Pretrained models

This section describes how pre-trained models can be downloaded and used in MatConvNet. Using the pre-trained model is easy; just start from the example code included in the quickstart guide (../quick/).

> **Remark:** The following CNN models may have been *imported from other reference implementations* and are equivalent to the originals up to numerical precision. However, note that:
>
> 1. Images need to be pre-processed (resized and cropped) before being submitted to a CNN for evaluation. Even small differences in the prepreocessing details can have a non-negligible effect on the results.
> 2. The example below shows how to evaluate a CNN, but does not include data augmentation or encoding normalization as for example provided by the VGG code (http://www.robots.ox.ac.uk/~vgg/research/deep_eval). While this is easy to implement, it is not done automatically here.
> 3. These models are provided here for convenience, but please credit the original authors.

Face recognition

These models are trained for face classification and verification.

- **VGG-Face**. The face classification and verification network from the VGG project (http://www.robots.ox.ac.uk/~vgg/software/vgg_face/).

  > Deep face recognition, O. M. Parkhi and A. Vedaldi and A. Zisserman, Proceedings of the British Machine Vision Conference (BMVC), 2015 (paper (http://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf)).

  - vgg-face (../models/vgg-face.mat) (../models/vgg-face.svg)

  See the script `examples/cnn_vgg_face.m` for an example of using VGG-Face for classification. To use this network for face verification instead, extract the 4K dimensional features by removing the last classification layer and normalize the resulting vector in L2 norm.

Semantic segmentation

These models are trained for semantic image segmentation using the PASCAL VOC category definitions.

- **Fully-Convolutional Networks** (FCN) training and evaluation code is available here (https://github.com/vlfeat/matconvnet-fcn).
- **BVLC FCN** (the original implementation) imported from the Caffe version (https://github.com/BVLC/caffe/wiki/Model-Zoo) [*DagNN format*].

  > 'Fully Convolutional Models for Semantic Segmentation', *Jonathan Long, Evan Shelhamer and Trevor Darrell*, CVPR, 2015 (paper (http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf)).

  - pascal-fcn32s-dag (../models/pascal-fcn32s-dag.mat) (../models/pascal-fcn32s-dag.svg)
  - pascal-fcn16s-dag (../models/pascal-fcn16s-dag.mat) (../models/pascal-fcn16s-dag.svg)
  - pascal-fcn8s-dag (../models/pascal-fcn8s-dag.mat) (../models/pascal-fcn8s-dag.svg)

  These networks are trained on the PASCAL VOC 2011 training and (in part) validation data, using Berekely's extended annotations (SBD (http://www.cs.berkeley.edu/~bharath2/codes/SBD/download.html)).

  The performance measured on the PASCAL VOC 2011 validation data subset used in the revised version of the paper above (dubbed RV-VOC11):

Model	Test data	Mean IOU	Mean pix. accuracy	Pixel accuracy
FNC-32s	RV-VOC11	59.43	89.12	73.28

FNC-16s	RV-VOC11	62.35	90.02	75.74
FNC-8s	RV-VOC11	62.69	90.33	75.86

- **Torr Vision Group FCN-8s**. This is the FCN-8s subcomponent of the CRF-RNN network from the paper:

  > 'Conditional Random Fields as Recurrent Neural Networks' *Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr*, ICCV 2015 (paper (http://www.robots.ox.ac.uk/~szheng/papers/CRFasRNN.pdf)).

  - pascal-fcn8s-tvg-dag (../models/pascal-fcn8s-tvg-dag.mat) (../models/pascal-fcn8s-tvg-dag.svg)

  These networks are trained on the PASCAL VOC 2011 training and (in part) validation data, using Berekely's extended annotations, as well as Microsoft COCO.

  While the CRF component is missing (it may come later to MatConvNet), this model still outperforms the FCN-8s network above, partially because it is trained with additional data from COCO. In the table below, the RV-VOC12 data is the subset of the PASCAL VOC 12 data as described in the 'Conditional Random Fields' paper:

Model	Tes data	mean IOU	mean pix. accuracy	pixel accuracy
FNC-8s-TVG	RV-VOC12	69.85	92.94	78.80

  TVG implementation note: The model was obtained by first fine-tuning the plain FCN-32s network (without the CRF-RNN part) on COCO data, then building built an FCN-8s network with the learnt weights, and finally training the CRF-RNN network end-to-end using VOC 2012 training data only. The model available here is the FCN-8s part of this network (without CRF-RNN, while trained with 10 iterations CRF-RNN).

ImageNet ILSVRC classification

These modesl are trained to perform classification in the ImageNet ILSVRC challenge data.

- **GoogLeNet** model imported from the Princeton version (http://vision.princeton.edu/pvt/GoogLeNet/) [*DagNN format*].

  > `Going Deeper with Convolutions', *Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich*, CVPR, 2015 (paper (http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf)).

  - imagenet-googlenet-dag (../models/imagenet-googlenet-dag.mat) (../models/imagenet-googlenet-dag.svg)

- **VGG-VD** models from the Very Deep Convolutional Networks for Large-Scale Visual Recognition (http://www.robots.ox.ac.uk/~vgg/research/very_deep/) project.

  > `Very Deep Convolutional Networks for Large-Scale Image Recognition', *Karen Simonyan and Andrew Zisserman,* arXiv technical report, 2014, (paper (http://arxiv.org/abs/1409.1556/)).

  - imagenet-vgg-verydeep-16 (../models/imagenet-vgg-verydeep-16.mat) (../models/imagenet-vgg-verydeep-16.svg)
  - imagenet-vgg-verydeep-19 (../models/imagenet-vgg-verydeep-19.mat) (../models/imagenet-vgg-verydeep-19.svg)

- **VGG-S,M,F** models from the Return of the Devil (http://www.robots.ox.ac.uk/~vgg/research/deep_eval) paper (v1.0.1).

  > `Return of the Devil in the Details: Delving Deep into Convolutional Networks', *Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman,* BMVC 2014 (BibTex and paper (http://www.robots.ox.ac.uk/~vgg/publications/2014/Chatfield14/)).

  - imagenet-vgg-f (../models/imagenet-vgg-f.mat) (../models/imagenet-vgg-f.svg)

- imagenet-vgg-m (../models/imagenet-vgg-m.mat) (../models/imagenet-vgg-m.svg)
- imagenet-vgg-s (../models/imagenet-vgg-s.mat) (../models/imagenet-vgg-s.svg)
- imagenet-vgg-m-2048 (../models/imagenet-vgg-m-2048.mat) (../models/imagenet-vgg-m-2048.svg)
- imagenet-vgg-m-1024 (../models/imagenet-vgg-m-1024.mat) (../models/imagenet-vgg-m-1024.svg)
- imagenet-vgg-m-128 (../models/imagenet-vgg-m-128.mat) (../models/imagenet-vgg-m-128.svg)

The following models have been trained using MatConvNet (beta17) and batch normalization using the code in the `examples/imagenet` directory:

- imagenet-matconvnet-vgg-f (../models/imagenet-matconvnet-vgg-f.mat) (../models/imagenet-matconvnet-vgg-f.svg)
- imagenet-matconvnet-vgg-m (../models/imagenet-matconvnet-vgg-m.mat) (../models/imagenet-matconvnet-vgg-m.svg)
- imagenet-matconvnet-vgg-s (../models/imagenet-matconvnet-vgg-s.mat) (../models/imagenet-matconvnet-vgg-s.svg)
- imagenet-matconvnet-vgg-verydeep-16 (../models/imagenet-matconvnet-vgg-verydeep-16.mat) (../models/imagenet-matconvnet-vgg-verydeep-16.svg)

- **Caffe reference model** obtained here (http://caffe.berkeleyvision.org/getting_pretrained_models.html) (version downloaded on September 2014).

  > Citation: please see the Caffe homepage (http://caffe.berkeleyvision.org).

  - imagenet-caffe-ref (../models/imagenet-caffe-ref.mat) (../models/imagenet-caffe-ref.svg)

- **AlexNet**

  > `ImageNet classification with deep convolutional neural networks', *A. Krizhevsky and I. Sutskever and G. E. Hinton,* NIPS 2012 (BibTex and paper (http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-))

  - imagenet-caffe-alex (../models/imagenet-caffe-alex.mat) (../models/imagenet-caffe-alex.svg)
  - imagenet-matconvnet-alex (../models/imagenet-matconvnet-alex.mat) (../models/imagenet-matconvnet-alex.svg)

  The first model has been imported from Caffe (http://caffe.berkeleyvision.org/getting_pretrained_models.html).

  The MatConvNet model was trained using using MatConvNet (beta17) and batch normalization using the code in the `examples/imagenet` directory.

This is a summary of the performance of these models on the ILSVRC 2012 validation data:

model	top-1 err.	top-5 err.	images/s
matconvnet-alex	41.8	19.2	547.3
matconvnet-vgg-s	37.0	15.8	337.4
matconvnet-vgg-m	36.9	15.5	422.8
matconvnet-vgg-f	41.4	19.1	658.8
matconvnet-vgg-verydeep-16	28.3	9.5	79.1
caffe-ref	42.4	19.6	336.7
caffe-alex	42.6	19.6	332.2
vgg-s	36.7	15.3	321.7
vgg-m	37.3	15.9	404.3
vgg-f	41.1	18.8	661.6
vgg-m-128	40.8	18.4	388.5

vgg-m-1024	37.8	16.1	406.9
vgg-m-2048	37.1	15.8	401.6
vgg-verydeep-19	28.7	9.9	60.9
vgg-verydeep-16	28.5	9.9	73.9
googlenet-dag	32.2	11.6	231.4

Important notes:

- The model trained using MatConvNet are slightly better than the original, probably due to the use of batch normalization during training.
- Error rates are computed on a single centre-crop and are therefore higher than what reported in some publications, where multiple evaluations per image are combined.
- The **evaluation speed** was measured on a 12-cores machine using a single NVIDIA Titan Black GPU, MATLAB R2015a, and CuDNN v4; performance varies hugely depending on the network but also on how the data was preprocessed; for example, `caffe-ref` and `caffe-alex` should be as fast as `matconvnet-alex`, but they are not since images were pre-processed in such a way that MATLAB had to call `imresize` for each input image for the Caffe models.
- The GoogLeNet model performance is a little lower than expected (the model should be on par or a little better than VGG-VD). This network was imported from the Princeton version of GoogLeNet, not by the Google team, so the difference might be due to parameter setting during training. On the positive side, GoogLeNet is much smaller (in terms of parameters) and faster than VGG-VD.

File checksums

The following table summarizes the MD5 checksums for the model files.

MD5	File name
77ba5337725eb77362e9f318898af494	imagenet-caffe-alex.mat
7001959cb66a3d62a86d52efff42f168	imagenet-caffe-ref.mat
e646ea925dee772e34794f01ebbe1bd8	imagenet-googlenet-dag.mat
d79a53b79b62aee8a6c48755c29448fa	imagenet-matconvnet-alex.mat
5c00773832303a2a9656afec097fb1c1	imagenet-matconvnet-vgg-f.mat
2ccbc5c4d77a56fbfc288ca810d12206	imagenet-matconvnet-vgg-m.mat
79c64eedb1fa49668997342b02dea863	imagenet-matconvnet-vgg-s.mat
7502ff082bf53ce9cc67110399bbdf53	imagenet-matconvnet-vgg-verydeep-16.mat
4775484a70e8bac3e9521aed59f31dfc	imagenet-vgg-f.mat
0545aea6fc5173b2806784f2a8bd3333	imagenet-vgg-m-1024.mat
80ec27ef99e2faefb9c837216c9ea0e4	imagenet-vgg-m-128.mat
620aca6468345e4a791a0396c6a51ce1	imagenet-vgg-m-2048.mat
b4e8616c0ab66b1fda72854226f82d02	imagenet-vgg-m.mat
2f83043a38e71e9dd9b1c5c0cb3ef6f9	imagenet-vgg-s.mat
5a68244cf55c66fea59e23ee63cf56ef	imagenet-vgg-verydeep-16.mat
b9b4a9eb1c2fb3b50e1ec1aca6f22342	imagenet-vgg-verydeep-19.mat

210308543d1a510239ed85feb9b2e885	pascal-fcn16s-dag.mat
2dea567374085a63ad6e83b7b08fa482	pascal-fcn32s-dag.mat
8db84f60ba7d519de15cdf9d2c9a40e1	pascal-fcn8s-dag.mat
ad374f3aa98208847489e5c7bfd8e013	pascal-fcn8s-tvg-dag.mat
5069daad93d2937554325e30388463ca	vgg-face.mat

Older file versions

Older models for MatConvNet beta16 are available here (../models/beta16). They should be numerically equivalent, but in beta17 the format has changed slightly for SimpleNN models. Older models can also be updated using the `vl_simplenn_tidy` function.