

```
% Caricamento della rete VGG16 precedentemente allenata su Cats e Dogs
my_vgg16_trained = trainedNetwork_1
```

```
my_vgg16_trained =
  SeriesNetwork with properties:

    Layers: [41x1 nnet.cnn.layer.Layer]
    InputNames: {'input'}
    OutputNames: {'classoutput'}
```

```
% Indicare il percorso alla cartella contenente il test set
test_folder = "cats_and_dogs_small/testing/"
```

```
test_folder =
"cats_and_dogs_small/testing/"
```

```
% Estrazione del primo layer della VGG16 (utile al comando successivo)
layer1 = my_vgg16_trained.Layers(1)
```

```
layer1 =
  ImageInputLayer with properties:

    Name: 'input'
    InputSize: [224 224 3]
    SplitComplexInputs: 0

    Hyperparameters
      DataAugmentation: 'none'
      Normalization: 'zerocenter'
      NormalizationDimension: 'auto'
      Mean: [1x1x3 single]
```

```
% Estrazione della dimensione dell'immagine che il layer si aspetta in input
sz = layer1.InputSize
```

```
sz = 1x3
    224    224     3
```

```
% Estrazione di (w) altezza, (w) larghezza e (c) numero di canali
h = sz(1)
```

```
h = 224
```

```
w = sz(2)
```

```
w = 224
```

```
c = sz(3)
```

```
c = 3
```

```
% Caricamento delle informazioni sulle immagini di test (nome di ogni file, label
associata a ogni immagine, ..)
```

```
imds =  
imageDatastore(test_folder,"LabelSource","foldernames","IncludeSubfolders",true)
```

```

imds =
  ImageDatastore with properties:

    Files: {
        '...\cats_and_dogs_small\testing\cat\Rio-padova-gatto-5.jpg';
        '...\EsercitazioneUnimc\cats_and_dogs_small\testing\cat\images (10).jpg';
        '...\EsercitazioneUnimc\cats_and_dogs_small\testing\cat\images (4).jpg'
        ... and 17 more
    }
    Folders: {
        '...\ema-b\Documents\MATLAB\EsercitazioneUnimc\cats_and_dogs_small\testing'
    }
    Labels: [cat; cat; cat ... and 17 more categorical]
    AlternateFileSystemRoots: {}
    ReadSize: 1
    SupportedOutputFormats: ["png"    "jpg"    "jpeg"    "tif"    "tiff"]
    DefaultOutputFormat: "png"
    ReadFcn: @readDatastoreImage

```

```
% Funzione che effettua il ridimensionamento automatico delle
% immagini in base ad h,w specificati
imds.ReadFcn = @(loc)imresize(imread(loc),[h,w]);
```

```
% Predizione della VGG16 su tutte le immagini di test
```

```
labels_predicted = classify(my_vgg16_trained, imds)
```

[illegible]

```
% Confronto tra label vere e label predette e Stampa della Matrice di
% Confusione
confusion matrix = plotconfusion(imds.Labels, labels_predicted)
```

```
confusion_matrix =  
    Figure (PLOTCONFUSION1) with properties:  
        Number: 1  
        Name: 'Confusion (plotconfusion)'  
        Color: [0.9400 0.9400 0.9400]  
        Position: [383 84 600 600]  
        Units: 'pixels'
```

Show all properties

```
% Salvataggio matrice di confusione come immagine in formato .png  
saveas(confusion_matrix, 'confusion_matrix.png')
```

