

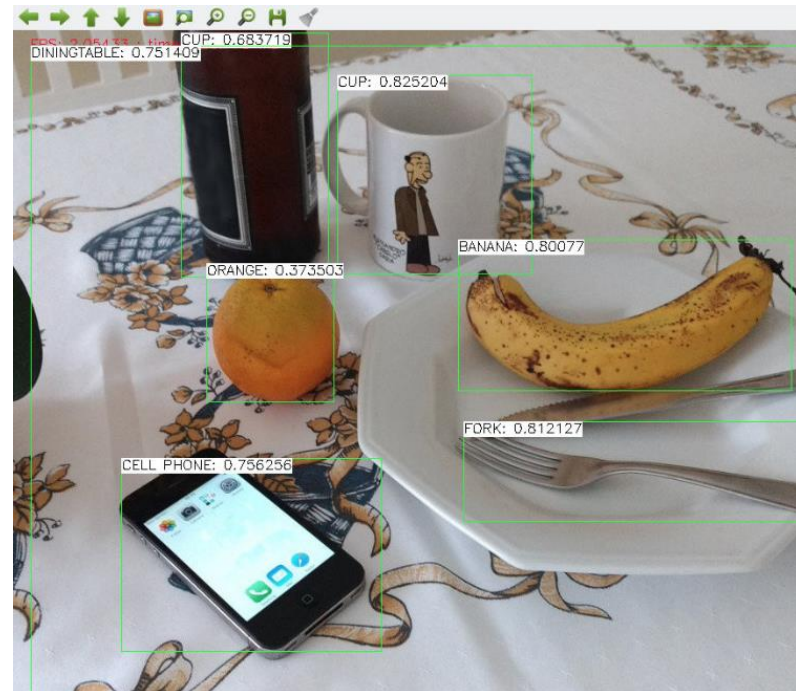
Object Detection using Deep Learning

Sung Soo Hwang

Deep learning in openCV

- From ver. 3.3, you can use deep learning in openCV!
 - openCV support many deep learning framework such as Caffe, TensorFlow, Darknet, and Torch/PyTorch
 - You can use pre-trained deep learning model and use it in C++, Python

Label: beagle, 81.14%



- How to use deep learning in openCV
 1. load deep learning model
 2. process an input image to a blob suitable for deep learning model
 3. obtain classification result by propagating the input blob

- Load deep learning model

```
String modelConfiguration = "yolov2.cfg";  
String modelBinary = "yolov2.weights";  
  
Net net = readNetFromDarknet(modelConfiguration, modelBinary);
```

\$ readNetFromDarknet() [1/3]

```
Net cv::dnn::readNetFromDarknet ( const String & cfgFile,  
                                const String & darknetModel = String()  
                                )
```

Reads a network model stored in [Darknet](#) model files.

Parameters

cfgFile path to the .cfg file with text description of the network architecture.
darknetModel path to the .weights file with learned network.

Returns

Network object that ready to do forward, throw an exception in failure cases.
Net object.

- Process an input image to a blob suitable for deep learning model

```
//Convert Mat to batch of images  
Mat inputBlob = blobFromImage(frame, 1 / 255.F, Size(416, 416), Scalar(), true, false);
```

§ blobFromImage() [1/2]

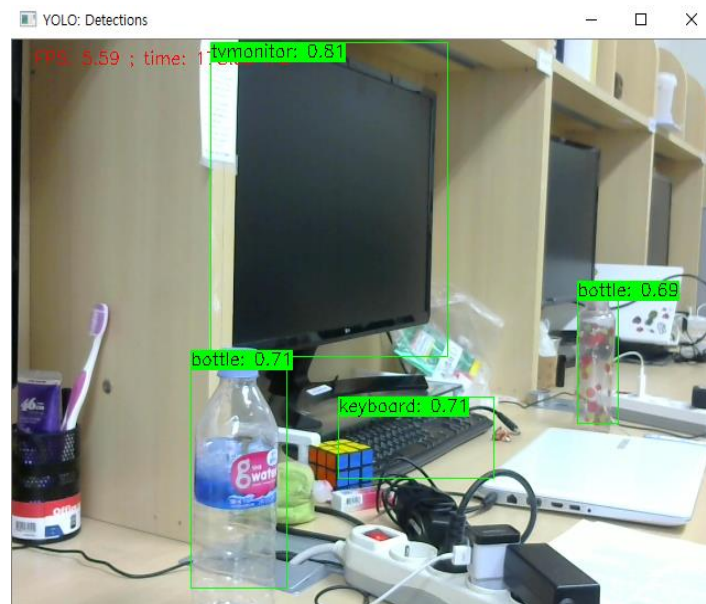
```
Mat cv::dnn::blobFromImage ( InputArray  image,  
                             double      scalefactor =  
                             1.0,  
                             const Size & size = Size(),  
                             mean =  
                             const Scalar & Scalar(),  
                             bool        swapRB = true,  
                             bool        crop = true,  
                             int         ddepth = CV_32F  
                             )
```

- Obtain classification result by propagating the input blob

```
net.setInput(inputBlob, "data");
```

```
Mat detectionMat = net.forward("detection_out");
```

- While GoogleNet classifies an image, yolo detects object in an image



■ Example code

```
int main(int argc, char** argv)
{
    String modelConfiguration = "deep/yolov2.cfg";
    String modelBinary = "deep/yolov2.weights";

    Net net = readNetFromDarknet(modelConfiguration, modelBinary);

    VideoCapture cap("downtown_road.wmv");

    vector<String> classNamesVec;
    ifstream classNamesFile("deep/coco.names");

    if (classNamesFile.is_open()) {
        string className = "";
        while (std::getline(classNamesFile, className)) classNamesVec.push_back(className);
    }

    while (1)
    {
        Mat frame;
        cap >> frame; // get a new frame from camera/video or read image
        if (frame.empty()) {
            waitKey();
            break;
        }

        if (frame.channels() == 4) cvtColor(frame, frame, COLOR_BGRA2BGR);
    }
}
```

■ Example code

```
//Convert Mat to batch of images
Mat inputBlob = blobFromImage(frame, 1 / 255.F, Size(416, 416), Scalar(), true, false);
net.setInput(inputBlob, "data"); //set the network input
Mat detectionMat = net.forward("detection_out"); //compute output

float confidenceThreshold = 0.24; //by default

for (int i = 0; i < detectionMat.rows; i++) {
    const int probability_index = 5;
    const int probability_size = detectionMat.cols - probability_index;
    float *prob_array_ptr = &detectionMat.at<float>(i, probability_index);
    size_t objectClass = max_element(prob_array_ptr, prob_array_ptr + probability_size) -
prob_array_ptr;
    //특정한 물체가 detection된 확률
    float confidence = detectionMat.at<float>(i, (int)objectClass + probability_index);

    //For drawing
    if (confidence > confidenceThreshold) {
        float x_center = detectionMat.at<float>(i, 0) * frame.cols;
        float y_center = detectionMat.at<float>(i, 1) * frame.rows;
        float width = detectionMat.at<float>(i, 2) * frame.cols;
        float height = detectionMat.at<float>(i, 3) * frame.rows;
```


■ Example code

```
Point p1(cvRound(x_center - width / 2), cvRound(y_center - height / 2));
Point p2(cvRound(x_center + width / 2), cvRound(y_center + height / 2));
Rect object(p1, p2);
Scalar object_roi_color(0, 255, 0);

rectangle(frame, object, object_roi_color);
String className = objectClass < classNamesVec.size() ?
classNamesVec[objectClass] : cv::format("unknown(%d)", objectClass);
String label = format("%s: %.2f", className.c_str(), confidence);
int baseLine = 0;

Size labelSize = getTextSize(label, FONT_HERSHEY_SIMPLEX, 0.5, 1, &baseLine);

rectangle(frame, Rect(p1, Size(labelSize.width, labelSize.height + baseLine)),
object_roi_color, FILLED);
putText(frame, label, p1 + Point(0, labelSize.height), FONT_HERSHEY_SIMPLEX,
0.5, Scalar(0, 0, 0));
    }
}
imshow("YOLO: Detections", frame);
if (waitKey(1) >= 0) break;
}
return 0;
}
```

