9. Steps to Reproduce the Results

1. Open the submitted **Google Colab notebook**.
2. Set the runtime to **GPU** (`Runtime → Change runtime type → GPU`).
3. Run all cells **in order**, without skipping any.
4. The notebook will automatically:
    - Clone the dataset
    - Load and preprocess images
    - Balance the dataset
    - Create train/validation/test sets
    - Build the InceptionV3 model
    - Train the model with the exact hyperparameters
    - Evaluate on the test set
    - Generate plots and metrics
5. The notebook also includes code for Test-Time Augmentation (TTA) for prediction enhancement.

10. Randomness and Reproducibility Notes

- All sampling steps use a fixed random seed (**42**).
- GPU operations may cause slight nondeterministic variations in floating-point computations.
- Aside from minor numerical differences, running the notebook again should reproduce the same performance trends.