

Data Collection

```
clear;clc;
animal = imageDatastore('Animal','IncludeSubfolders',true,...
    "LabelSource","foldernames");
[training, validation]=splitEachLabel(animal,0.8)
```

```
training =
    ImageDatastore with properties:
```

```
    Files: {
        '...\GitHub\Online-Matlab-Training-2020\Animal\Cat\download (16).jpg';
        '...\GitHub\Online-Matlab-Training-2020\Animal\Cat\download (17).jpg';
        '...\GitHub\Online-Matlab-Training-2020\Animal\Cat\download (18).jpg'
        ... and 37 more
    }
    Folders: {
        'C:\Users\Master\Documents\GitHub\Online-Matlab-Training-2020\Animal'
    }
    Labels: [Cat; Cat; Cat ... and 37 more categorical]
    AlternateFileSystemRoots: {}
    ReadSize: 1
    SupportedOutputFormats: ["png"    "jpg"    "jpeg"    "tif"    "tiff"]
    DefaultOutputFormat: "png"
    ReadFcn: @readDatastoreImage
```

```
validation =
    ImageDatastore with properties:
```

```
    Files: {
        '...\GitHub\Online-Matlab-Training-2020\Animal\Cat\images (44).jpg';
        '...\GitHub\Online-Matlab-Training-2020\Animal\Cat\images (45).jpg';
        '...\GitHub\Online-Matlab-Training-2020\Animal\Dog\images (46).jpg'
        ... and 6 more
    }
    Folders: {
        'C:\Users\Master\Documents\GitHub\Online-Matlab-Training-2020\Animal'
    }
    Labels: [Cat; Cat; Dog ... and 6 more categorical]
    AlternateFileSystemRoots: {}
    ReadSize: 1
    SupportedOutputFormats: ["png"    "jpg"    "jpeg"    "tif"    "tiff"]
    DefaultOutputFormat: "png"
    ReadFcn: @readDatastoreImage
```

Preprocessing Size Normalization

```
augTraining = augmentedImageDatastore([200 200 3],training);
augValidation = augmentedImageDatastore([200 200 3],validation);
```

Layer / Architecture creation for CNN

```
layers = [imageInputLayer([200 200 3])
    convolution2dLayer(12,120)
    reluLayer
    maxPooling2dLayer(3,'Stride',3)

    fullyConnectedLayer(3)
    softmaxLayer
    classificationLayer];
```

training option

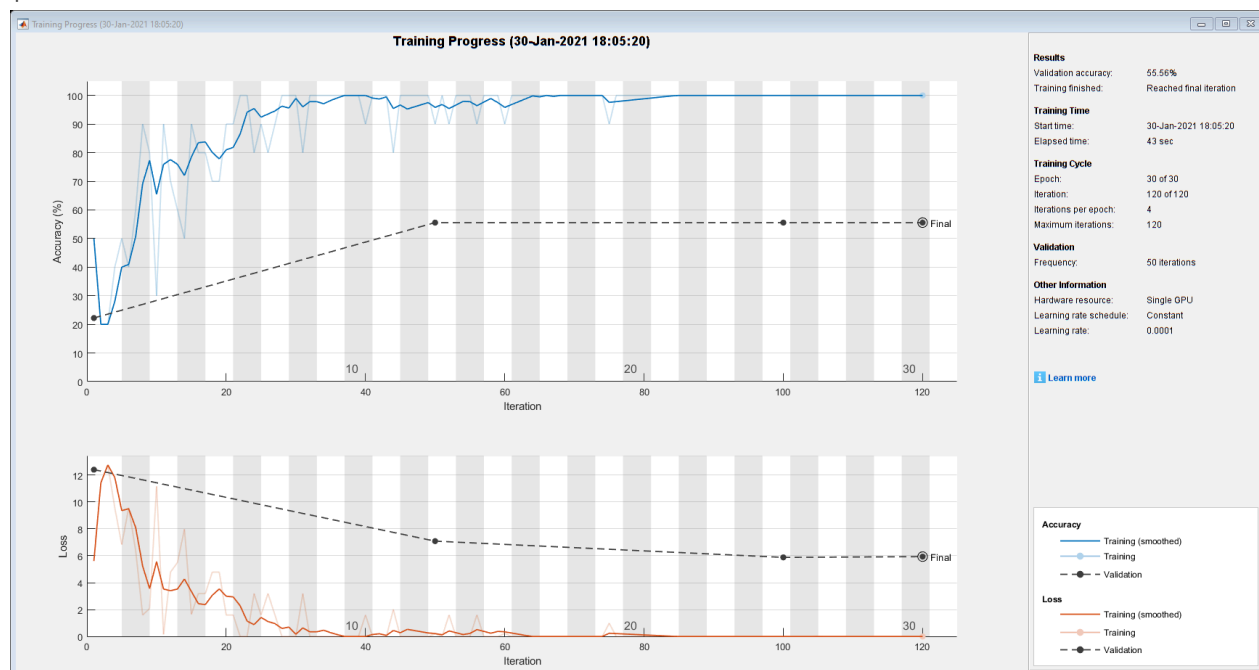
```
opts = trainingOptions('adam', ...  
    'ExecutionEnvironment','gpu',...  
    'MaxEpochs',30, ...  
    'MiniBatchSize',10,...  
    'InitialLearnRate',.0001,...  
    'Shuffle','every-epoch', ...  
    'Plots','training-progress', ...  
    'ValidationData',augValidation);
```

Training

```
AnimalNet = trainNetwork(augTraining, layers, opts);
```

Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:16	50.00%	22.22%	5.6096	12.3996	1.0000e-
13	50	00:00:28	90.00%	55.56%	0.1890	7.0855	1.0000e-
25	100	00:00:39	100.00%	55.56%	-0.0000e+00	5.8784	1.0000e-
30	120	00:00:43	100.00%	55.56%	-0.0000e+00	5.9352	1.0000e-



save **AnimalNet**

```
testdata= imageDatastore('testimage',"IncludeSubfolders",true,...  
    "LabelSource","foldernames")
```

testdata =
ImageDatastore with properties:

```

Files: {
    ' ...\Documents\GitHub\Online-Matlab-Training-2020\testimage\Cat\3 (1).jpg';
    ' ...\Documents\GitHub\Online-Matlab-Training-2020\testimage\Cat\3 (2).jpg';
    ' ...\Documents\GitHub\Online-Matlab-Training-2020\testimage\Cat\3 (3).jpg'
    ... and 6 more
}
Folders: {
    'C:\Users\Master\Documents\GitHub\Online-Matlab-Training-2020\testimage'
}
Labels: [Cat; Cat; Cat ... and 6 more categorical]
AlternateFileSystemRoots: {}
ReadSize: 1
SupportedOutputFormats: ["png"    "jpg"    "jpeg"    "tif"    "tiff"]
DefaultOutputFormat: "png"
ReadFcn: @readDatastoreImage

```

testdata.Files

```

ans = 9x1 cell
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...

```

TrueResult =testdata.Labels

```

TrueResult = 9x1 categorical
Cat
Cat
Cat
Cat
Dog
Dog
Dog
Fish
Fish

```

```

augTest = augmentedImageDatastore([200 200 3],testdata);
ypred =classify(AnimalNet,augTest)

```

```

ypred = 9x1 categorical
Dog
Fish
Dog
Fish
Dog
Dog
Dog
Cat
Fish

```

```

acurracy = 100*sum(TrueResult==ypred)/length(TrueResult)

```

```

acurracy = 44.4444

```

AlexNet Transfer learning Model

Transfer Learning Model on AlexNet

Load data

```
clear;clc;
animal = imageDatastore('Animal','IncludeSubfolders',true,...
    "LabelSource","foldernames");
animal.Labels;
[training, validation]=splitEachLabel(animal,0.8);
```

Preprocessing Size Normalization

```
augTraining = augmentedImageDatastore([227 227 3],training)
```

```
augTraining =
    augmentedImageDatastore with properties:
```

```
    NumObservations: 40
           Files: {40x1 cell}
AlternateFileSystemRoots: {}
    MiniBatchSize: 128
    DataAugmentation: 'none'
    ColorPreprocessing: 'none'
           OutputSize: [227 227]
    OutputSizeMode: 'resize'
DispatchInBackground: 0
```

```
augValidation = augmentedImageDatastore([227 227 3],validation);
```

load the network

```
Animal = alexnet
```

```
Animal =
    SeriesNetwork with properties:
```

```
    Layers: [25x1 nnet.cnn.layer.Layer]
InputNames: {'data'}
OutputNames: {'output'}
```

Modification

```
layer = Animal.Layers
```

```
layer =
    25x1 Layer array with layers:
```

1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [0 0 0 0]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]

10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3x3x192 convolutions with stride [1 1] and padding
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3x3x192 convolutions with stride [1 1] and padding
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	'fc8'	Fully Connected	1000 fully connected layer
24	'prob'	Softmax	softmax
25	'output'	Classification Output	crossentropyex with 'tench' and 999 other classes

```
layer(end-2)=fullyConnectedLayer(3, 'Name', 'FC8');
layer(end-1)=softmaxLayer;
layer(end)= classificationLayer;
layer
```

```
layer =
    25x1 Layer array with layers:
```

1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5x5x48 convolutions with stride [1 1] and padding
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3x3x192 convolutions with stride [1 1] and padding
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3x3x192 convolutions with stride [1 1] and padding
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	'FC8'	Fully Connected	3 fully connected layer
24	' '	Softmax	softmax
25	' '	Classification Output	crossentropyex

```
Feature = activations(Animal,augTraining, 'fc7', 'outputAs', 'rows')
```

```
Feature = 40x4096 single matrix
```

```
-8.8818    2.3330    7.2759    1.4746   -10.6424   -6.3625    0.7598   -8.5189 ...
-5.5060   -2.9293    6.0321   -0.5590   -6.4372   -9.9942   -6.1717   -4.4854
-6.8229    3.4013   14.4665   -1.6858    1.2201    1.2401    1.2348   -2.8175
-7.7028    1.6325   13.7761    3.3604   -6.1240   -2.0845    2.5479   -3.8642
-2.5616  -10.0646    7.2906   -6.1556   -1.0970   -6.3276   -7.4261    1.9720
-1.4215   -1.4443    3.8550   -0.8108    2.6760   -1.7222    0.5920    0.7704
-4.8340    5.9780    5.4808    0.7784   -0.6616   -2.9787    5.6973    0.6709
-4.0687   -3.3364    0.3390   -2.5603    0.9639    2.5717    1.3966    5.3310
-1.8797   -4.7819    5.5507    5.3286   -0.5373   -3.1587    0.8290   -3.8505
-2.9863    0.7112    6.3555   -0.0189   -1.1070   -0.9814    1.0461   -0.6549
```

:

```
Data = array2table(Feature);
```

option setting

```
opts = trainingOptions('adam', ...  
    'ExecutionEnvironment','gpu',...  
    'MaxEpochs',30, ...  
    'MiniBatchSize',10,...  
    'InitialLearnRate',.0001,...  
    'Shuffle','every-epoch', ...  
    'Plots','training-progress', ...  
    'ValidationData',augValidation);
```

Training

```
% Animal= trainNetwork(augTraining,layer,opts);  
% save Animal
```

testing

```
testdata= imageDatastore('testimage',"IncludeSubfolders",true,...  
    "LabelSource","foldernames")
```

```
testdata =  
    ImageDatastore with properties:  
  
        Files: {  
            ' ...\Documents\GitHub\Online-Matlab-Training-2020\testimage\Cat\3 (1).jpg';  
            ' ...\Documents\GitHub\Online-Matlab-Training-2020\testimage\Cat\3 (2).jpg';  
            ' ...\Documents\GitHub\Online-Matlab-Training-2020\testimage\Cat\3 (3).jpg'  
            ... and 6 more  
        }  
        Folders: {  
            'C:\Users\Master\Documents\GitHub\Online-Matlab-Training-2020\testimage'  
        }  
        Labels: [Cat; Cat; Cat ... and 6 more categorical]  
    AlternateFileSystemRoots: {}  
        ReadSize: 1  
    SupportedOutputFormats: ["png"    "jpg"    "jpeg"    "tif"    "tiff"]  
        DefaultOutputFormat: "png"  
        ReadFcn: @readDatastoreImage
```

testdata.Files

```
ans = 9x1 cell  
'C:\Users\Master\Documents\G...  
'C:\Users\Master\Documents\G...  
'C:\Users\Master\Documents\G...
```

```
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
'C:\Users\Master\Documents\G...
```

```
TrueResult =testdata.Labels
```

```
TrueResult = 9x1 categorical
Cat
Cat
Cat
Cat
Dog
Dog
Dog
Fish
Fish
```

```
augTest = augmentedImageDatastore([227 227 3],testdata);
Feature= activations(Animal,augTest,'fc7','outputAs','rows')
```

```
Feature = 9x4096 single matrix
-3.7040 -4.0207 6.9389 -4.1380 -1.5438 -4.7077 0.7486 -4.9141 ...
-6.2373 -0.0359 2.9892 -0.7966 -1.3321 -2.2590 -5.5211 -4.5043
-3.8468 1.9625 3.5403 -2.1399 -3.3939 -8.3012 -2.3898 -0.4123
-3.5776 0.7790 1.9836 2.0885 0.6192 -0.1691 -0.5863 2.9320
-4.6737 -10.7557 -2.6639 -2.6674 -2.6928 3.0222 -5.0183 -5.2023
-8.6383 -2.4282 7.7503 4.5747 -3.4632 2.1686 -0.3550 -0.2040
-2.7102 -9.5178 -6.3956 -5.3575 1.3930 -0.2508 -9.9938 -0.6837
7.2297 -5.2218 6.0790 -4.1648 1.5208 -6.9712 -4.1483 -3.6217
2.5376 -5.2789 -2.1645 3.4112 -5.5597 1.6672 -4.5795 -9.9543
```

```
%
% ypred =classify(Animal,augTest)
%
% accuracy = 100*sum(TrueResult==ypred)/length(TrueResult)
```