國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Digital Medicine 2021

Case Presentation 2

COVID-19 Pneumonia Detection

2021/10/31

Chest X-Ray Image Resources

Media Advisory Wednesday, September 27, 2017

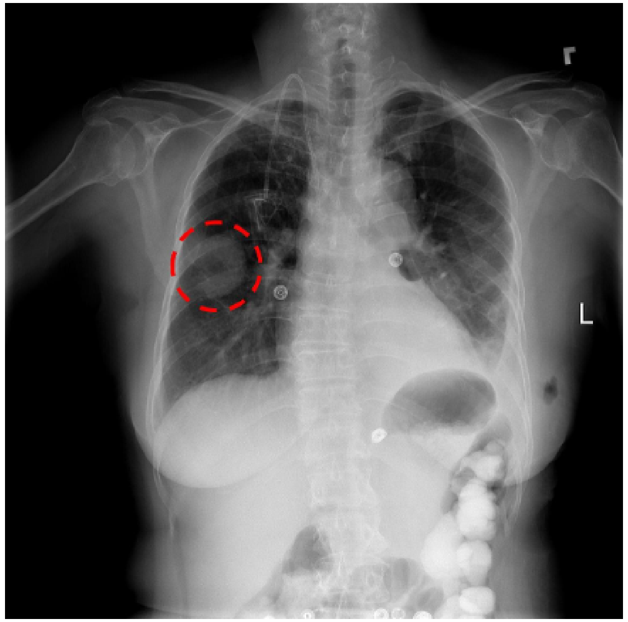
NIH Clinical Center provides one of the largest publicly available chest x-ray datasets to scientific community

The dataset of scans is from more than 30,000 patients, including many with advanced lung disease.

What

The NIH Clinical Center recently released over 100,000 anonymized chest x-ray images and their corresponding data to the scientific community. The release will allow researchers across the country and around the world to freely access the datasets and increase their ability to teach computers how to detect and diagnose disease. Ultimately, this artificial intelligence mechanism can lead to clinicians making better diagnostic decisions for patients.

NIH compiled the dataset of scans from more than 30,000 patients, including many with advanced lung disease. Patients at the NIH Clinical Center, the nation's largest hospital devoted entirely to clinical research, are partners in research and voluntarily enroll to participate in clinical trials. With patient privacy being paramount, the dataset was rigorously screened to remove all



Radiological Society of North America

RSNA®
Radiological Society
of North America

Membership | Annual Meeting | Journals | Education | Research | Practice Tools | COVID-19

RSNA® 2021
REDEFINING RADIOLOGY
November 28 to December 2

Registration is Open for RSNA 2021!

Register now and get ready to join us in Chicago for the most important week in radiology. Reserve your hotel with us for the best pricing, selection and flexibility.

Register Now

RSNA Challenge

Featured Prediction Competition

RSNA Pneumonia Detection Challenge

Can you build an algorithm that automatically detects potential pneumonia cases?

$30,000
Prize Money

Radiological Society of North America · 1,499 teams · 2 years ago

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions Late Submission

Overview

- Description
- Evaluation
- Timeline
- Prizes
- Getting Started
- Acknowledgements

In this competition, you're challenged to build an algorithm to detect a visual signal for pneumonia in medical images. Specifically, your algorithm needs to automatically locate lung opacities on chest radiographs.

Here's the backstory and why solving this problem matters.

Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. In 2015, 920,000 children under the age of 5 died from the disease. In the United States, pneumonia accounts for over 500,000 visits to emergency departments [1] and over 50,000 deaths in 2015 [2], keeping the ailment on the list of top 10 causes of death in the country.

While common, accurately diagnosing pneumonia is a tall order. It requires review of a chest radiograph (CXR) by highly trained specialists and confirmation through clinical history, vital signs and laboratory exams. Pneumonia usually manifests as an area or areas of increased opacity [3] on CXR. However, the diagnosis of pneumonia on CXR is complicated because of a number of other conditions in the lungs such as fluid overload (pulmonary edema), bleeding, volume loss (atelectasis or collapse), lung cancer, or post-radiation or surgical changes. Outside of the lungs, fluid in the pleural space (pleural effusion) also

Featured Code Competition

SIIM-FISABIO-RSNA COVID-19 Detection

Identify and localize COVID-19 abnormalities on chest radiographs

$100,000
Prize Money

Society for Imaging Informatics in Medicine (SIIM) · 229 teams · 2 months to go (2 months to go until merger deadline)

Overview Data Code Discussion Leaderboard Rules

Join Competition

Overview

- Description
- Evaluation
- Timeline
- Prizes
- Code Requirements
- Call For Models
- Acknowledgments
- Partners: HP & Intel

Five times more deadly than the flu, COVID-19 causes significant morbidity and mortality. Like other pneumonias, pulmonary infection with COVID-19 results in inflammation and fluid in the lungs. COVID-19 looks very similar to other viral and bacterial pneumonias on chest radiographs, which makes it difficult to diagnose. Your computer vision model to detect and localize COVID-19 would help doctors provide a quick and confident diagnosis. As a result, patients could get the right treatment before the most severe effects of the virus take hold.

Currently, COVID-19 can be diagnosed via polymerase chain reaction to detect genetic material from the virus or chest radiograph. However, it can take a few hours and sometimes days before the molecular test results are back. By contrast, chest radiographs can be obtained in minutes. While guidelines exist to help radiologists differentiate COVID-19 from other types of infection, their assessments vary. In addition, non-radiologists could be supported with better localization of the disease, such as with a visual bounding box.

RSNA Pneumonia Dataset (Practice)

- 1000 Chest X-Ray Images with Pneumonia vs. 1000 images without Pneumonia
- DICOM Image Format

DICOM File Format

DICOM℠

Digital Imaging and Communications in Medicine

RNSA Pneumonia Dataset (Practice)

名稱	修改日期	類型	大小
0	2021/10/31 下午 05:16	檔案資料夾	
1	2021/10/31 下午 05:17	檔案資料夾	
Codings	2020/6/7 下午 10:20	文字文件	1 KB
dcm2datastore	2020/6/8 下午 08:54	MATLAB Code	1 KB
dicompreprocess	2021/5/30 下午 11:38	MATLAB Code	1 KB
record_computation	2021/5/30 下午 11:22	MATLAB Code	4 KB

Codings - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
0: Not Pneumonia
1: Pneumonia
```

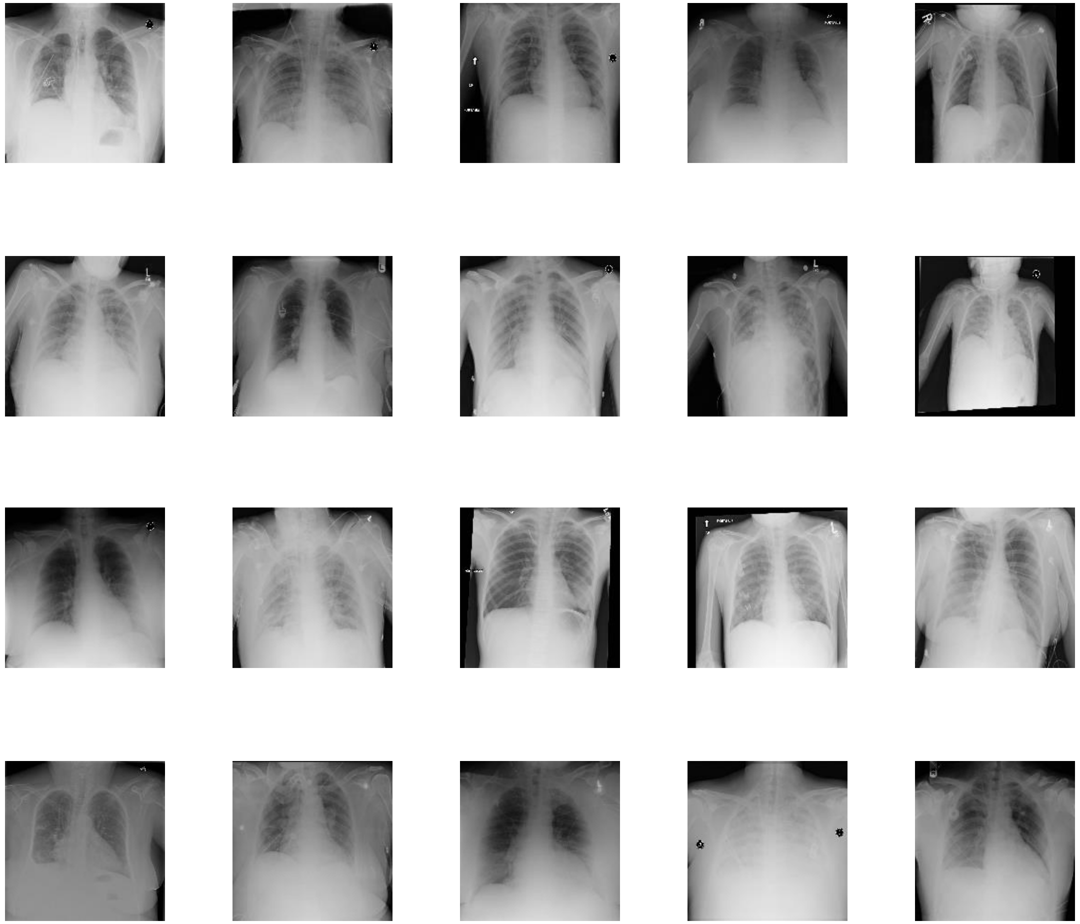
Read Image Data

- img = dicomread('00a85be6-6eb0-421d-8acf-ff2dc0007e8a.dcm');
- imshow(img)

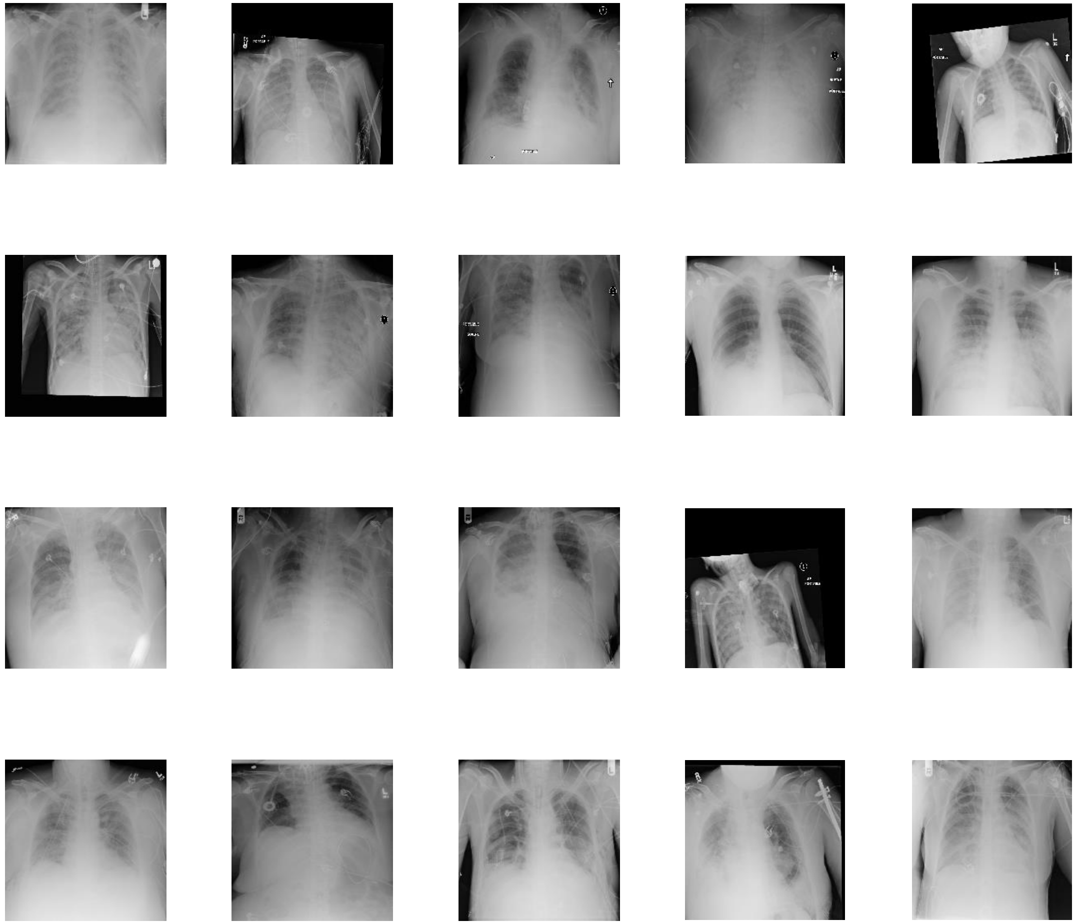
Show 20 Random Images

```
files = dir('*.dcm');
figure;
perm = randperm(1000,20);
for i = 1:20
    subplot(4,5,i);
    img = dicomread(files(perm(i)).name);
    imshow(img);
end
```

Show 20 Random Chest X-Ray without Pneumonia



Show 20 Random Chest X-Ray with Pneumonia



Check Image Size

- img = dicomread('00a85be6-6eb0-421d-8acf-ff2dc0007e8a.dcm');
- imgsize = size(img);

Create Image Datastore

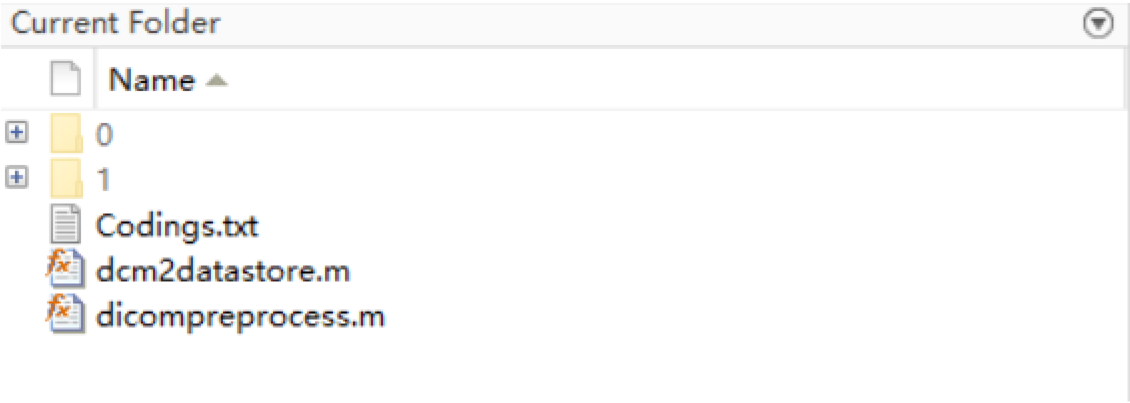
```
function imds = dcm2datastore(datapath,file_ext,label_option)

% Get folder list
dinfo = dir(datapath);
dirFlags = [dinfo.isdir];
dinfo = dinfo(dirFlags);
dinfo(ismember( {dinfo.name}, {'.', '..'})) = [];

% Initiate parameters
if length(label_option)<=1
    label_option = 0:length(dinfo)-1;
end

% Create image datastore using foldername and input file extension
filelocation = {};
for i=1:length(dinfo)
    if ismember(i-1,label_option)
        filelocation{i} = [datapath '\' dinfo(i).name];
    end
end
%imds = imageDatastore(filelocation,'FileExtensions',file_ext,'LabelSource','foldernames','ReadFcn',@dicomread);
imds = imageDatastore(filelocation,'FileExtensions',file_ext,'LabelSource','foldernames','ReadFcn',@dicompreprocess);

end
```

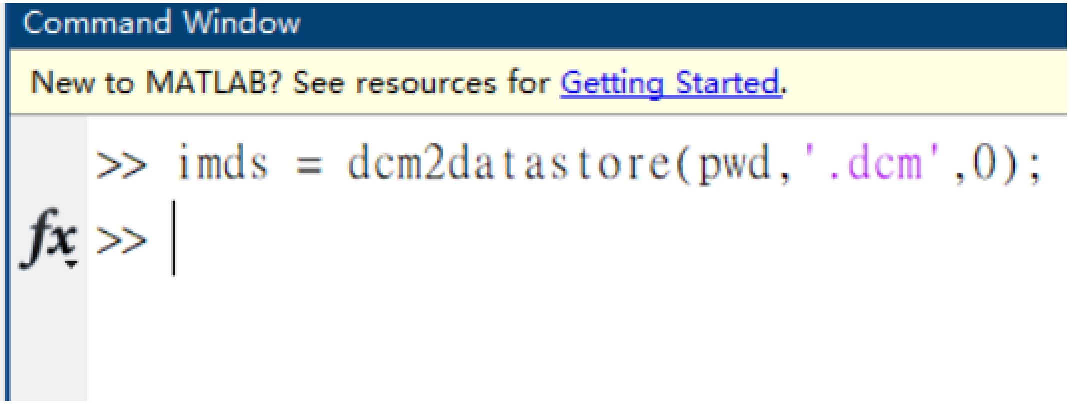
DICOM Preprocessing

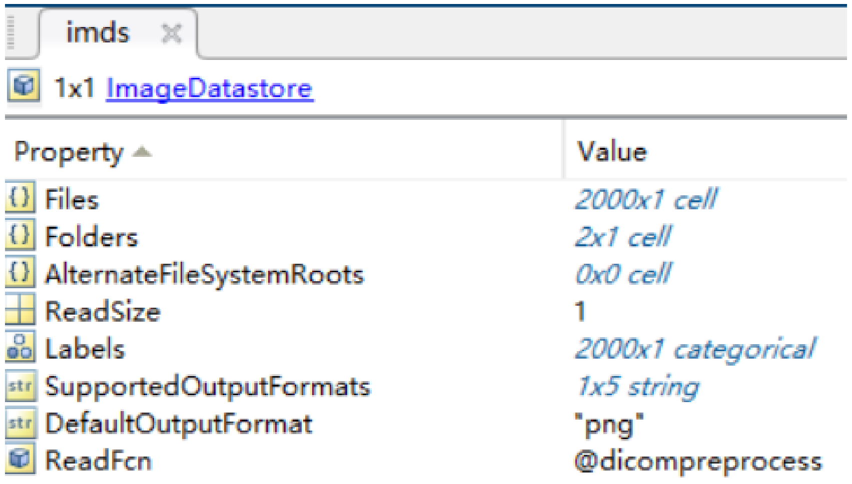
```
function output = dicompreprocess(filename)

% Code for Simple CNN model
dcm = dicomread(filename);
dcm_resize = imresize(dcm,[50 50]);
output = dcm_resize;

% Code for Transfer Learning Model
% dcm_resize = imresize(dcm,[227 227]);
% output = cat(3,dcm_resize,dcm_resize,dcm_resize);

end
```

Create Image Datastore

- imds = dcm2datastore(pwd,'.dcm',0);







Count Number of Images for Each Label

- labelCount = countEachLabel(imds);
- labelCount = labelCount.Count;
- min_labelCount = min(labelCount);

Specify Image Size

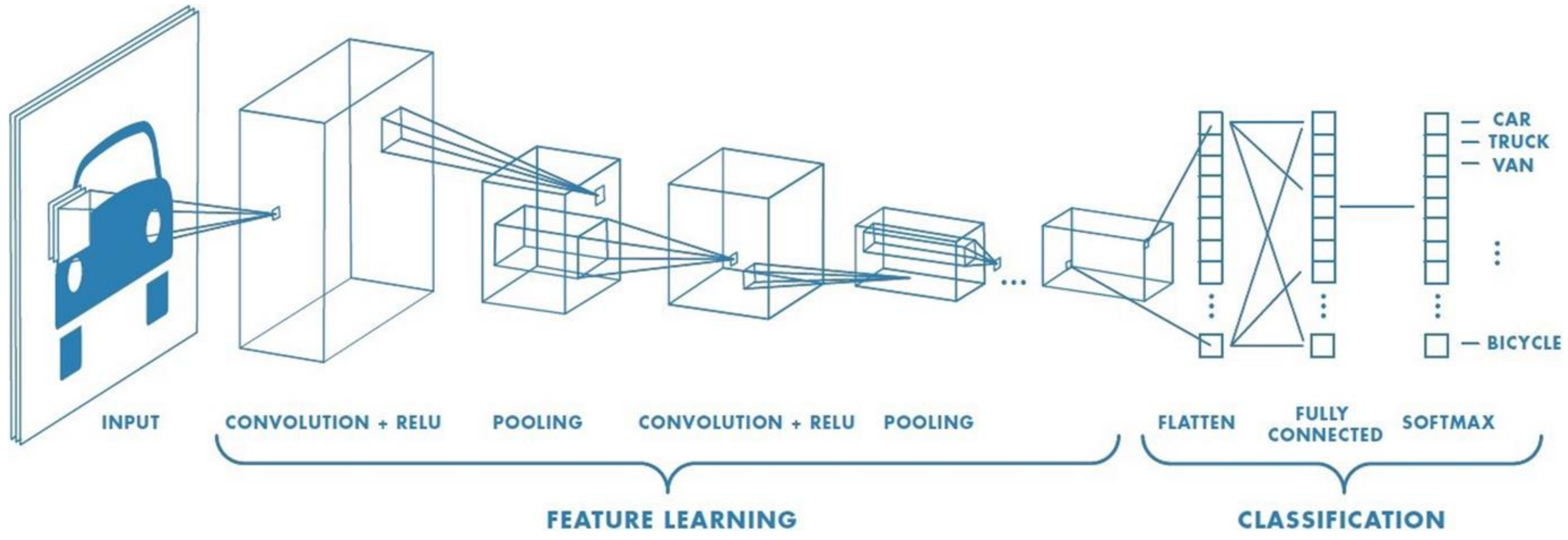
- filepath = imds.Files{1};
- img = dicompreprocess(filepath);
- imgsize = size(img);
- if length(imgsize)==2
-     imgsize(3) = 1;
- end

Specify Training and Validation Sets

- train_ratio = 0.7;
- numTrainFiles = fix(min_labelCount*train_ratio);
- [imdsTrain,imdsValidation] = splitEachLabel(imds,numTrainFiles,'randomize');

Specify Convolution Layer Parameters

- filter_size = 3;
- num_filters = 8;



Specify CNN Architechure

```
layers = [
    imageInputLayer(imgsize)

    convolution2dLayer(filter_size,num_filters,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(filter_size,num_filters*2,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(filter_size,num_filters*4,'Padding','same')
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(length(labelCount))
    softmaxLayer
    classificationLayer];
```

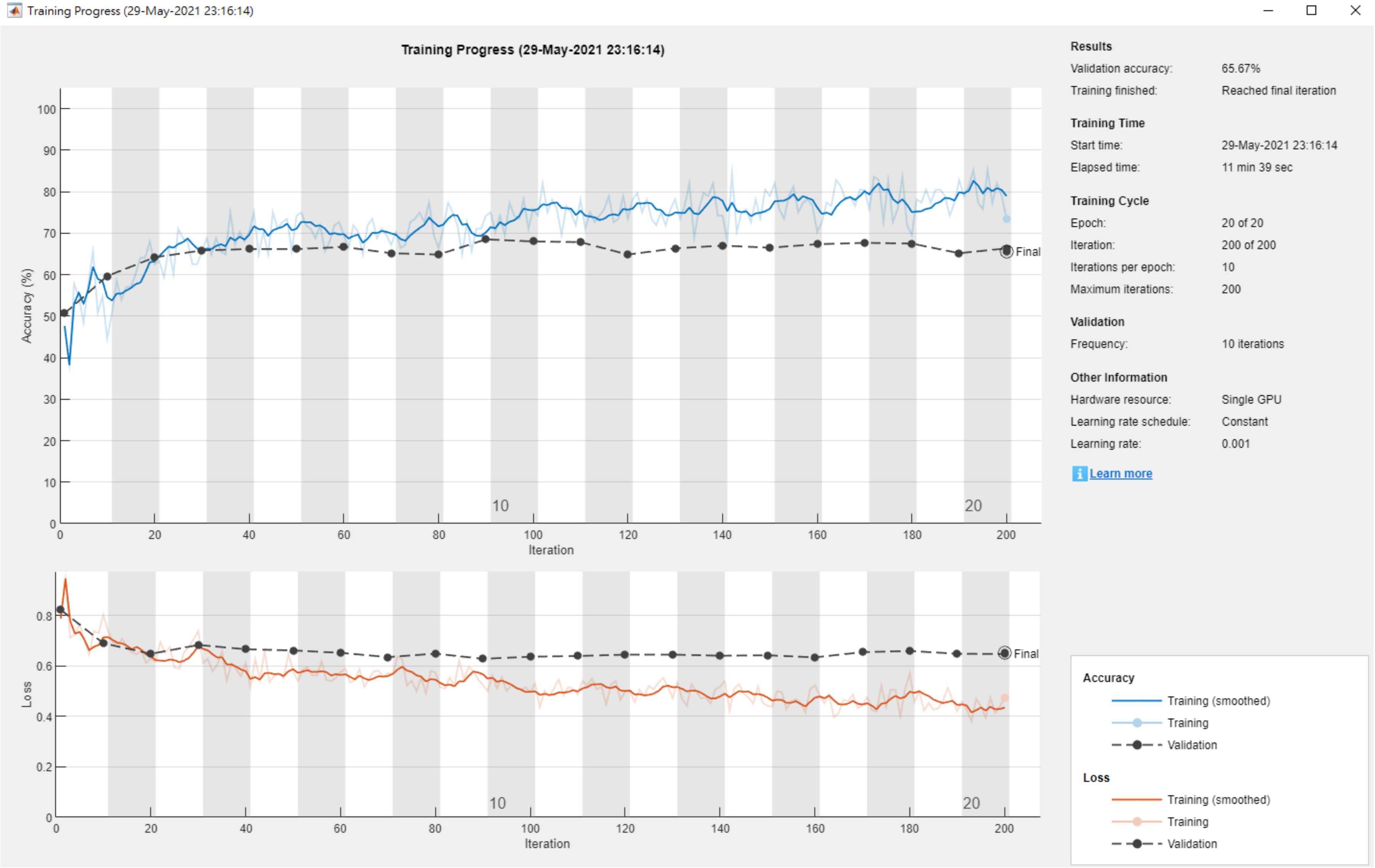
Specify Training Options

- options = trainingOptions('sgdm', ...
- 'InitialLearnRate',0.001, ...
- 'MaxEpochs',20, ...
- 'Shuffle','every-epoch', ...
- 'ValidationData',imdsValidation, ...
- 'ValidationFrequency',10, ...
- 'Verbose',false, ...
- 'Plots', 'training-progress');

Start Training

- tic;
- [net netinfo]= trainNetwork(imdsTrain,layers,options);
- toc;

Training Progress

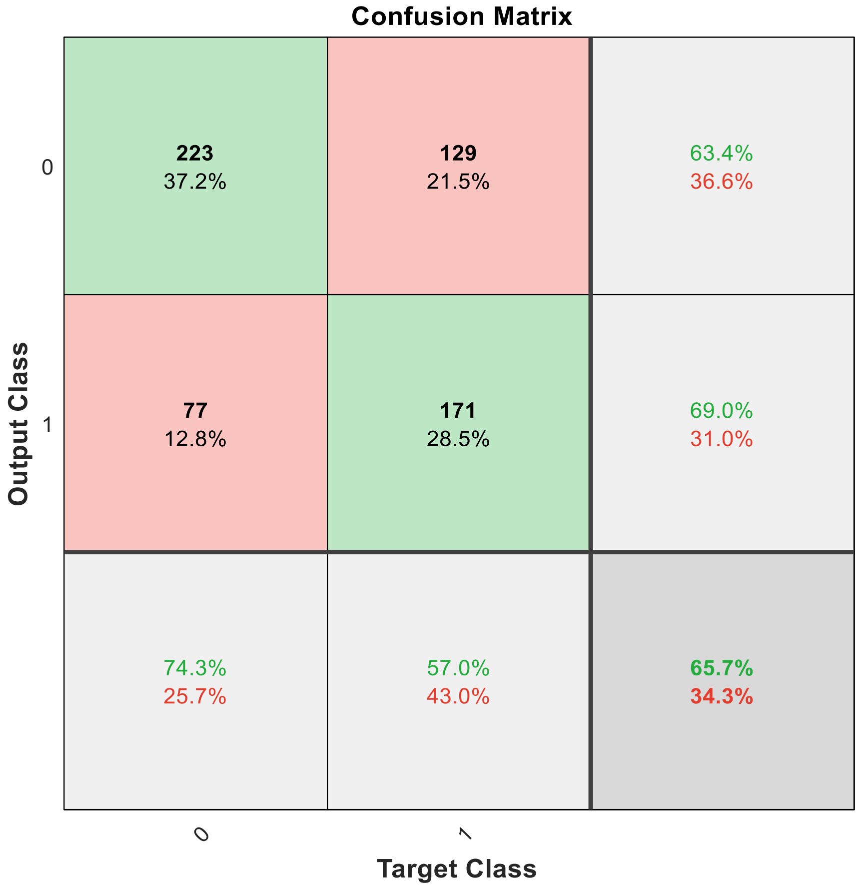


Compute Accuracy

- YPred = classify(net,imdsValidation);
- YValidation = imdsValidation.Labels;
-
- accuracy = sum(YPred == YValidation)/numel(YValidation);

Plot Confusion Matrix

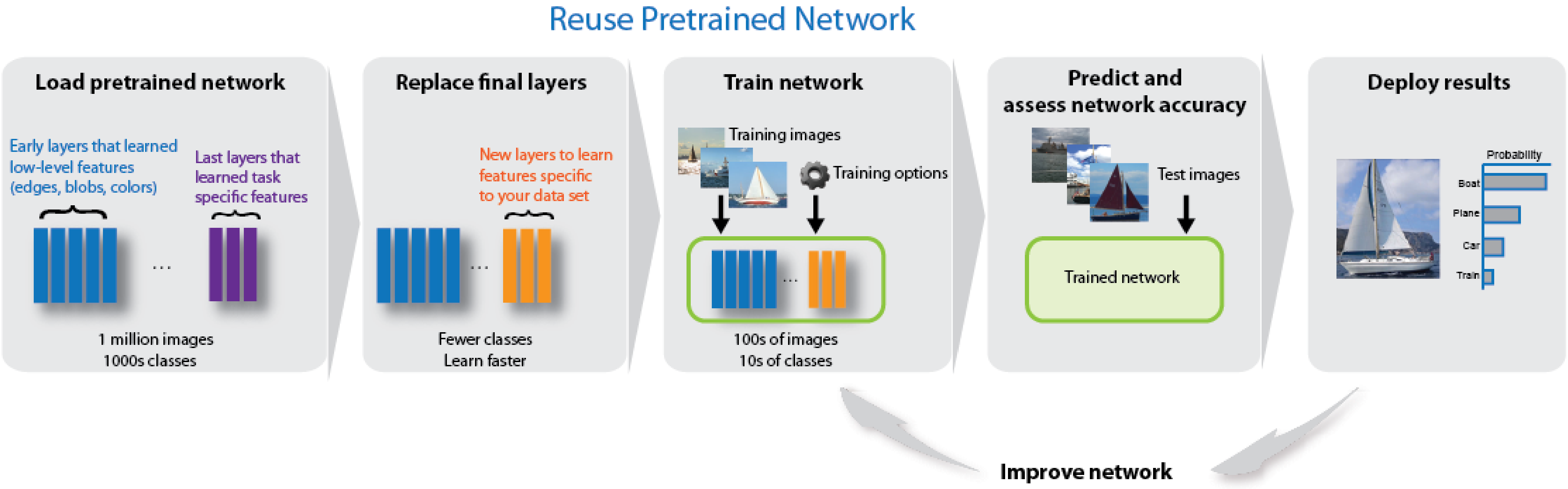
- plotconfusion(YValidation,YPred)



Run CNN at Once…

- imds = dcm2datastore(pwd,'.dcm',0);
- [net netinfo nstats] = cnn(imds,0.7,1,3,8);

Transfer Learning



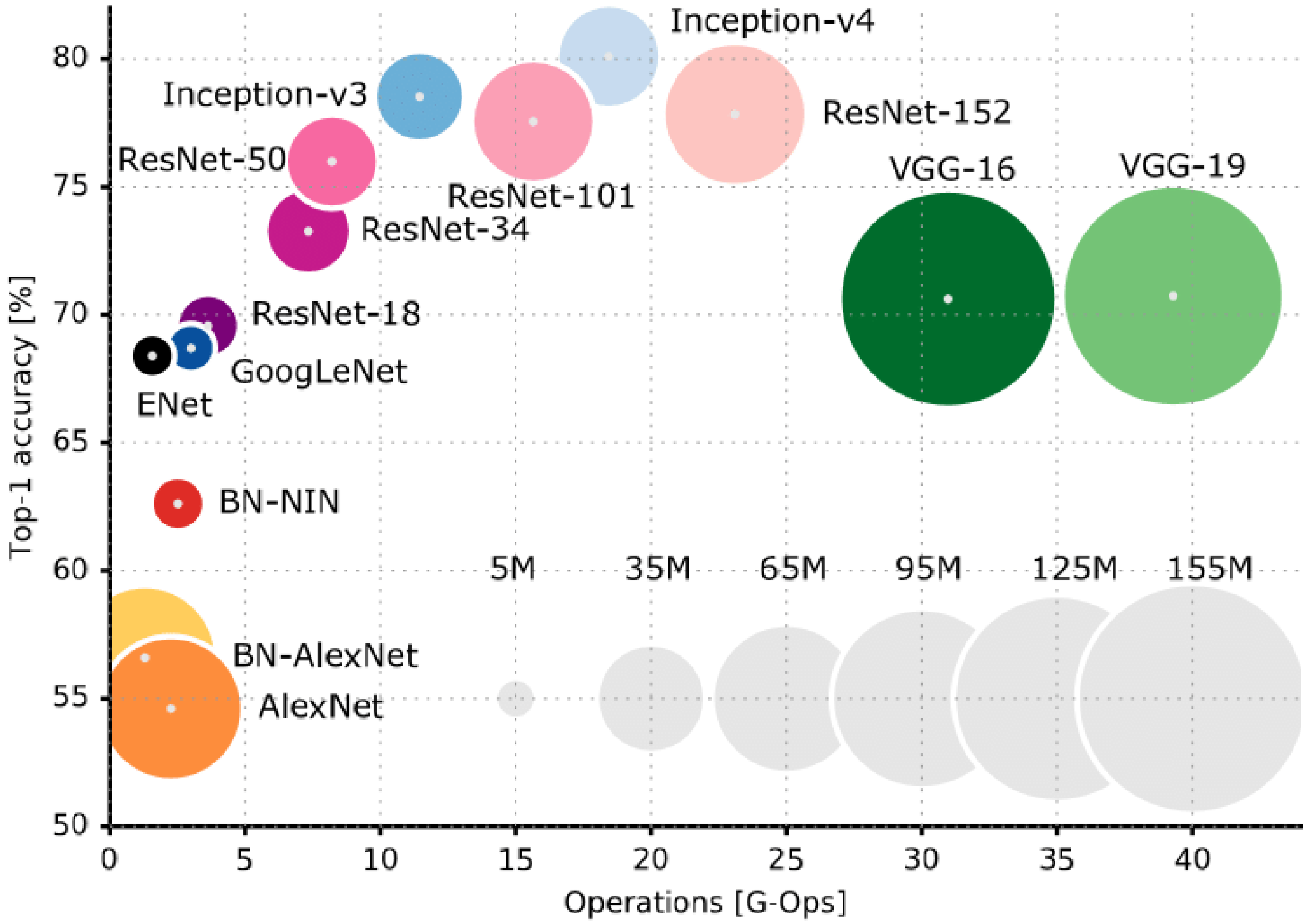
https://www.mathworks.com/help/deeplearning/ug/transfer-learning-using-alexnet.html

Apply AlexNet to Pneumonia Data

- Resize DICOM images according to the AlexNet
- Triple the one channel DICOM data to the three channels
- Run AlexNet

Pretrained Model in Matlab

- resnet18
- resnet50
- googlenet
- inceptionv3
- Densenet201
- …



Load Pretrained Network

- net = alexnet;

Analyze Pretrained Network

- analyzeNetwork(net)

Check Input Image Size

- inputSize = net.Layers(1).InputSize

```
inputSize = 1×3

   227   227     3
```

Prepare Data and Adjust Preprocessing Parameters

- imds = dcm2datastore(pwd,'.dcm',0);
- labelCount = countEachLabel(imds);
- labelCount = labelCount.Count;
- min_labelCount = min(labelCount);
- train_ratio = 0.7;
- numTrainFiles = fix(min_labelCount*train_ratio);
- [imdsTrain,imdsValidation] = splitEachLabel(imds,numTrainFiles,'randomize');

```
function output = dicompreprocess(filename)

% Code for Simple CNN model
dcm = dicomread(filename);
%dcm_resize = imresize(dcm,[50 50]);
%output = dcm_resize;

% Code for Transfer Learning Model
 dcm_resize = imresize(dcm,[277 277]);
 output = cat(3,dcm_resize,dcm_resize,dcm_resize);

end
```

Replace Final Layers

- layersTransfer = net.Layers(1:end-3);
- numClasses = numel(categories(imdsTrain.Labels))

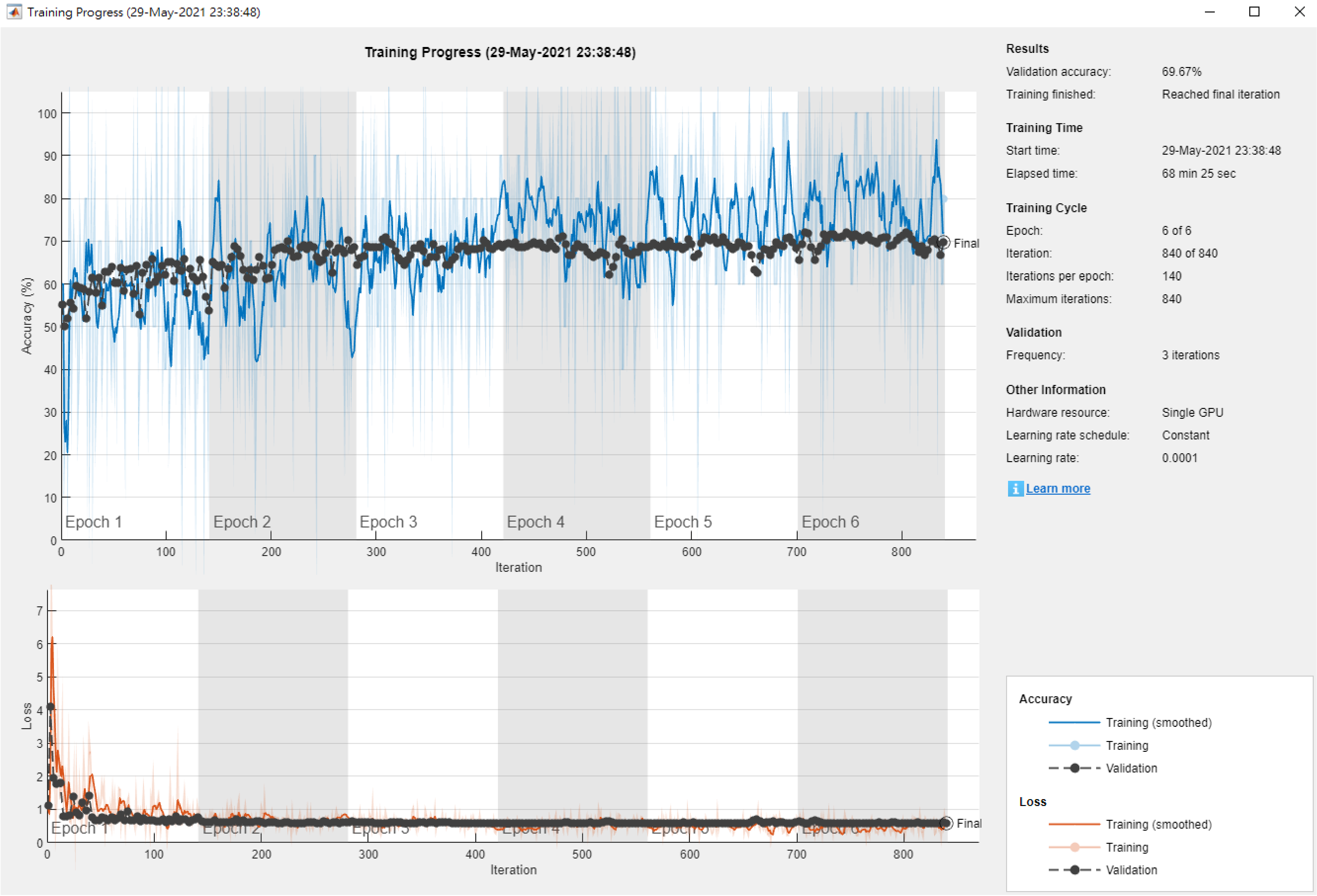
- layers = [
- layersTransfer
- fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor',20)
- softmaxLayer
- classificationLayer];

Specify Training Options

- options = trainingOptions('sgdm', ...
- 'MiniBatchSize',10, ...
- 'MaxEpochs',6, ...
- 'InitialLearnRate',1e-4, ...
- 'Shuffle','every-epoch', ...
- 'ValidationData',imdsValidation, ...
- 'ValidationFrequency',3, ...
- 'Verbose',false, ...
- 'Plots','training-progress');

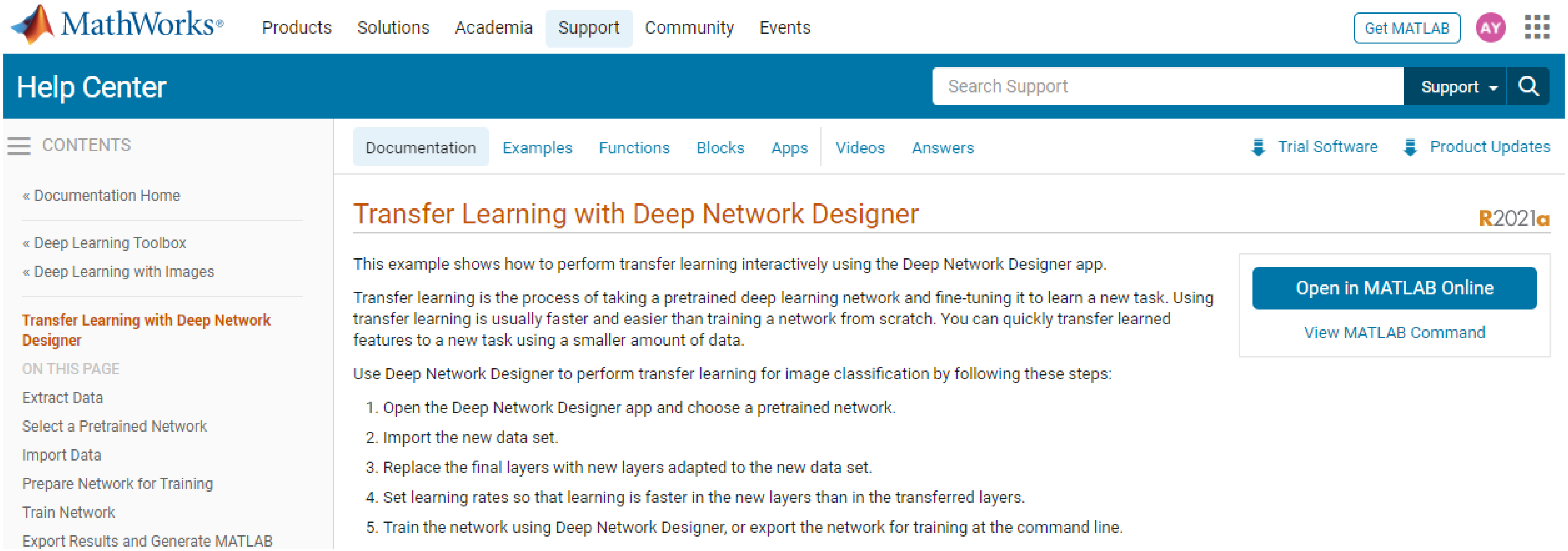
Start Training Transfer Network

- tic;
- netTransfer = trainNetwork(imdsTrain,layers,options);
- toc;


Training Progress (29-May-2021 23:38:48)
Training Progress (29-May-2021 23:38:48)
Accuracy (%)
Epoch 1
Epoch 2
Epoch 3
Epoch 4
Epoch 5
Epoch 6
Final
Iteration
Loss
Results
Validation accuracy: 69.67%
Training finished: Reached final iteration
Training Time
Start time: 29-May-2021 23:38:48
Elapsed time: 68 min 25 sec
Training Cycle
Epoch: 6 of 6
Iteration: 840 of 840
Iterations per epoch: 140
Maximum iterations: 840
Validation
Frequency: 3 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0001
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation

Matlab Deep Network Designer

- https://www.mathworks.com/help/deeplearning/ug/transfer-learning-with-deep-network-designer.html


MathWorks® Products Solutions Academia Support Community Events Get MATLAB AY

Help Center Search Support Support

CONTENTS

« Documentation Home

« Deep Learning Toolbox

« Deep Learning with Images

Transfer Learning with Deep Network Designer

ON THIS PAGE

Extract Data

Select a Pretrained Network

Import Data

Prepare Network for Training

Train Network

Export Results and Generate MATLAB

Documentation Examples Functions Blocks Apps Videos Answers Trial Software Product Updates

Transfer Learning with Deep Network Designer

R2021a

Open in MATLAB Online

View MATLAB Command

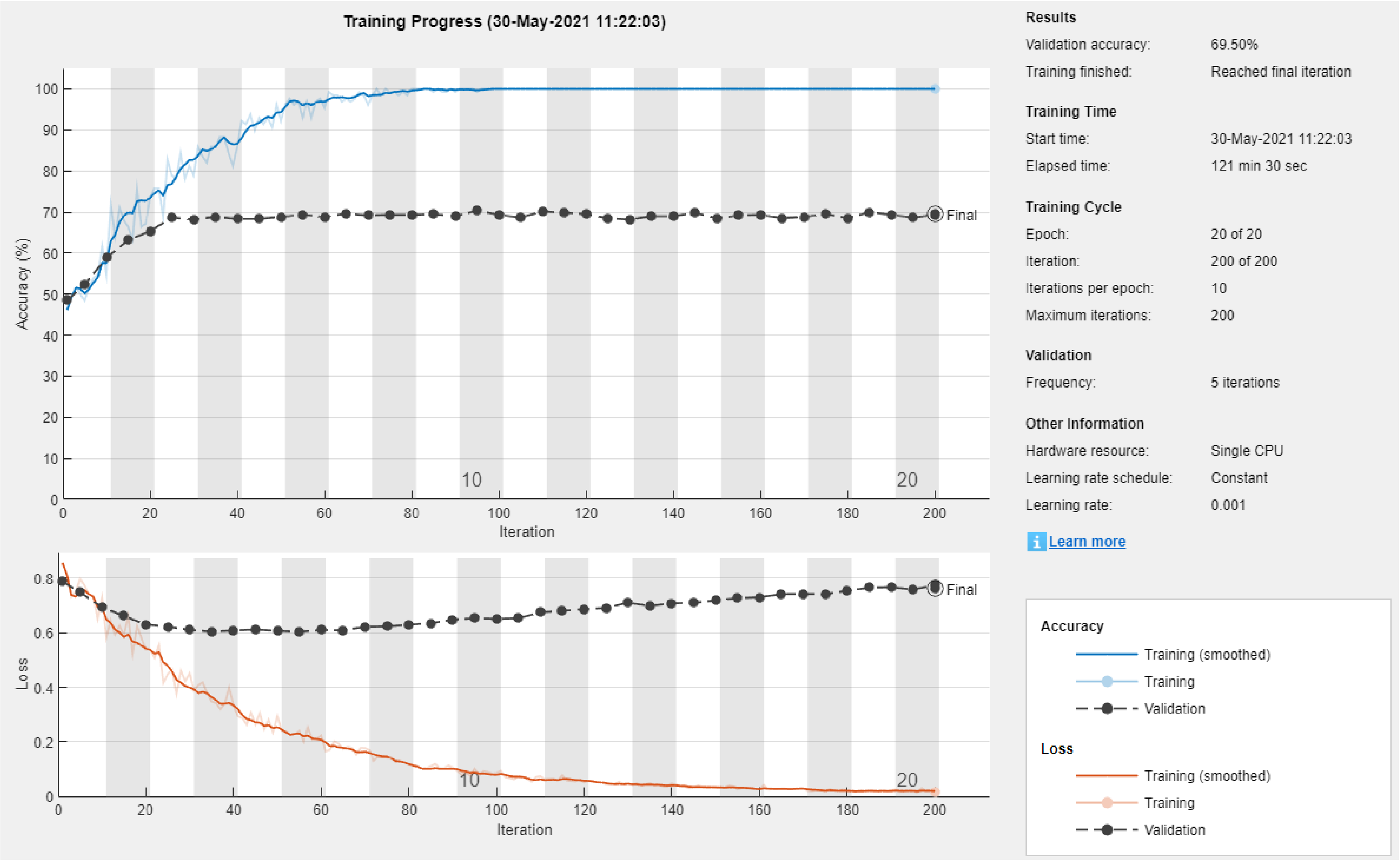
This example shows how to perform transfer learning interactively using the Deep Network Designer app.

Transfer learning is the process of taking a pretrained deep learning network and fine-tuning it to learn a new task. Using transfer learning is usually faster and easier than training a network from scratch. You can quickly transfer learned features to a new task using a smaller amount of data.

Use Deep Network Designer to perform transfer learning for image classification by following these steps:

1. Open the Deep Network Designer app and choose a pretrained network.
2. Import the new data set.
3. Replace the final layers with new layers adapted to the new data set.
4. Set learning rates so that learning is faster in the new layers than in the transferred layers.
5. Train the network using Deep Network Designer, or export the network for training at the command line.

ResNet50



Training Progress (30-May-2021 11:22:03)
Accuracy (%)
Iteration
Loss
Final
Results
Validation accuracy: 69.50%
Training finished: Reached final iteration
Training Time
Start time: 30-May-2021 11:22:03
Elapsed time: 121 min 30 sec
Training Cycle
Epoch: 20 of 20
Iteration: 200 of 200
Iterations per epoch: 10
Maximum iterations: 200
Validation
Frequency: 5 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation

COVID-19 Pneumonia Detection

- Dataset for developing AI Models
  - 400 Non-Pneumonia
  - 400 Typical Pneumonia
  - 400 Atypical Pneumonia

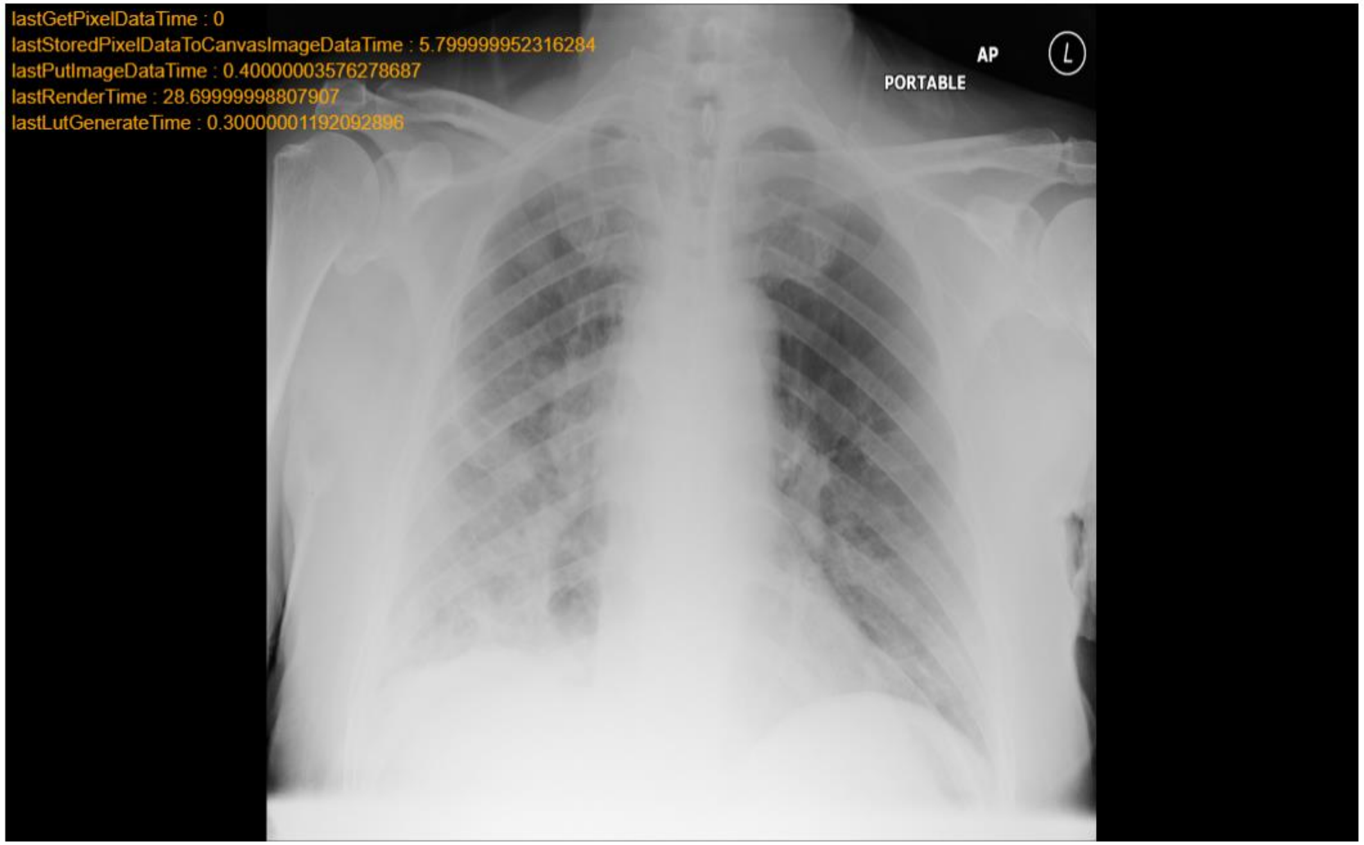
- Dataset for validating AI Models
  - 50 Non-Pneumonia
  - 50 Typical Pneumonia
  - 50 Atypical Pneumonia

	A	B	C	D
1	FileID	Negative	Typical	Atypical
2	0003b2210c64	0	1	0
3	00af6f8c2a3d	1	0	0
4	00c9033fbc2e	0	0	1
5	00e0ce73dac8	0	0	1
6	00f0a591f18a	0	1	0
7	01113d3e0910	1	0	0
8	018861e85a54	0	0	1
9	01978984ac60	0	0	1
10	01a7576432b3	1	0	0
11	01ef587469f2	0	0	1
12	021f1372c819	0	0	1
13	026427c2156b	0	1	0
14	02a3e261c938	0	1	0
15	03379b1a9e12	0	0	1
16	034247332fec	1	0	0
17	038cd47a6ab8	0	0	1
18	039308b26a85	0	0	1
19	03bec103ef51	0	1	0
20	03f9dafeb772	1	0	0

Chest X-Ray AI Assessment

Click "Choose File" and select a Chest X-Ray image on your local file system or drag and drop a Chest X-Ray image.

選擇檔案 | 選擇檔案



Tentative Assessment:

Lungs:
Presence of lung infiltration.
Presence of emphysema.
Presence of lung fibrosis.
Typical presentation of pneumonia.

Pleura:
Pleural effusion is suspected.

Heart:
Heart size is within normal limit.

Filename:0e91afec-c246-450b-9f10-e07a75ae0086.dcm

https://www.chestxray-ai.com/

Digital Medicine Center at National Yang Ming Chiao Tung University