The most recent OpenCV

From Beginner To Advanced

Deep Learning for Computer Vision

TensorFlow, Caffe, OpenCV 4, Visual Studio 2017 (C++), Windows 10 x64 bit

https://www.youtube.com/tiziran

You can view and comment presentation file

https://docs.google.com/presentation/d/12OqgsInMveeGbJYPnNOG8jwMQbp76DjqK4_NZtJE7bA/edit?usp=sharing

Download Source Code from GitHub

https://github.com/pirahansiah/opencv4

Outline

- Compile full OpenCV 4 (main and contribute) from source
  - https://www.youtube.com/watch?v=VK70YdaMD44
- Setup and configure Visual Studio 2017 project (C++) for using OpenCV 4 library
  - https://www.youtube.com/watch?v=J0Exttz4_m4
- Using Caffe model in OpenCV 4
  - https://www.youtube.com/watch?v=lj7AxKDoP9I
- Using TensorFlow model in OpenCV 4
  - https://www.youtube.com/watch?v=5JqX0CbxtNk
- OpenCV4

OpenCV 4

August 2018

C++11

Deep learning

Some updates:

- Hardware-accelerated Video I/O module
- HighGUI module
- Graph API module
- Point Cloud module
- Tracking, Calibration, and Stereo Modules

Compile OpenCV 4

Build OpenCV 4 with Visual Studio 2017 (C++) on Windows 10 x64 bit

OpenCV version 4 (August 2018) https://www.youtube.com/tiziran

You need Cmake & Visual studio 2017 & Windows 10 (64 bit)

If you want to use GPU for Deep Learning framework you need NVidia GPU like GTX 1080 and also need to install CUDA from NVidia website

Compile OpenCV 4 - part 1

Download

https://github.com/opencv/opencv

https://github.com/opencv/opencv_contrib

Create build folder

Using Cmake to generate visual studio project

Use the path of … \opencv_contrib-master\modules

Open “OpenCV.sln” project

Compile OpenCV 4 - part 2

Build

- Debug
- Release
- Install

Set the environment variable

- Add “C:\opencv4\build\install\x64\vc15\bin” to windows path

Setup and configure your project (debug mode (64 bit))

Setup visual studio project to use OpenCV 4 (you only need install folder inside the build folder of compiled OpenCV 4)
For the debug mode (64 bit)

- Add the include folder of OpenCV 4 to include path of your visual studio project
  - C:\opencv4\build\install\include
- Add the lib folder of OpenCV 4 to lib path of your visual studio project
  - C:\opencv4\build\install\x64\vc15\lib

Put list of lib files for the debug mode (64 bit) (linker->input->additional dependencies)

opencv_aruco400d.lib;opencv_bgsegm400d.lib;opencv_bioinspired400d.lib;opencv_calib3d400d.lib;opencv_ccalib400d.lib;opencv_core400d.lib;opencv_datasets400d.lib;opencv_dnn400d.lib;opencv_dnn_objdetect400d.lib;opencv_dpm400d.lib;opencv_face400d.lib;opencv_features2d400d.lib;opencv_flann400d.lib;opencv_fuzzy400d.lib;opencv_hfs400d.lib;opencv_highgui400d.lib;opencv_imgcodecs400d.lib;opencv_imgproc400d.lib;opencv_img_hash400d.lib;opencv_line_descriptor400d.lib;opencv_ml400d.lib;opencv_objdetect400d.lib;opencv_optflow400d.lib;opencv_phase_unwrapping400d.lib;opencv_photo400d.lib;opencv_plot400d.lib;opencv_reg400d.lib;opencv_rgbd400d.lib;opencv_saliency400d.lib;opencv_shape400d.lib;opencv_stereo400d.lib;opencv_stitching400d.lib;opencv_structured_light400d.lib;opencv_superres400d.lib;opencv_surface_matching400d.lib;opencv_text400d.lib;opencv_tracking400d.lib;opencv_video400d.lib;opencv_videoio400d.lib;opencv_videostab400d.lib;opencv_xfeatures2d400d.lib;opencv_ximgproc400d.lib;opencv_xobjdetect400d.lib;opencv_xphoto400d.lib

Setup and configure your project (release mode (64 bit))

Setup visual studio project to use OpenCV 4 (you only need install folder inside the build folder of compiled OpenCV 4)
For the released mode (64 bit)

- Add the include folder of OpenCV 4 to include path of your visual studio project
  - C:\opencv4\build\install\include
- Add the lib folder of OpenCV 4 to lib path of your visual studio project
  - C:\opencv4\build\install\x64\vc15\lib

Put list of lib files for the released mode (64 bit) (linker->input->additional dependencies)

opencv_aruco400.lib;opencv_bgsegm400.lib;opencv_bioinspired400.lib;opencv_calib3d400.lib;opencv_ccalib400.lib;opencv_core400.lib;opencv_datasets400.lib;opencv_dnn400.lib;opencv_dnn_objdetect400.lib;opencv_dpm400.lib;opencv_face400.lib;opencv_features2d400.lib;opencv_flann400.lib;opencv_fuzzy400.lib;opencv_hfs400.lib;opencv_highgui400.lib;opencv_imgcodecs400.lib;opencv_imgproc400.lib;opencv_img_hash400.lib;opencv_line_descriptor400.lib;opencv_ml400.lib;opencv_objdetect400.lib;opencv_optflow400.lib;opencv_phase_unwrapping400.lib;opencv_photo400.lib;opencv_plot400.lib;opencv_reg400.lib;opencv_rgbd400.lib;opencv_saliency400.lib;opencv_shape400.lib;opencv_stereo400.lib;opencv_stitching400.lib;opencv_structured_light400.lib;opencv_superres400.lib;opencv_surface_matching400.lib;opencv_text400.lib;opencv_tracking400.lib;opencv_video400.lib;opencv_videoio400.lib;opencv_videostab400.lib;opencv_xfeatures2d400.lib;opencv_ximgproc400.lib;opencv_xobjdetect400.lib;opencv_xphoto400.lib

Simple program to test

```
#include "stdafx.h"
#include <opencv2/core.hpp>
#include <opencv2/highgui.hpp>
int main()
{
    cv::String filePath = "C:/opencv4/OpenCV4VS2017/OpenCV4VS2017/opencv-logo.png";
    cv::Mat src = cv::imread(filePath);
    cv::imshow("test", src);
    cv::waitKey(10000);
        return 0;
}
```

Using Caffe model in OpenCV 4 - part 1

Download

1. Synset_words.txt
   https://raw.githubusercontent.com/HoldenCaulfieldRye/caffe/master/data/ilsvrc12/synset_words.txt
2. Bvlc_googlenet.caffemodel
   http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel
3. Bvlc_googlenet.prototxt
   https://raw.githubusercontent.com/opencv/opencv_extra/master/testdata/dnn/bvlc_googlenet.prototxt

Using caffe model in OpenCV 4 - part 2

```
static void getMaxClass(const cv::Mat &probBlob, int *classId, double *classProb)
{
    cv::Mat probMat = probBlob.reshape(1, 1); //reshape the blob to 1x1000 matrix
    cv::Point classNumber;
    cv::minMaxLoc(probMat, NULL, classProb, NULL, &classNumber);
    *classId = classNumber.x;
}
static std::vector<cv::String> readClassNames(const char *filename = "C:/opencv4/OpenCV4VS2017/DeepLearningFiles/synset_words.txt")
{
    std::vector<cv::String> classNames;
    std::ifstream fp(filename);
    if (!fp.is_open())
    {
            std::cerr << "File with classes labels not found: " << filename << std::endl;
            exit(-1);
    }
    std::string name;
    while (!fp.eof())
    {
            std::getline(fp, name);
            if (name.length())
                    classNames.push_back(name.substr(name.find(' ') + 1));
    }
    fp.close();
    return classNames;
}
```

Using caffe model in OpenCV 4 - part 3

```
        float scale = 1.0;
        cv::Scalar meanG = cv::Scalar(104, 117, 123);
        bool swapRB = true;
        int inpWidth = 224;
        int inpHeight = 224;
        cv::String model = path + "bvlc_googlenet.caffemodel";
        cv::String config = path + "bvlc_googlenet.prototxt";
        cv::String nameClass = path + "synset_words.txt";
        cv::String nameImage = path + "space_shuttle.jpg";
        cv::String framework;
        int backendId = 0;
        int targetId = 0;
        cv::dnn::Net net;
        cv::Mat blob;
        cv::Mat img = imread(nameImage);
        net = cv::dnn::readNet(model, config);
        net.setPreferableBackend(backendId);
        net.setPreferableTarget(targetId);
        cv::dnn::blobFromImage(img, blob, scale, cv::Size(inpWidth, inpHeight), meanG, swapRB, false);
        net.setInput(blob);
        cv::Mat prob = net.forward();
```

```
      cv::Point classIdPoint;
       double confidence;
       minMaxLoc(prob.reshape(1, 1), 0, &confidence, 0, &classIdPoint);
       int classId = classIdPoint.x;
       std::vector<double> layersTimes;
       double freq = cv::getTickFrequency() / 1000;
       double t = net.getPerfProfile(layersTimes) / freq;
       std::string label = cv::format("Inference time: %.2f ms", t);
       putText(img, label, cv::Point(0, 15), cv::FONT_HERSHEY_SIMPLEX, 0.5, cv::Scalar(0, 255, 0));
       cv::imshow("https://www.youtube.com/tiziran", img);
       std::vector<cv::String> classNames = readClassNames();
       std::cout << "Best class: #" << classId << " '" << classNames.at(classId) << "'" << std::endl;
       std::cout << "Probability: " << confidence * 100 << "%" << std::endl;
       cv::waitKey(10000);
```

Using TensorFlow model in OpenCV 4 - part 1

TensorFlow API only for OpenCV 4 **Download the inception model based on TensorFlow** ● **https://storage.googleapis.com/download.tensorflow.org/models/inception5h.zip**	

Using TensorFlow model in OpenCV 4 - part 2

```
float scale = 1.0;
    cv::Scalar meanG = cv::Scalar(104, 117, 123);
    bool swapRB = true;
    int inpWidth = 224;
    int inpHeight = 224;
    cv::String model = path + "tensorflow_inception_graph.pb";
    cv::String nameImage = path + "space_shuttle.jpg";
    cv::String framework;
    int backendId = 0;
    int targetId = 0;
    cv::dnn::Net net;
    cv::Mat blob;
    cv::Mat img = imread(nameImage);
    net = cv::dnn::readNetFromTensorflow(model);
    net.setPreferableBackend(backendId);
    net.setPreferableTarget(targetId);
    cv::dnn::blobFromImage(img, blob, scale, cv::Size(inpWidth, inpHeight), meanG, swapRB, false);
    net.setInput(blob);
```

```
cv::Mat prob = net.forward();
cv::Point classIdPoint;
    double confidence;
    minMaxLoc(prob.reshape(1, 1), 0, &confidence, 0, &classIdPoint);
    int classId = classIdPoint.x;
    std::vector<double> layersTimes;
    double freq = cv::getTickFrequency() / 1000;
    double t = net.getPerfProfile(layersTimes) / freq;
    std::string label = cv::format("Inference time: %.2f ms", t);
    putText(img, label, cv::Point(0, 15), cv::FONT_HERSHEY_SIMPLEX, 0.5, cv::Scalar(0, 255, 0));
    cv::imshow("https://www.youtube.com/tiziran", img);
    std::vector<cv::String> classNames = readClassNamesTensorFlow();
    std::cout << "Best class: #" << classId << " "" << classNames.at(classId) << """ << std::endl;
    std::cout << "Probability: " << confidence * 100 << "%" << std::endl;
    cv::waitKey(10000);
```

Using

-

Using

-

Using

-

Using

-

Previous Videos, Courses, Classes, Training, Tutorials

- Compile full OpenCV 4 (main and contribute) from source
  - https://www.youtube.com/watch?v=VK70YdaMD44
- Setup and configure Visual Studio 2017 project (C++) for using OpenCV 4 library
  - https://www.youtube.com/watch?v=J0Exttz4_m4

- Using Caffe model in OpenCV 4
  - https://www.youtube.com/watch?v=lj7AxKDoP9I
- Using TensorFlow model in OpenCV 4
  - https://www.youtube.com/watch?v=5JqX0CbxtNk
  -

You can view and comment this presentation file

https://docs.google.com/presentation/d/12OqgsInMveeGbJYPnNOG8jwMQbp76DjqK4_NZtJE7bA/edit?usp=sharing

Please subscribe our youtube channel

https://www.youtube.com/tiziran