

首先将 `detection.cpp`, `detection.h` 两个文件复制到工程的目录下，这些文件是基于 `darknet` 官方源码修改而成，如图 1 所示：

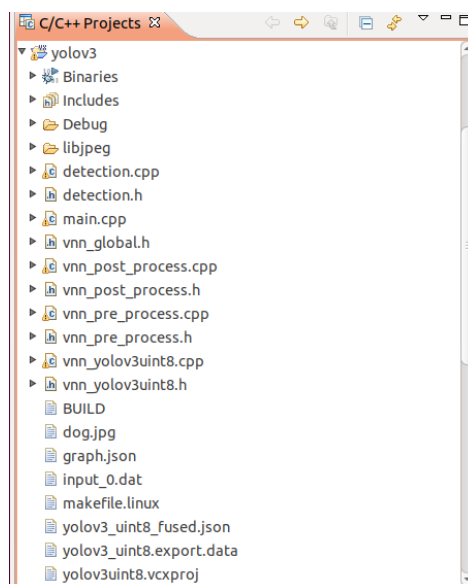


图 1 工程目录

打开 `vnn_post_process.cpp`，增加包含头文件 `#include "detection.h"`。

最后在 `vnn_PostProcessYolov3Uint8` 函数中注释掉对 `show_top5` 函数的调用，取而代之的是刚刚添加的 `show_result` 函数，如图 2 所示：

```
vsi_status vnn_PostProcessYolov3Uint8(vsi_nn_graph_t *graph)
{
    vsi_status status = VSI_FAILURE;

    /* Show the top5 result */
    status = show_result(graph, vsi_nn_GetTensor(graph, graph->output.tensors[0]));
    //status = show_top5(graph, vsi_nn_GetTensor(graph, graph->output.tensors[0]));
    TEST_CHECK_STATUS(status, final);

    /* Save all output tensor data to txt file */
    save_output_data(graph);

final:
    return VSI_SUCCESS;
}
```

图 2 `vnn_post_process.cpp` 添加内容

值得注意的是，如果用户不需要将 `tensor` 文件保存下来，可以将 `save_output_data` 注释掉。

打开 `detection.cpp`，根据网络的不同修改 `Variable definitions` 部分的参数，目前支持 `yolov3`, `yolov4`, `yolov5`, `yolov3-tiny`, `yolov4-tiny`，如图 3 所示：

```

/*-----
                        Variable definitions
-----*/

//#define YOLOV3_TINY_SUPPORT
//#define YOLOV4_TINY_SUPPORT
#define YOLOV3_SUPPORT
//#define YOLOV4_SUPPORT
//#define YOLOV5_SUPPORT

char *image_name=(char*)"dog.jpg";
char *class_names[] = {"person","bicycle","car","motorbike","aeroplane","bus","train","truck","boat","traffic light",
};
float thresh = 0.2;    //thresh
float nms = 0.45;     //non-maximum suppression

```

图 3 后处理参数

用户可以根据自己的网络类型在提供的五种定义中选择一个，并将处理的图片名称以及目标名称替换为自己的，同时阈值和 **nms** 亦可在此定义。