

Quick start

If you are new to MatConvNet, cut & paste the following code in a MATLAB window to try out MatConvNet. The code downloads and compiles MatConvNet, downloads a pre-trained CNN, and uses the latter to classify one of MATLAB stock images.

This example requires MATLAB to be interfaced to a C/C++ compiler (try `mex -setup` if you are unsure). Depending on your Internet connection speed, downloading the CNN model may require some time.

```
% install and compile MatConvNet (needed once)
untar('http://www.vlfeat.org/matconvnet/download/matconvnet-1.0-beta17.tar.gz') ;
cd matconvnet-1.0-beta17
run matlab/vl_compilenn

% download a pre-trained CNN from the web (needed once)
urlwrite(...
    'http://www.vlfeat.org/matconvnet/models/imagenet-vgg-f.mat',
    ...
    'imagenet-vgg-f.mat') ;

% setup MatConvNet
run matlab/vl_setupnn

% load the pre-trained CNN
net = load('imagenet-vgg-f.mat') ;

% load and preprocess an image
im = imread('peppers.png') ;
im_ = single(im) ; % note: 0-255 range
im_ = imresize(im_, net.meta.normalization.imageSize(1:2)) ;
im_ = im_ - net.meta.normalization.averageImage ;

% run the CNN
res = vl_simplenn(net, im_) ;

% show the classification result
scores = squeeze(gather(res(end).x)) ;
[bestScore, best] = max(scores) ;
figure(1) ; clf ; imagesc(im) ;
title(sprintf('%s (%d), score %.3f', ...
    net.meta.classes.description{best}, best, bestScore)) ;
```

In order to compile the GPU support and other advanced features, see the installation instructions (`../install/`).

Using DAG models

The example above exemplifies using a model using the SimpleNN wrapper. More complex models use instead the DagNN wrapper. For example, to run GoogLeNet use:

```
% download a pre-trained CNN from the web (needed once)
urlwrite(...
    'http://www.vlfeat.org/matconvnet/models/imagenet-googlenet-dag.mat', ...
    'imagenet-googlenet-dag.mat') ;

% load the pre-trained CNN
net = dagnn.DagNN.loadobj(load('imagenet-googlenet-dag.mat')) ;

% load and preprocess an image
im = imread('peppers.png') ;
im_ = single(im) ; % note: 0-255 range
im_ = imresize(im_, net.meta.normalization.imageSize(1:2)) ;
im_ = im_ - net.meta.normalization.averageImage ;

% run the CNN
net.eval({'data', im_}) ;

% obtain the CNN output
scores = net.vars(net.getVarIndex('prob')).value ;
scores = squeeze(gather(scores)) ;

% show the classification results
[bestScore, best] = max(scores) ;
figure(1) ; clf ; imagesc(im) ;
title(sprintf('%s (%d), score %.3f', ...
    net.meta.classes.description{best}, best, bestScore)) ;
```