# Adversarial Attacks And Interpretability Covid Chestxray Dataset

Cognitive Computing and Artificial Intelligence - UniCT a.a 2019/2020

Raiti Mario O55000434
Nardo Gabriele Salvatore O55000430
Sortino Renato O55000405

# Introduction

- The project is available on github at following link : https://github.com/RaiMar96/AdversarialAttacksAndInterpretability-covid-chestxray-
- The Dataset is available on github at following link : https://github.com/ieee8023/covid-chestxray-dataset
- The project has been implemented in python on Jupyter environment, using Google Colab platform.

# Informazioni Preliminari

- In the following experiments a subset of the provided dataset has been utilized (images from perspective ‘PA’), made of 303 samples.
- The following ratios have been applied: Train set 70% , Validation set 15%, Test set 15%

# Outline

- Dataset Creation
- From Scratch CNN, train and evaluation
- Model interpretabilty using Integrated Gradients
- Adversarial attack FSGM
- Adversarial Training, Fine tuning and evaluation of the model

# Dataset Creation

# Dataset Creation

- Dataset creation has been realized with the image_dataset class, which takes as input parameters the file paths of CSV file containing the structure of the dataset, and the image fodler, the transformations to apply and the relative phase for the subset (Transformations are defined separately for train and test).

# Dataset Creation

- After creating the image dataset, the train, validation and test subset are generated using the class torch.Subset, and the relative dataloaders.

# From Scratch CNN, train and evaluation

# From Scratch CNN, train and evaluation

- The from scratch model is made of 6 convolutional levels, each of which is followed by ReLU, BatchNormalization and MaxPooling.
- Binary Cross Entropy is used as evaluation criterion for loss computation.
- As optimizer, ***Adamo chiddu de scummissi*** is used**.**

# From Scratch CNN, train and evaluation

- For model evaluation two functions, testAccuracy and testCovid, are used.
- *testAccuracy* computes test accuracy for the test subset.
- *testCovid* computes accuracy of Covid/NOCovid classes.

# Model interpretabilty using Integrated Gradients

# Model interpretabilty using Integrated Gradients

- Pytorch Captum module is used for model interpretability.
- modelInterpretation function use integrated gradient criterion to define feature meaning after model computations.
- Attributes are plotted trough '*visualize_image_attr*' called on '*visualization*' element of captum.

# Adversarial attack FGSM

# Adversarial attack FGSM

- The adversarial attack taked into consideration is ***Fast Gradient Sign Method***.
- Fgsm_attack function perform images perturbation that adds noise to the original images, multiplying an epsilon value to the gradient sign of the data. In our experiment, epsilon = 0.025 is used.
- Test function verify model accuracy after image perturbation.

# Adversarial Training, Fine tuning and evaluation of the model

# Adversarial Training, Fine tuning and evaluation of the model

- Same operations performed before, are repeated in adversarial mode.
- A new adversarial image dataset is created; the new loaders are created through concatenation of the original dataset and the adversarial one; to do this, ConcatDataset class is used.
- Model is trained with perturbed images, tuning parameters. Than, evaluation operations are repeated.

# Results

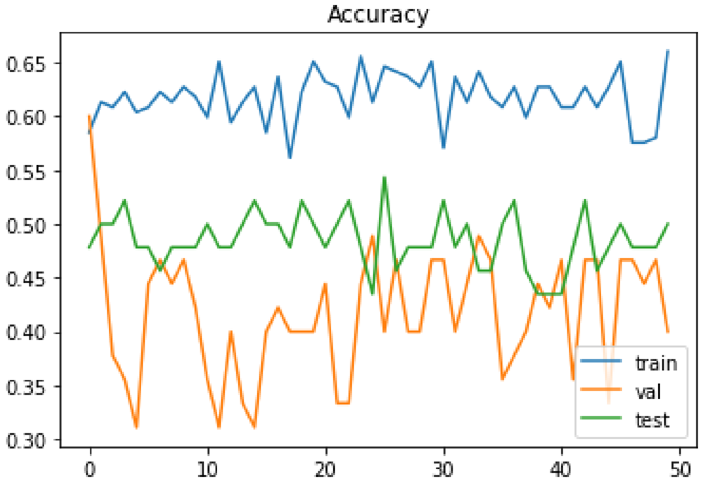
# Setup

For test, the following parameters are used:

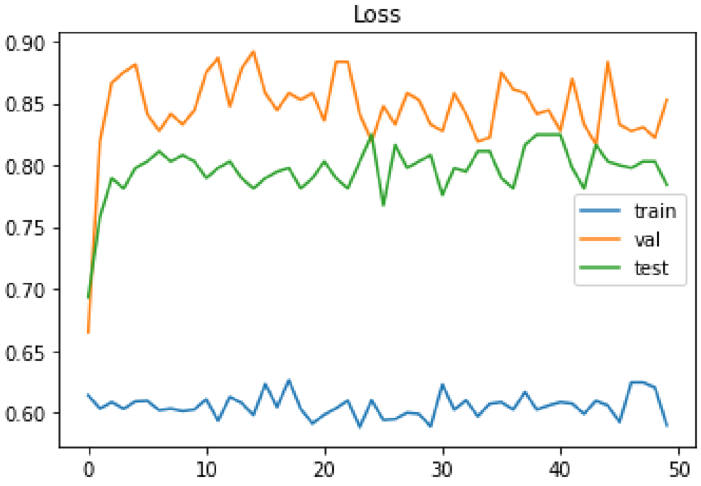
- Batch size = 8;
- Learning rate = 10^-4
- Num of epochs= 50

# From Scratch Model Train Results

## Accuracy

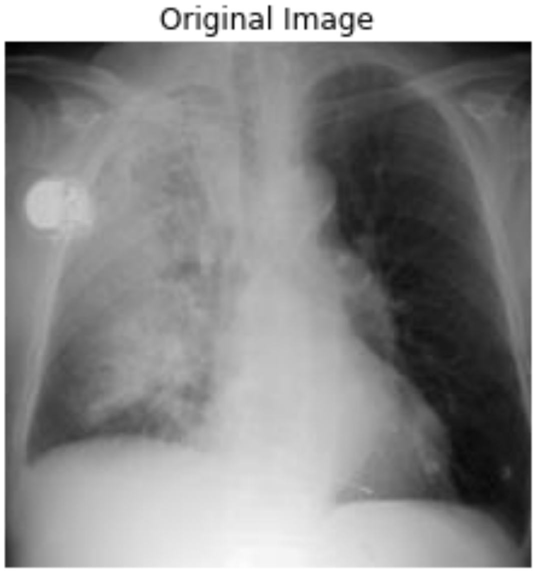


## Loss

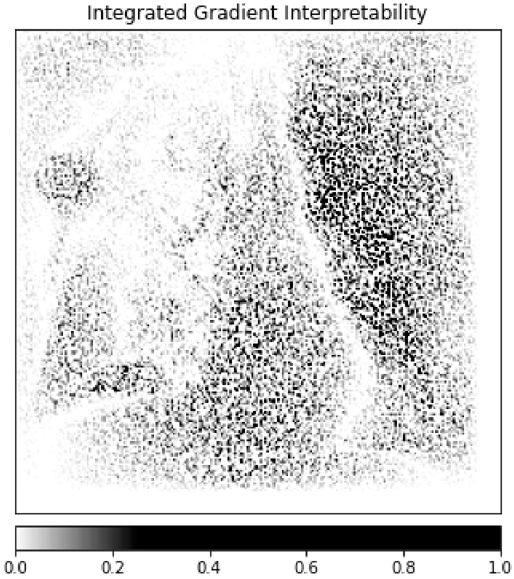


# From Scratch Model Train Results

## Image

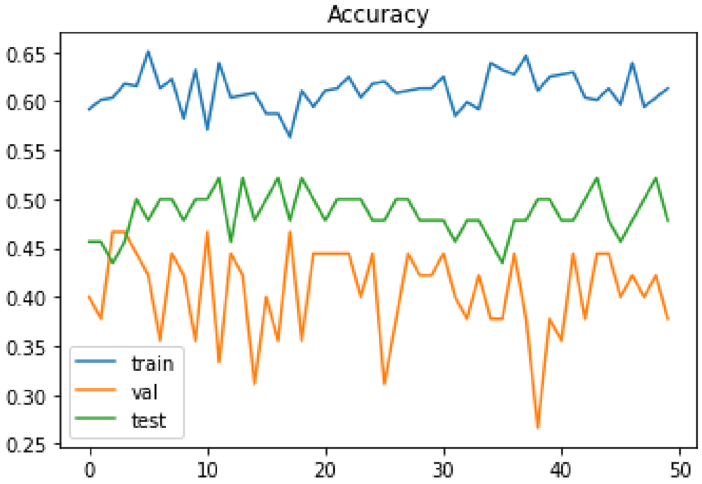


## Interpretation

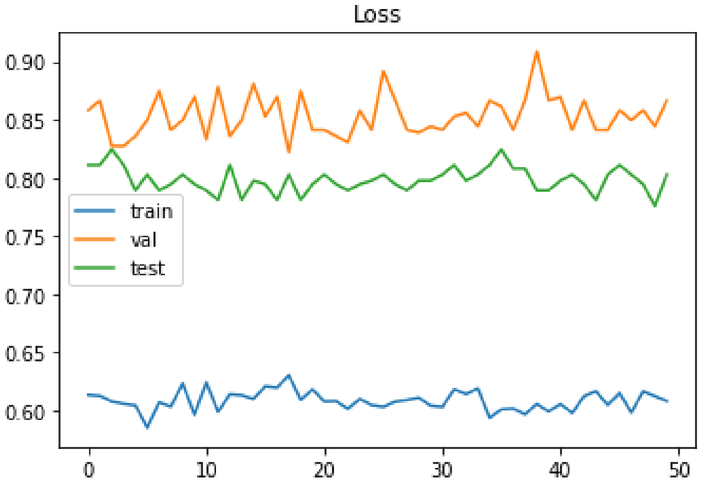


# Adversarial Fine Tuned Model Train Results

## Accuracy

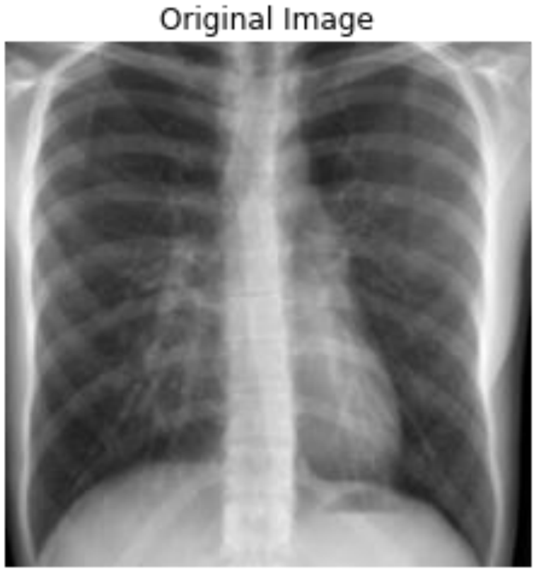


## Loss



# Adversarial Fine Tuned Model Interpretability Results

## Image



## Interpretation