Individual Study III

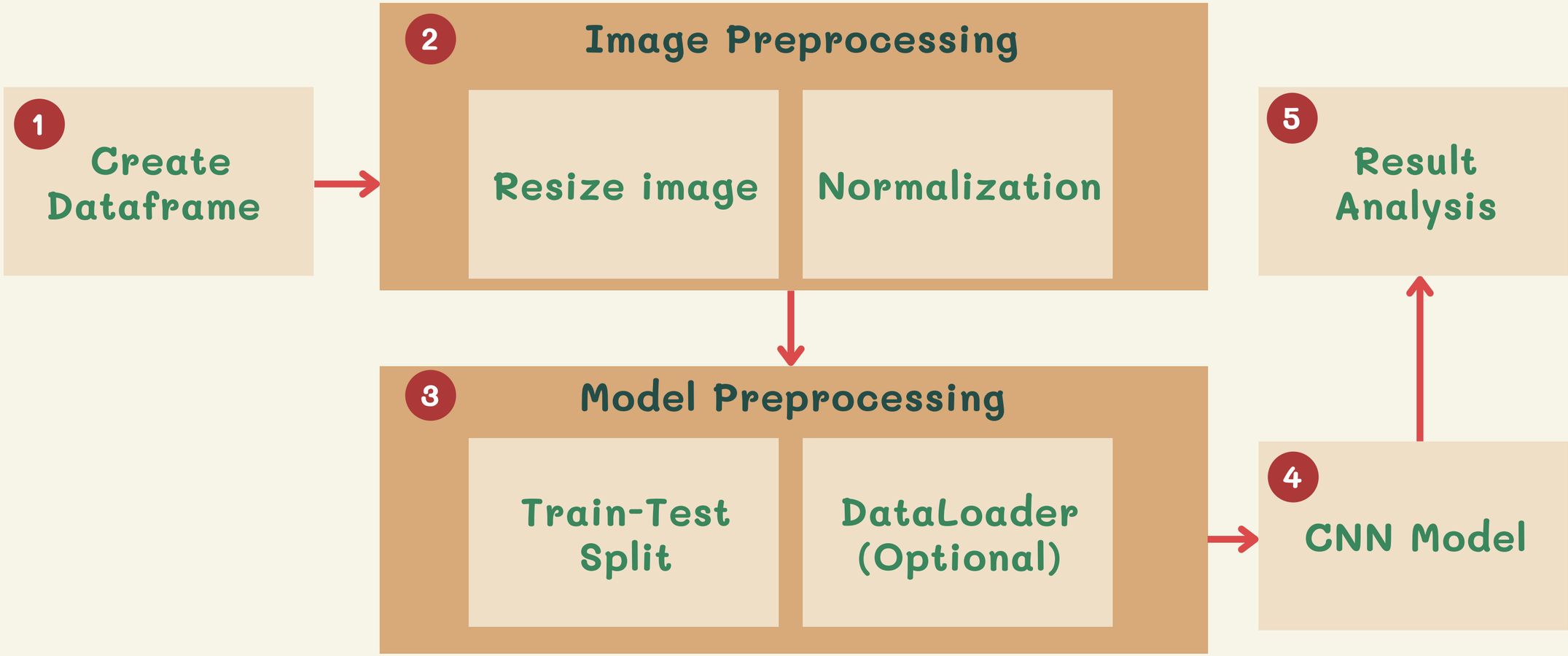
Tuberculosis Chest X-rays

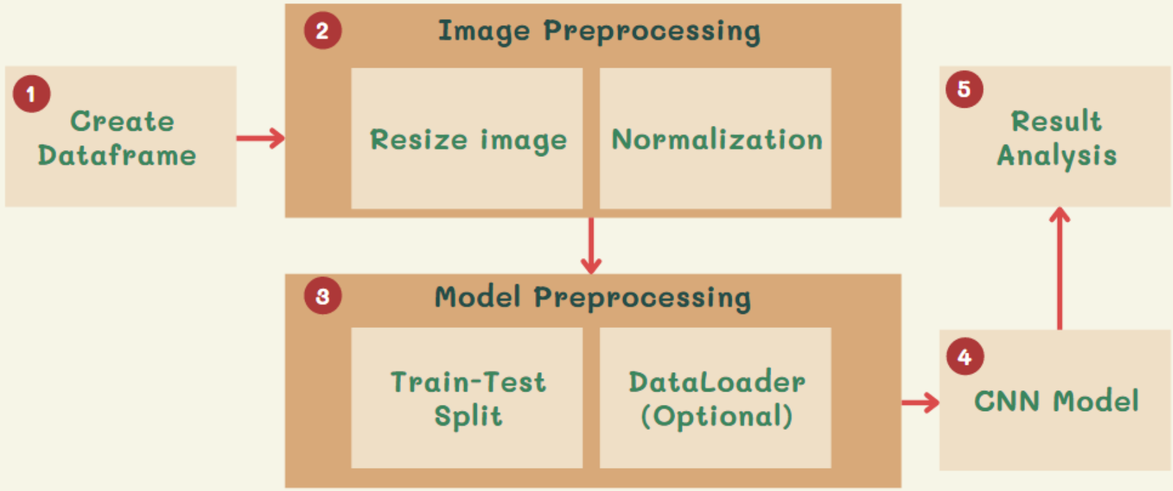
Peeranath Theerawatanachai 6231343821

Objective

1. To predict Tuberculosis from Chest X-ray images
2. To study about Machine Learning (Deep learning from computer vision) which involves
Convolutional Neural Network

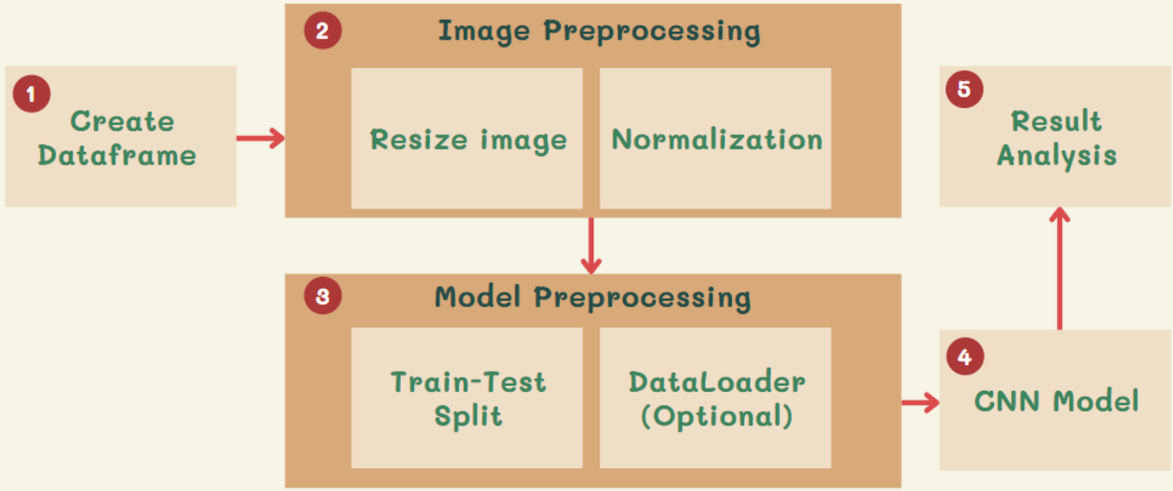
Architectures





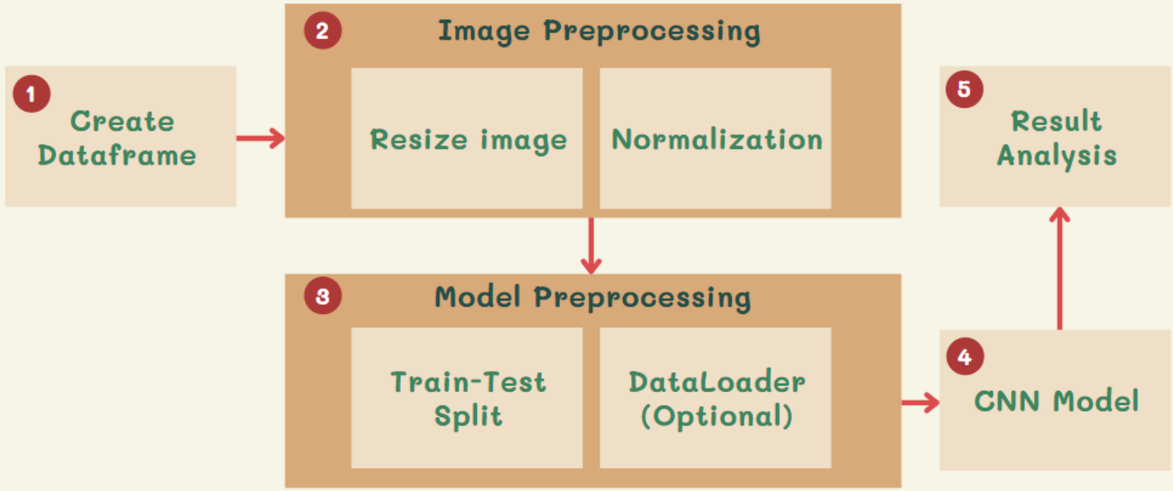
1. Create dataframe

- Download data from Kaggle
- Create dataframe from shenzhen_metadata.csv by adding column label from column finding where 0 means normal and 1 means abnormal (Tuberculosis)



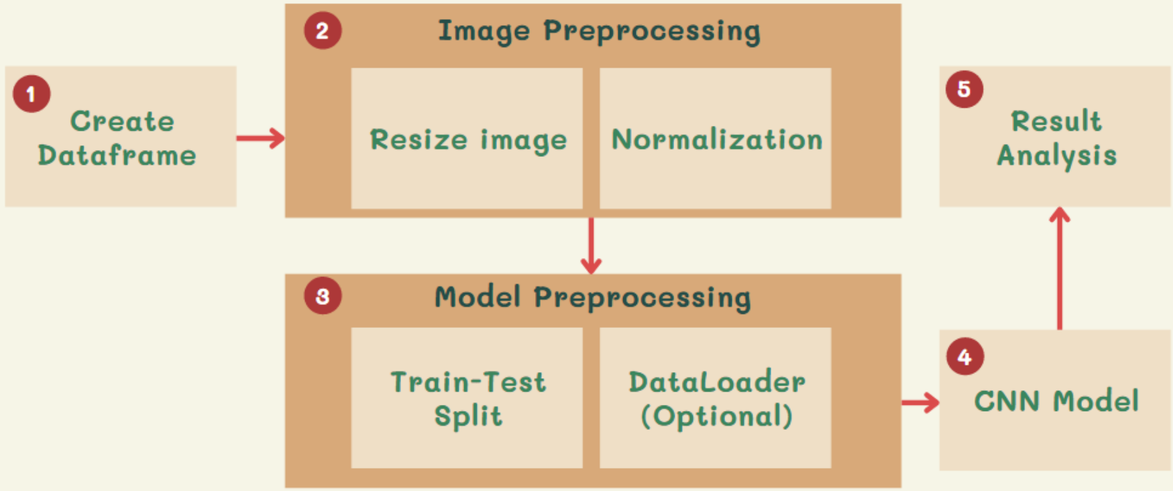
2. Image Preprocessin

- **Resize** Chest X-ray images to 256*256 and 224*224 pixels
- **Normalize** each image by dividing every pixel by 255, then subtract with mean of all images, and divide by standard deviance of all images to make each image has better intensity value distribution



3. Model Preprocessin

- Use **Train-test split** with split size = 15% to divide images for training (562 images) and validating (100 images)
- Create Class XRayDataset to keep image intensity level and labels
- Use **DataLoader** to divide batch for training and validating by using batch size = 16 for training set and batch size = 32 for validating set

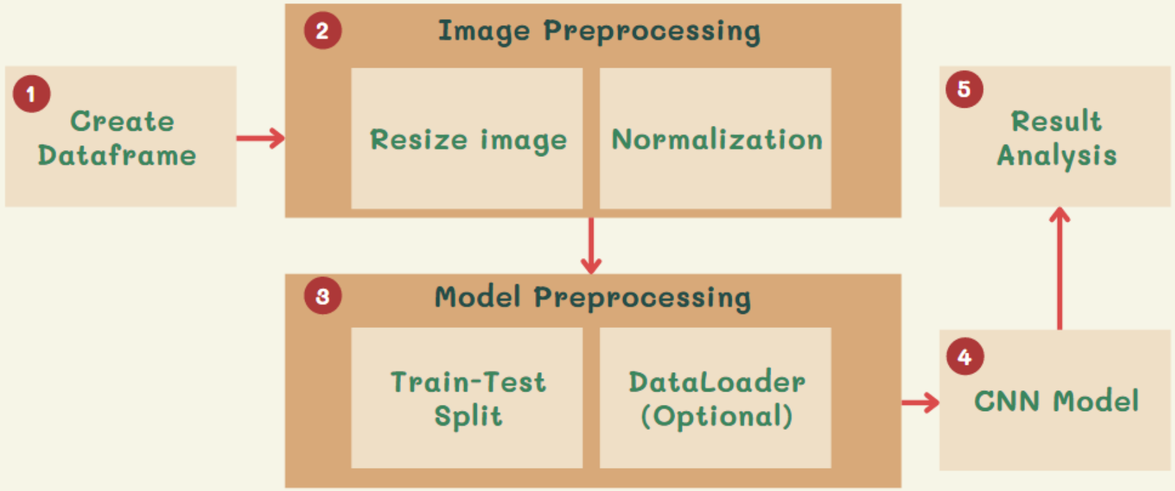


4-5. CNN Model & Result analysis

Model 1

```
Net(
  (cnn_layers): Sequential(
    (0): Conv2d(1, 4, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(4, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(4, 4, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): BatchNorm2d(4, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU(inplace=True)
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (linear_layers): Sequential(
    (0): Linear(in_features=16384, out_features=2, bias=True)
  )
)
```

4-5. CNN Model & Result analysis



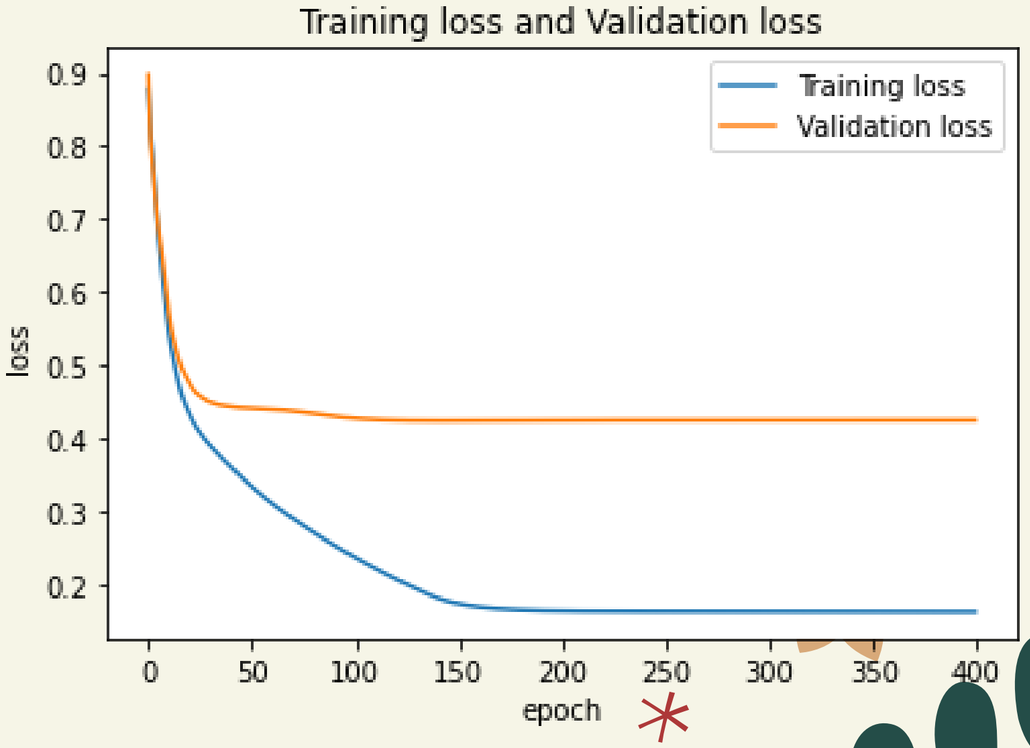
Training 1 (Overfit)

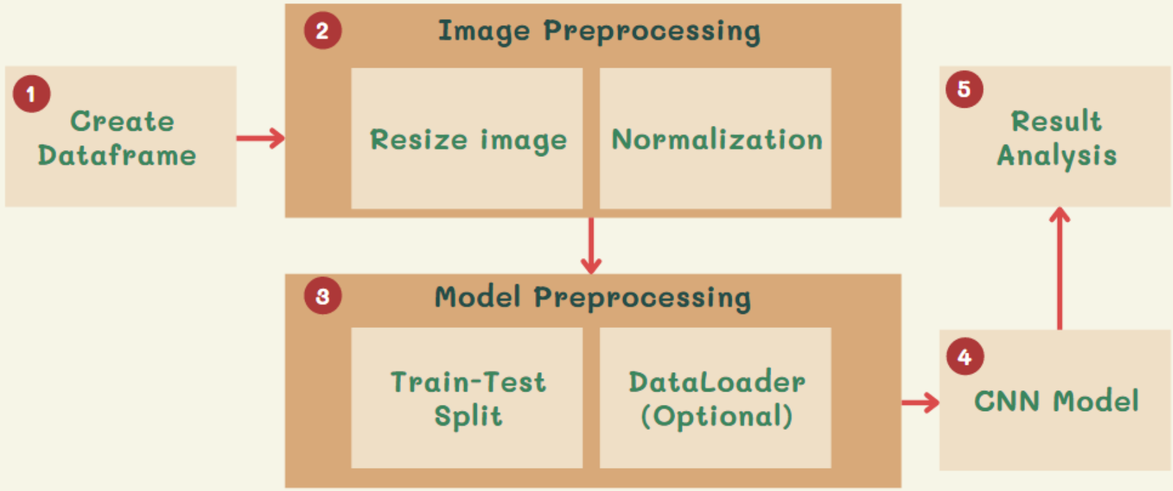
Parameter:

optimizer = Adam
Learning rate = 0.00004
criterion = CrossEntropyLoss
scheduler = ReduceLROnPlateau
number of epochs = 400

Accuracy:

Training accuracy: 0.960854092526690 4
Validate accuracy: 0.84





4-5. CNN Model & Result analysis

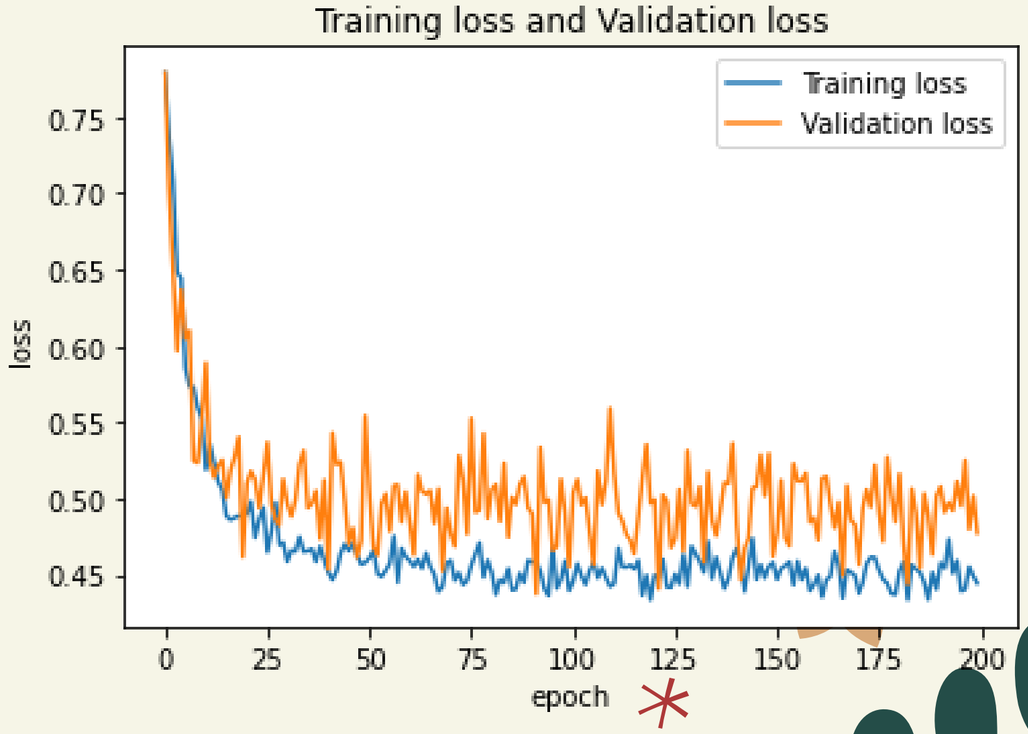
Training 2: Add Dropout = 0.2

Parameter:

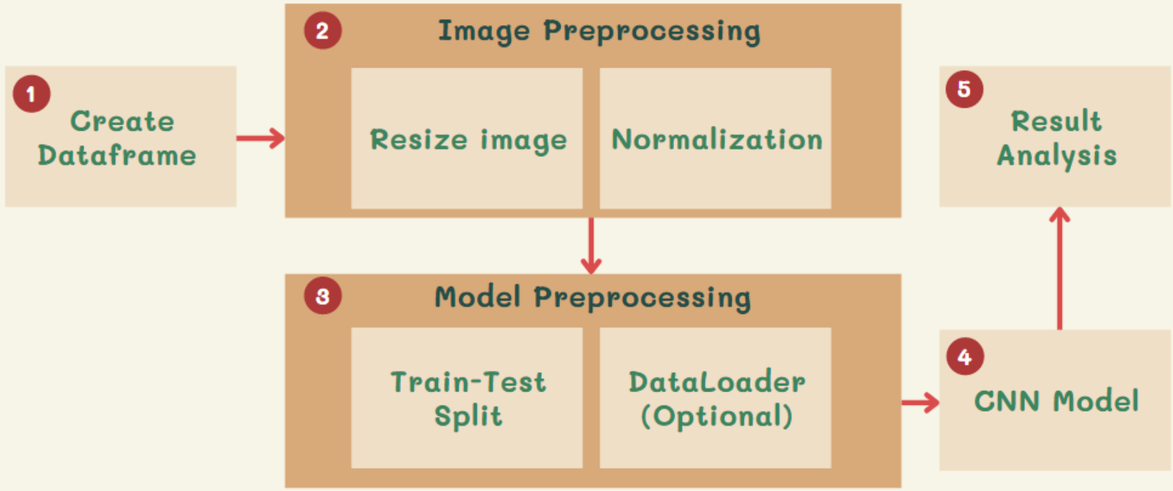
optimizer = Adam
Learning rate = 0.00004
criterion = CrossEntropyLoss
scheduler = ReduceLROnPlateau
number of epochs = 200

Accuracy:

Training accuracy: 0.8096085409252669
Validate accuracy: 0.79



4-5. CNN Model & Result analysis



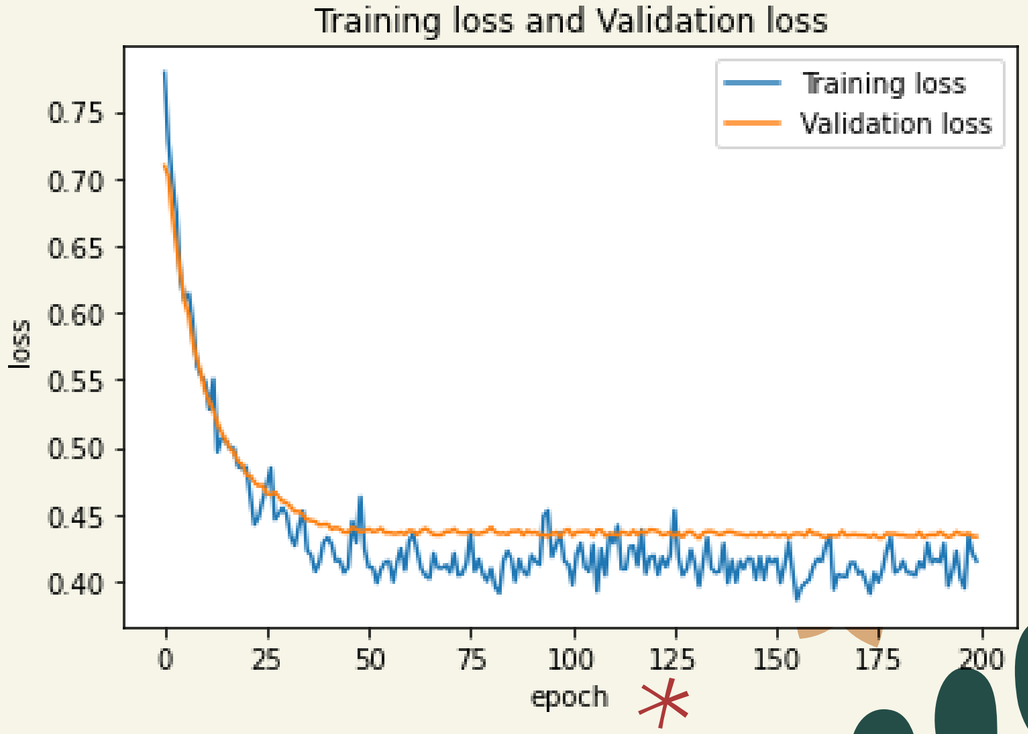
Training 3: Using DataLoader + Dropout = 0.2

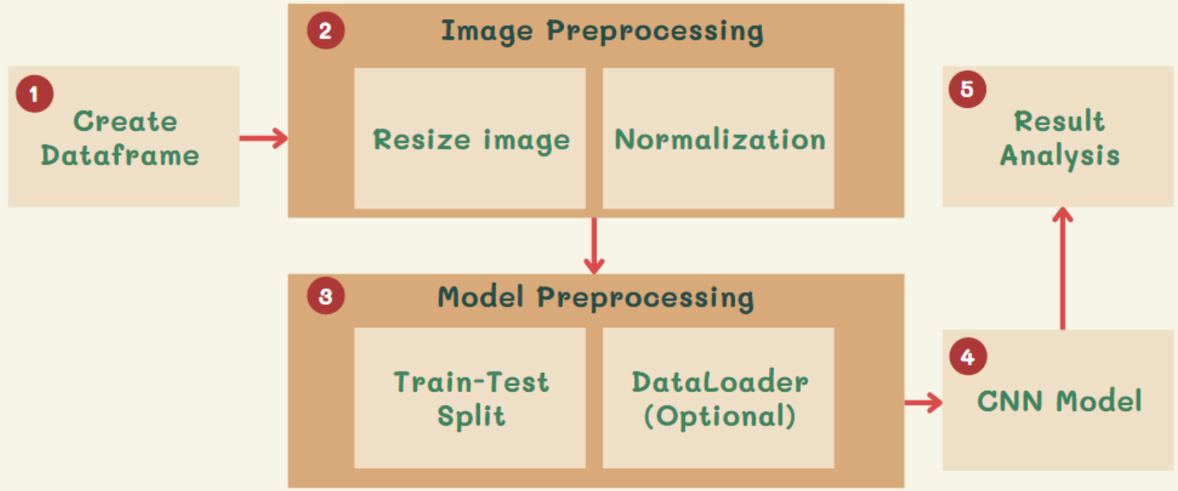
Parameter:

optimizer = Adam
Learning rate = 0.000004
criterion = CrossEntropyLoss
scheduler = ReduceLROnPlateau
number of epochs = 200

Accuracy:

Training accuracy: 0.8454861044883728
Validate accuracy: 0.8671875



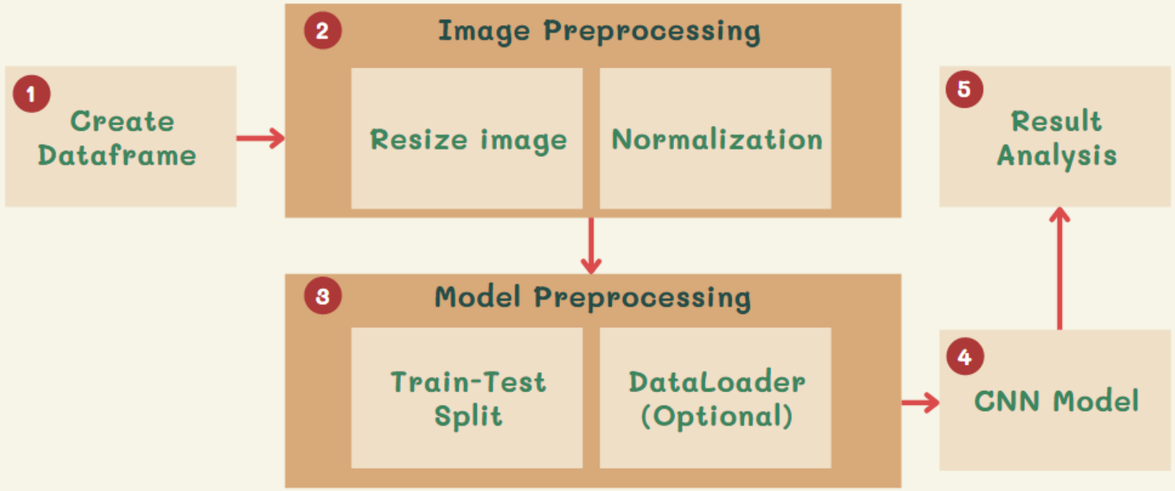


4-5. CNN Model & Result analysis

Model 2 (Using DataLoader)

```
XrayClassification(
  (network): Sequential(
    (0): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU()
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU()
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU()
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU()
    (14): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (15): Flatten(start_dim=1, end_dim=-1)
    (16): Linear(in_features=262144, out_features=1024, bias=True)
    (17): ReLU()
    (18): Linear(in_features=1024, out_features=512, bias=True)
    (19): ReLU()
    (20): Linear(in_features=512, out_features=2, bias=True)
  )
)
```

4-5. CNN Model & Result analysis



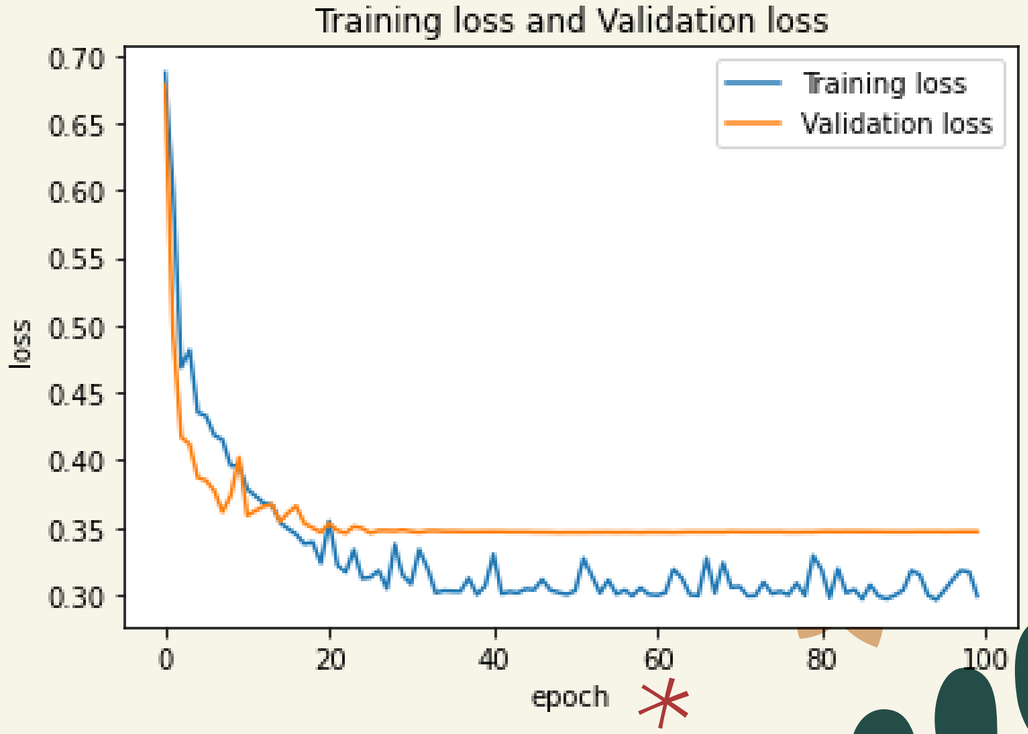
Training 1

Parameter:

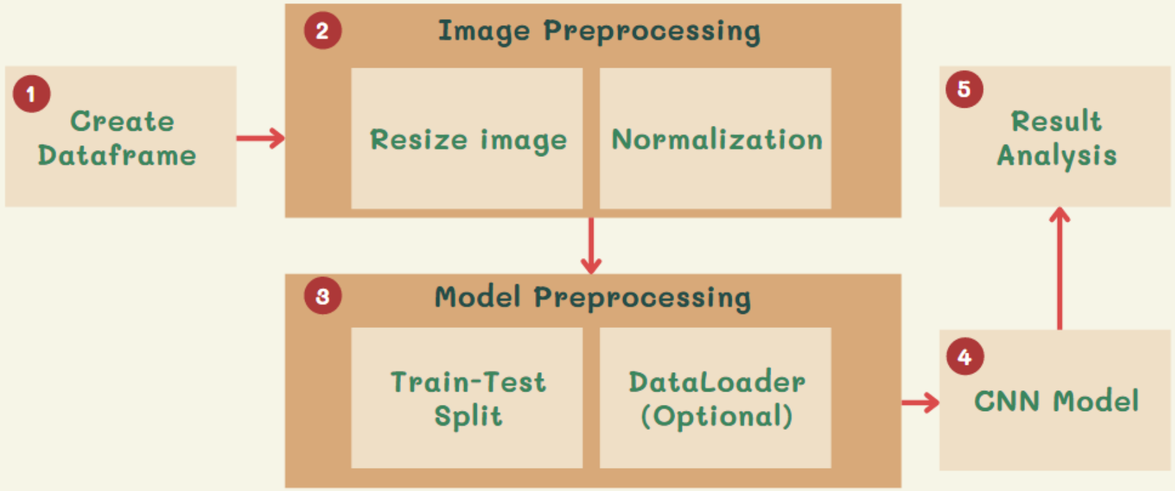
optimizer = Adam
Learning rate = 0.00001
criterion = CrossEntropyLoss
scheduler = ReduceLROnPlateau
number of epochs = 100

Accuracy:

Training accuracy: 0.8784722089767456
Validate accuracy: 0.8515625



4-5. CNN Model & Result analysis



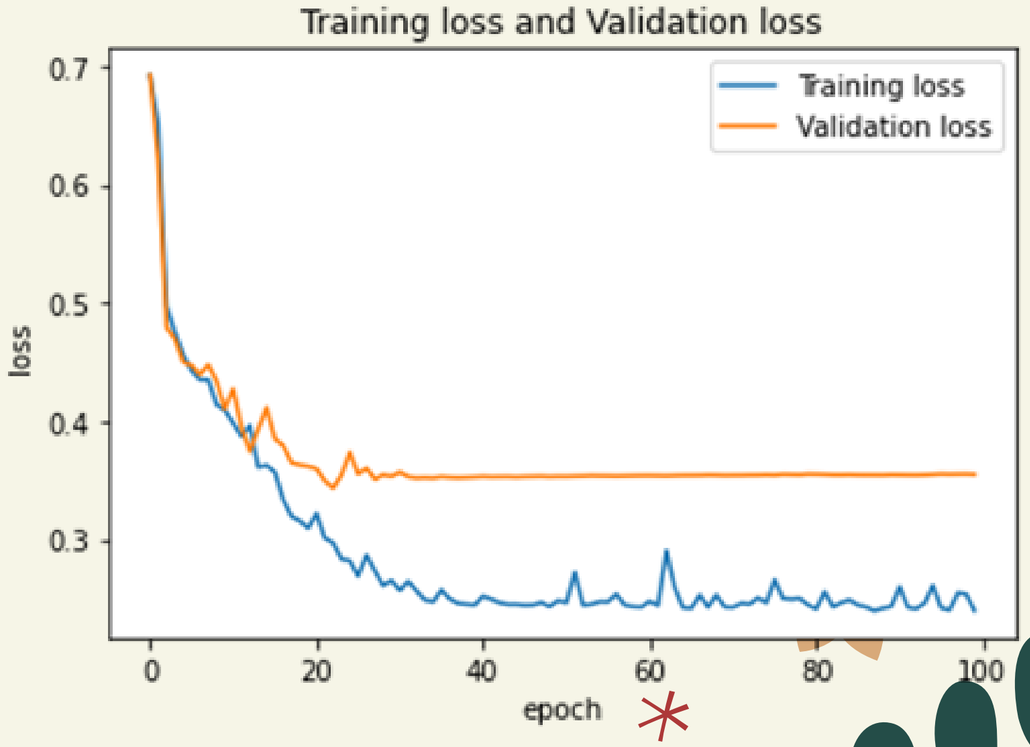
Training 2: Add Dropout = 0.3

Parameter:

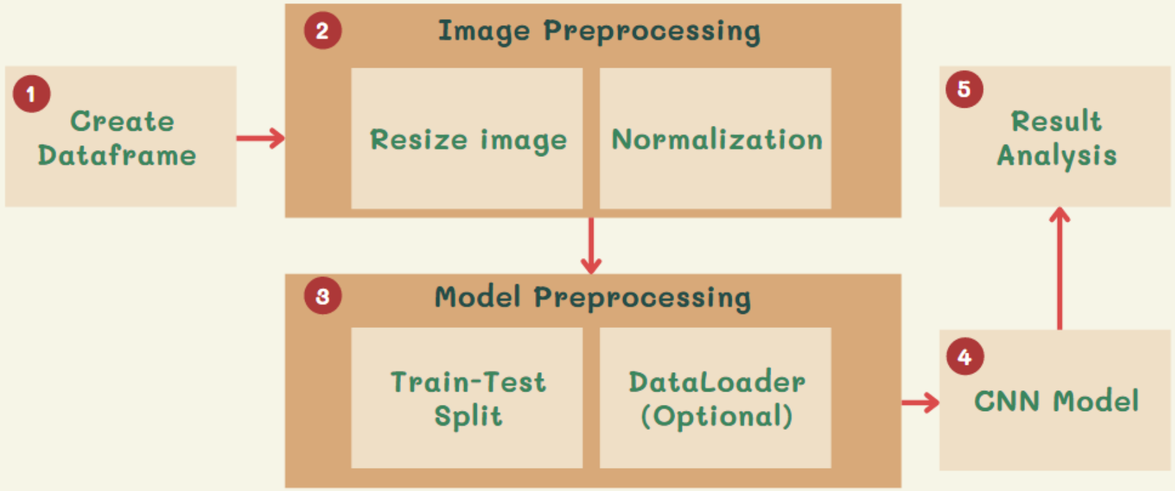
optimizer = Adam
Learning rate = 0.00001
criterion = CrossEntropyLoss
scheduler = ReduceLROnPlateau
number of epochs = 100

Accuracy:

Training accuracy: 0.8888888955116272
Validate accuracy: 0.88



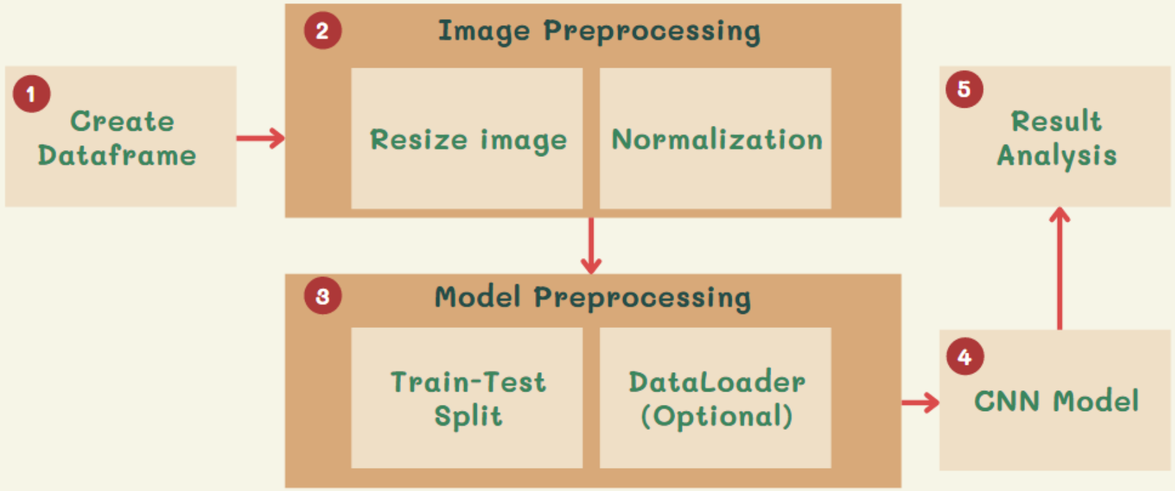
4-5. CNN Model & Result analysis



Model 3: Resnet18 (Best Model)

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

4-5. CNN Model & Result analysis



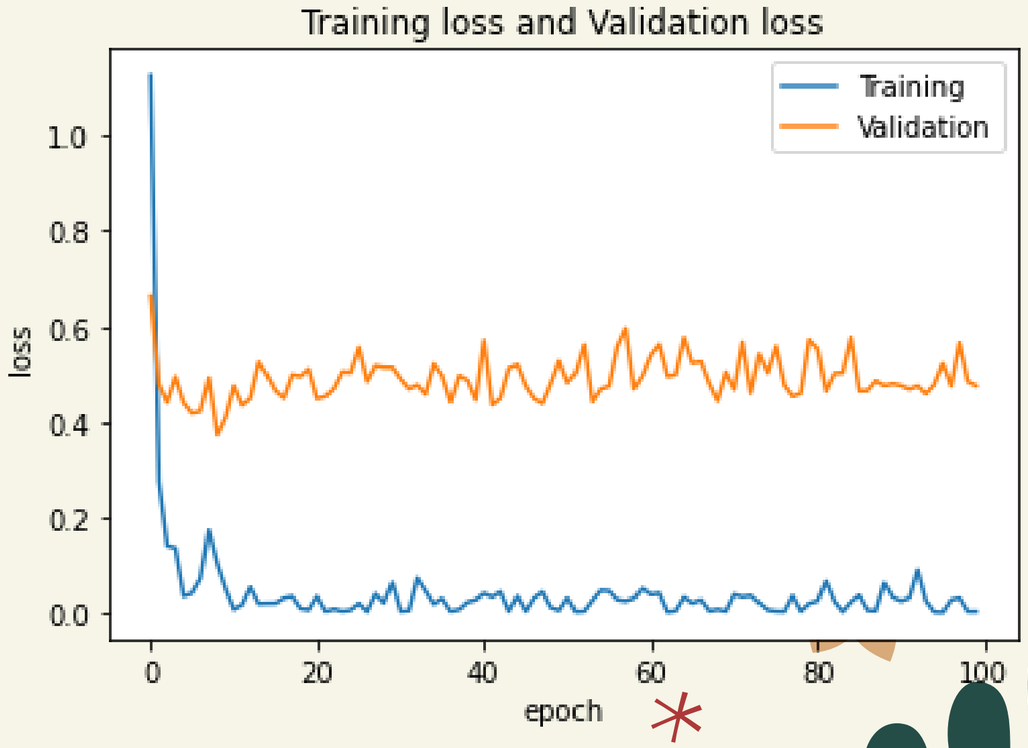
Training 1

Parameter:

optimizer = Adam
Learning rate = 0.0001
criterion = CrossEntropyLoss
scheduler = ReduceLROnPlateau
number of epochs = 100

Accuracy:

Training accuracy: 1.0
Validate accuracy: 0.89



Model Improvement

1. Use image augmentation (crop/resize/change intensity level) to create more dataset
2. Use Grid search/Random search to find the best parameter for each model

Reference

- https://www.youtube.com/playlist?list=PL5-TkQAfAZFbzxjBHtzdVCWE0Zbhomg7r
- https://www.kaggle.com/pranjalsoni17/natural-scene-classification
- https://www.analyticsvidhya.com/blog/2021/09/convolutional-neural-network-pytorch-implementation-on-cifar10-dataset/
- https://pytorch.org/tutorials/beginner/basics/buildmodel_tutorial.html
- https://www.kaggle.com/raddar/tuberculosis-chest-xrays-shenzhen
- https://www.pluralsight.com/guides/introduction-to-resnet

THANK YOU