Pretrained Cascade Image Classification

Updated: July 23, 2015

Creates a pretrained image classification model for frontal faces using the OpenCV Library

Category: OpenCV Library Modules (https://msdn.microsoft.com/en-us/library/azure/dn905946.aspx)

Module Overview

You can use the **Pretrained Cascade Image Classification** module to detect faces in images. This particular model has already been trained, and classifies images as either containing a human face in frontal view, or not.

The model is based on the OpenCV (http://opencv.org/) library. The OpenCV Library provides a list of predefined models, each optimized to detect a particular type of object. This particular classification model is optimized for facial detection, and uses the Viola-Jones object detection algorithm.

Warning

Currently only one OpenCV image classification model is provided, but additional pre-trained models might be available in later releases.

Note

Models based on **Pretrained Cascade Image Classification** cannot be retrained. These models are pre-trained, and the format is incompatible with that of other modules such as Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx) and Cross-Validate Model (https://msdn.microsoft.com/en-us/library/azure/dn905852.aspx).

How to Use Pre-Trained Image Classification

1. Add the **Pretrained Cascade Image Classification** module to your experiment.

2. Set parameters that control how images are analyzed, using the properties in **Pretrained Cascade Image Classification**.

See the Options section for details.

3. The model is optimized for a specific image type and has already been trained using a large dataset. Therefore, all you need to do is provide a set of images as a dataset, and the module generates a score that indicates whether each image contains the target image type.

   To do this, add the Image Reader (https://msdn.microsoft.com/en-us/library/azure/dn905877.aspx) module to your experiment.

Tip
Be sure to read the help for Image Reader (https://msdn.microsoft.com/en-us/library/azure/dn905877.aspx) to ensure that the images you use meet the requirements, and for help in configuring access to the images.

4. Add the Score Model (https://msdn.microsoft.com/en-us/library/azure/dn905995.aspx) module to your experiment.

   Connect the pre-trained image classifier to the left input of the Score Model (https://msdn.microsoft.com/en-us/library/azure/dn905995.aspx) module, and connect your dataset of images to the right input of Score Model (https://msdn.microsoft.com/en-us/library/azure/dn905995.aspx).

5. Run the experiment.

6. The output of Score Model (https://msdn.microsoft.com/en-us/library/azure/dn905995.aspx) includes the image name, the scored label, and the probability score for the label (either 0 or 1). The classifier outputs a "1" if the image is likely to show the object (a face), and "0" otherwise. For example:

Image name	Scored Labels	Scored Probabilities
MAN001.png	TRUE	1
TABLE001.PNG	FALSE	0
CHAIR001.PNG	FALSE	0

Tip
The output also contains the RGB values for all color channels in the dataset, so we recommend that in your experiment you use Project Columns (https://msdn.microsoft.com/en-us/library/azure/dn905883.aspx) to output just the result columns.

Options

Pre-trained classifier

Choose a pre-trained classifier from the list.

Currently, only one classifier (**Frontal face**) is available, but more might be available in future.

Scale factor

Type a value that specifies how much the image size is reduced at each image scale.

In the OpenCV Library, the classifier is designed so that it can be easily “resized” in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales.

We recommend that you try different scaling factors to see which provides the best image classification results.

Minimum number of neighbors

Type the minimum number of overlapping rectangles that are required to detect that a face is included in a region.

In the OpenCV library, the classifier detects objects of different sizes in the input image. The detected objects are returned as a list of rectangles. The *neighbors* parameter controls how many possible matches are required to qualify as a detected face or feature. Thus, increasing this value tends to increase precision at the cost of coverage.

For examples of how neighbors are calculated, see this article in the OpenCv Library documentation: Eigenfaces in OpenCV (http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html?highlight=neighbor)

Minimum height

Type the pixel height of the smallest image. If you specify a value for this property, images smaller than this are ignored.

In general, images should be the same size.

Minimum width

Type the pixel width of the smallest image. If you specify a value for this property, images smaller than this are ignored.

Maximum height

Type the pixel width of the largest image. If you specify a value for this property, images larger than this are ignored.

Maximum width

Type the pixel width of the largest image. If you specify a value for this property, images larger than this are ignored.

Technical Notes

The facial recognition model provided by this module is based on the Viola-Jones face detection algorithm. For more information, see these resources:

- This video explains the basic concepts of facial recognition, including a definition of *Haar features* and how they are used in facial detection: Facial Detection - Part 1 (https://youtu.be/sWTvK72-SPU)

- This Wikipedia article describes the method used for the classifier, based on the paper by Navneet Dalal and Bill Triggs: Histogram of oriented gradients (https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients)

- For the documentation of the face recognition algorithm provided in the OpenCV library, see Cascade Classifier (http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html?highlight=detectmultiscale).

  This module does not output the full collection of information produced by the OpenCV library. In particular, this module only outputs the prediction of whether a face is present or not, and does not include the coordinates of the face or any other information. If you need this additional information, consider using other open source libraries, such as the Project Oxford Face API.

Module Parameters

Name	Range	Type	Default	Description
Pre-trained classifier	List	PretrainedClassifier	Frontal face	Pretrained classifier from standard OpenCV distribution.
Scale factor	>=1.0000000000000002	Float	1.1	Parameter that specifies how much the image size is reduced at each image scale.
Minimum number of neighbors	>=0	Integer	3	Parameter that specifies how many neighbors each candidate

				rectangle should have to retain it.
Minimum height	>=1	Integer	100	Minimum possible object height (in pixels). Objects smaller than this are ignored. The parameter is optional.
Minimum width	>=1	Integer	100	Minimum possible object width (in pixels). Objects smaller than this are ignored. The parameter is optional.
Maximum height	>=1	Integer	200	Maximum possible object height (in pixels). Objects larger than this are ignored. The parameter is optional.
Maximum width	>=1	Integer	200	Maximum possible object width (in pixels). Objects larger than this are ignored. The parameter is optional.

Output

Name	Type	Description

Trained model	ILearner interface (https://msdn.microsoft.com/en-us/library/azure/dn905938.aspx)	Trained binary classification model

Exceptions

For a list of all error messages, see Machine Learning Module Error Codes (https://msdn.microsoft.com/en-us/library/azure/dn905910.aspx).

Exception	**Description**
Error 0005 (https://msdn.microsoft.com/en-us/library/azure/dn906042.aspx)	Exception occurs if parameter is less than a specific value.

See Also

Image Reader (https://msdn.microsoft.com/en-us/library/azure/dn905877.aspx)