Jetson TX2 기반 YOLO 응용 과정 - TX2에 YOLO 설치

2018. 12

모두의연구소 RL4RWS

양재 R&CD 혁신허브
Yangjae Innovation Hub

모두의연구소

목 차

01 TX2 보드에 OPENCV 셋업

02 TX2 보드에 Tensorflow 셋업

03 TX2 보드에 YOLO V3 셋업

04 물체추적 기능

05 YOLO V3 객체명과 커맨드

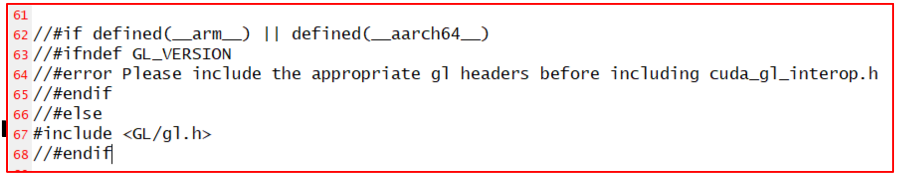
OPENCV 설치

OPENCV 3.4.0 버전을 설치하기 위해서 터미널에서 관련된 라이브러리를 설치한다.

```
$ sudo apt-get purge libopencv*
$ sudo apt-get purge python-numpy
$ sudo apt autoremove -y
$ sudo apt-add-repository universe
$ sudo apt-get update -y
$ sudo apt-get dist-upgrade -y
$ sudo apt-get install -y --only-upgrade g++-5 cpp-5 gcc-5

$ sudo apt-get install -y build-essential make cmake cmake-curses-gui g++ libavformat-dev libavutil-dev libswscale-dev libv4l-dev
libeigen3-dev libglew-dev libgtk2.0-dev
$ sudo apt-get install -y libdc1394-22-dev libxine2-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
$ sudo apt-get install -y libjpeg8-dev libjpeg-turbo8-dev libtiff5-dev libjasper-dev libpng12-dev libavcodec-dev
$ sudo apt-get install -y libxvidcore-dev libx264-dev libgtk-3-dev libatlas-base-dev gfortran
$ sudo apt-get install -y qt5-default
$ sudo apt-get install -y python3-dev python3-pip python3-tk
$ sudo pip3 install numpy
#$ sudo pip3 install matplotlib
#$ sudo gedit /usr/local/lib/python3.5/dist-packages/matplotlib/mpl-data/matplotlibrc

$ sudo apt-get install -y python-dev python-pip python-tk
$ sudo pip2 install numpy
#$ sudo pip2 install matplotlib
#$ sudo gedit /usr/local/lib/python2.7/dist-packages/matplotlib/mpl-data/
```



```

//#if defined(__arm__) || defined(__aarch64__)
//#ifndef GL_VERSION
//#error Please include the appropriate gl headers before including cuda_gl_interop.h
//#endif
//#else
#include <GL/gl.h>
//#endif
```

```
$ sudo patch -N /usr/local/cuda/include/cuda_gl_interop.h ~/OpenGLHeader.patch ➔ 한번만 실행하기
$ cd /usr/lib/aarch64-linux-gnu/
$ sudo ln -sf tegra/libGL.so libGL.so
```

OPENCV 설치

OPENCV 3.4.0 버전을 다운로드, 컴파일 및 설치하기

```
### Download opencv-3.4.0 source code

$ mkdir -p ~/opencv
$ cd ~/opencv
$ wget https://github.com/opencv/opencv/archive/3.4.0.zip \-O opencv-3.4.0.zip
$ unzip opencv-3.4.0.zip

### Build opencv (CUDA_ARCH_BIN="6.2" for TX2, or "5.3" for TX1)
$ cd ~/opencv/opencv-3.4.0
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local \-D WITH_CUDA=ON -D CUDA_ARCH_BIN="6.2" -
D CUDA_ARCH_PTX="" \-D WITH_CUBLAS=ON -D ENABLE_FAST_MATH=ON -D CUDA_FAST_MATH=ON \-D ENABLE_NEON=ON -D
WITH_LIBV4L=ON -D BUILD_TESTS=OFF \-D BUILD_PERF_TESTS=OFF -D BUILD_EXAMPLES=OFF \-D
ENABLE_PRECOMPILED_HEADERS=OFF \-D WITH_QT=ON -D WITH_OPENGL=ON ..

$ make -j4
$ sudo make install
```

OPENCV 설치

설치가 정상적으로 완료되면 "done" 메시지가 보인다.

```
--    Libraries:             /usr/lib/aarch64-linux-gnu/libpython2.7.so (ver 2.7.12)
--    numpy:                 /usr/local/lib/python2.7/dist-packages/numpy/core/include (ver 1.14.0)
--    packages path:             lib/python2.7/dist-packages
---  Python 3:
--    Interpreter:            /usr/bin/python3 (ver 3.5.2)
--    Libraries:             /usr/lib/aarch64-linux-gnu/libpython3.5m.so (ver 3.5.2)
--    numpy:                 /usr/local/lib/python3.5/dist-packages/numpy/core/include (ver 1.14.0)
--    packages path:             lib/python3.5/dist-packages
---  Python (for build):         /usr/bin/python2.7
---  Java: --     ant:                  NO
--    JNI:                NO
--    Java wrappers:            NO
--    Java tests:            NO ---   Matlab:              NO –
--  Install to:              /usr/local
-- -----------------------------------------------------------------
-- Configuring done
-- Generating done
-- Build files have been written to: /home/nvidia/src/opencv-3.4.0/build
```

OPENCV 설치

동작에 문제가 없는지 아래 커맨드를 사용해서 확인한다.

opencv 버전 확인하기

```
$pkg-config --modversion opencv
```

```
nvidia@tegra-ubuntu:~$ pkg-config --modversion opencv
3.4.0
```

opencv 설치 후 적용하기

```
$sudo ldconfig
```

opencv 동작 확인하기

```
$ ls /usr/local/lib/python3.5/dist-packages/cv2*
$ ls /usr/local/lib/python2.7/dist-packages/cv2*
$ python3 -c 'import cv2; print(cv2.__version__)'
$ python2 -c 'import cv2; print(cv2.__version__)'
```

```
nvidia@tegra-ubuntu:~/opencv/opencv-3.4.0/build$ ls /usr/local/lib/python3.5/dist-packages/cv2*
/usr/local/lib/python3.5/dist-packages/cv2.cpython-35m-aarch64-linux-gnu.so
nvidia@tegra-ubuntu:~/opencv/opencv-3.4.0/build$ ls /usr/local/lib/python2.7/dist-packages/cv2*
/usr/local/lib/python2.7/dist-packages/cv2.so
nvidia@tegra-ubuntu:~/opencv/opencv-3.4.0/build$ python3 -c 'import cv2; print(cv2.__version__)'
3.4.0
nvidia@tegra-ubuntu:~/opencv/opencv-3.4.0/build$ python2 -c 'import cv2; print(cv2.__version__)'
3.4.0
```

pip 설치

```
$ sudo apt-get install python-pip
$ sudo apt-get install python3-pip
```

pip 업그레이드

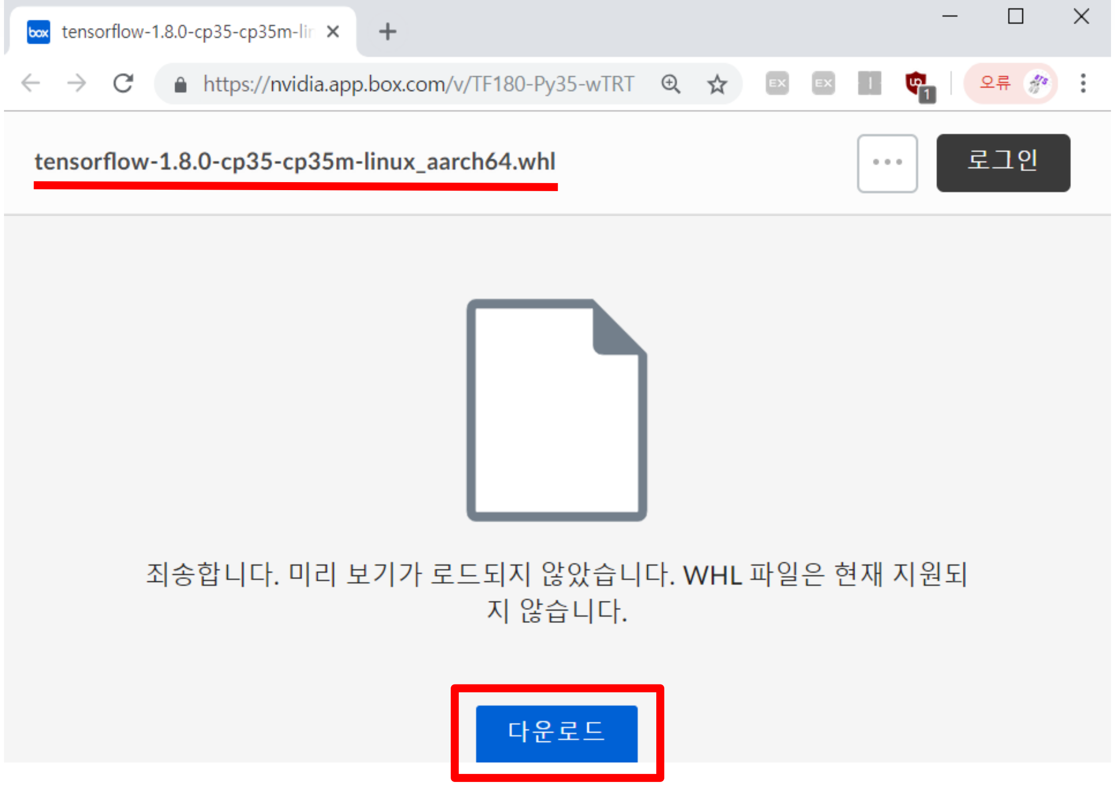
```
$ sudo -H pip3 install --upgrade pip
```

```
You are using pip version 8.1.1, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
nvidia@tegra-ubuntu:~/Downloads$ sudo -H pip3 install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/c2/d7/90f34cb0d83a6c5631cf71
    100% |████████████████████████████████| 1.3MB 370kB/s
Installing collected packages: pip
  Found existing installation: pip 8.1.1
    Not uninstalling pip at /usr/lib/python3/dist-packages, outside environment /usr
Successfully installed pip-18.1
```

Tensorflow 셋업 - 다운로드

아래 사이트에서 tensorflow-1.8.0-cp35-cp35mu-linux_aarch64 버전을 다운로드 한다.

https://nvidia.app.box.com/v/TF180-Py35-wTRT


tensorflow-1.8.0-cp35-cp35m-li
https://nvidia.app.box.com/v/TF180-Py35-wTRT
오류
로그인
다운로드

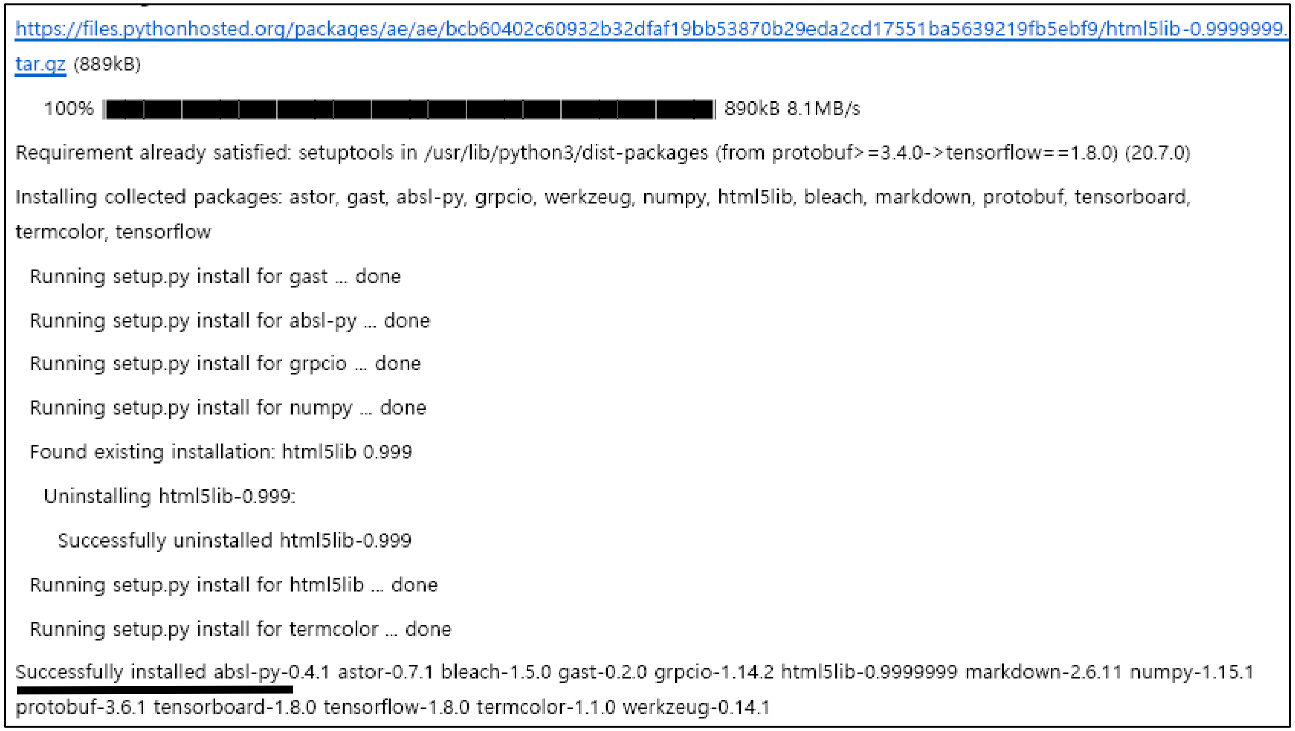
tensorflow-1.8.0-cp35-cp35m-linux_aarch64.whl

죄송합니다. 미리 보기가 로드되지 않았습니다. WHL 파일은 현재 지원되지 않습니다.

Tensorflow 셋업 - 설치

아래 커맨드를 사용하여 tensorflow 설치한다.

python3.5 : $ sudo pip3 install ~/Downloads/tensorflow-1.8.0-cp35-cp35mu-linux_aarch64.whl
python2.7 : $ sudo pip2 install ~/Downloads/tensorflow-1.8.0-cp27-cp27mu-linux_aarch64.whl



```
https://files.pythonhosted.org/packages/ae/ae/bcb60402c60932b32dfaf19bb53870b29eda2cd17551ba5639219fb5ebf9/html5lib-0.9999999.tar.gz (889kB)
    100% |████████████████████████████████| 890kB 8.1MB/s
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from protobuf>=3.4.0->tensorflow==1.8.0) (20.7.0)
Installing collected packages: astor, gast, absl-py, grpcio, werkzeug, numpy, html5lib, bleach, markdown, protobuf, tensorboard, termcolor, tensorflow
  Running setup.py install for gast ... done
  Running setup.py install for absl-py ... done
  Running setup.py install for grpcio ... done
  Running setup.py install for numpy ... done
  Found existing installation: html5lib 0.999
    Uninstalling html5lib-0.999:
      Successfully uninstalled html5lib-0.999
  Running setup.py install for html5lib ... done
  Running setup.py install for termcolor ... done
Successfully installed absl-py-0.4.1 astor-0.7.1 bleach-1.5.0 gast-0.2.0 grpcio-1.14.2 html5lib-0.9999999 markdown-2.6.11 numpy-1.15.1 protobuf-3.6.1 tensorboard-1.8.0 tensorflow-1.8.0 termcolor-1.1.0 werkzeug-0.14.1
```

YOLO V3 셋업

최신 darknet 소스를 복제한다.

```
$ mkdir ~/project
$ cd ~/project
$ git clone https://github.com/pjreddie/darknet yolov3
$ cd yolov3
```

Makefile을 gedit 사용하여 열어서 아래와 같이 수정한다. TX2의 CUDA 아키텍처는 "62"이며 TX1은 "53"입니다.

```
GPU=1
CUDNN=1
OPENCV=1
OPENMP=0
DEBUG=0

ARCH= -gencode arch=compute_53,code=[sm_53,compute_53]\
    -gencode arch=compute_62,code=[sm_62,compute_62]
#ARCH= -gencode arch=compute_30,code=sm_30 \
#     -gencode arch=compute_35,code=sm_35 \
#     -gencode arch=compute_50,code=[sm_50,compute_50] \
#     -gencode arch=compute_52,code=[sm_52,compute_52]
#     -gencode arch=compute_20,code=[sm_20,sm_21] \ This one is deprecated?
# This is what I use, uncomment if you know your arch and want to specify
# ARCH= -gencode arch=compute_52,code=compute_52
```

컴파일이 완료 되면 darknet 파일이 생성된다.

```
$ make
```

```
nvidia@tegra-ubuntu:~/project/yolov3$ ls
backup  data     libdarknet.a  LICENSE.fuck  LICENSE.meta  Makefile  README.md  src
cfg     examples  libdarknet.so  LICENSE.gen  LICENSE.mit  obj      results
darknet  include  LICENSE      LICENSE.gpl  LICENSE.v1   python   scripts
```

YOLO V3 - 실행

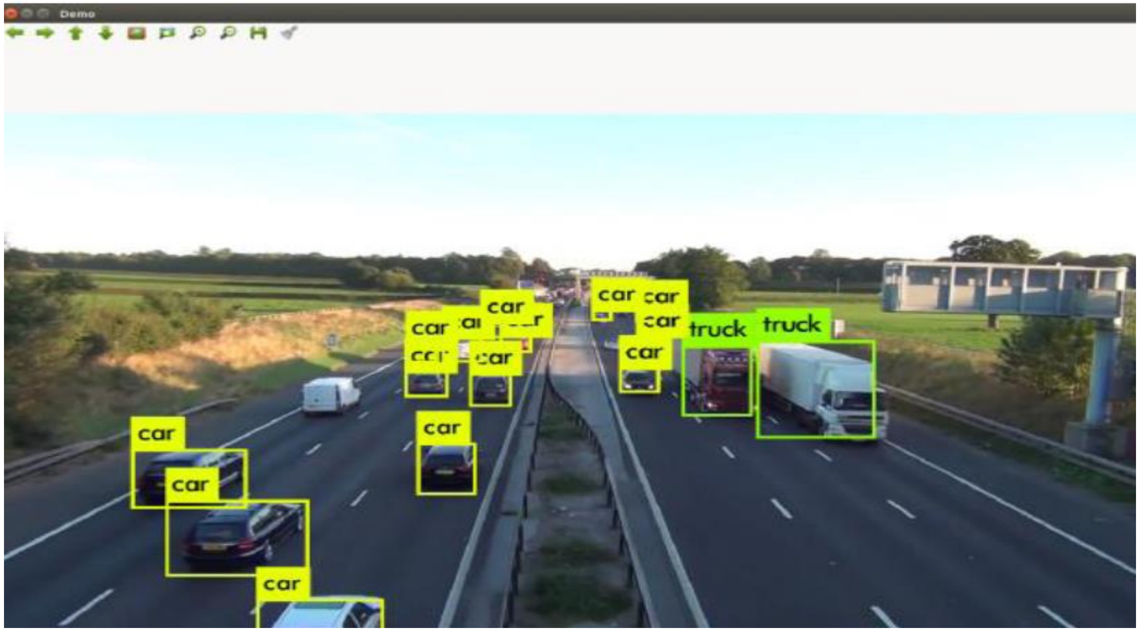
YOLO V3 모델에서 미리 학습된 weight 파일과 cfg 을 다운로드 한다.

```
$ cd ~/project/yolov3
$ wget https://pjreddie.com/media/files/yolov3-tiny.weights
```

YOLO V3 실행하면 아래와 같이 영상 인식이 처리되는 것을 확인 할 수 있다.

```
$ ./darknet detector demo cfg/coco.data cfg/yolov3-tiny.cfg yolov3-tiny.weights -c 1
```

-c 1 은 USB Camera 장치에 해당하는 video1 설정하는 것을 의미한다.
비디오 장치 확인은 ls /dev/video* 커맨드로 확인할 수 있다.



YOLO V3 - 실행

USB 카메라를 사용하지 않고 NVIDIA 카메라로 테스트 하는 경우에 아래와 같이 커맨드를 입력한다.

```
$ ./darknet detector demo cfg/coco.data cfg/yolov3-tiny.cfg yolov3-tiny.weights "nvcamerasrc ! video/x-raw(memory:NVMM),width=(int)1280, height=(int)720,format=(string)I420, framerate=(fraction)30/1 ! nvvidconv flip-method=0 ! video/x-raw,format=(string)BGRx ! videoconvert ! video/x-raw, format=(string)BGR ! appsink"
```

물체 추적 기능 – demo 파일 수정

\src\demo.c 파일 수정하기

```
//code is added -->
#include <fcntl.h>
#include <sys/stat.h>
#include <pthread.h>
#define  FIFO_FROM_YOLO   "/tmp/from_yolo_fifo"
#define  FIFO_TO_YOLO    "/tmp/to_yolo_fifo"
#define  BUFF_SIZE   1024

int   target_class_a = -1;    // add pc
char  buff_a[BUFF_SIZE];      // add pc

int   fd_from_yolo;
int   fd_to_yolo;
//code is added <--
```

t_function_a 스레드 함수 추가하기

```
// code is added -->
void *t_function_a(void *data)
{
   int id;
   int i = 0;
   id = *((int *)data);

   while(1)
   {
      while(read(fd_to_yolo, buff_a, BUFF_SIZE) != NULL)
      {
         if(buff_a[0] == 'A')              // apple  39 - bottle
         {
            target_class_a = 39;
            printf("class %d\n", target_class_a);
            buff_a[0]=0;
         }

      }
   }
}
// code is added <--

void demo(char *cfgfile, char *weightfile, float thresh, ....){

}
```

t_function는 안드로이드 어플로부터 전송받은 객체명에 따라서 target class을 설정한다.
이후 YOLO는 target class을 추적하는 기능을 수행한다.

양재 R&CD 혁신허브
Yangjae Innovation Hub

물체 추적 기능 – demo 파일 수정

\src\demo.c 파일 수정하기

void demo(char *cfgfile, char *weightfile,) : 함수에서 수정

```
{

    // code is added -->
    pthread_t p_thread[2];
    int thr_id;
    int status;
    int a = 1;
    int b = 2;

    thr_id = pthread_create(&p_thread[0], NULL, t_function_a, (void *)&a);
    if (thr_id < 0)
    {
        perror("thread create error : ");
        exit(0);
    }
    // from wifi thread
    if ( -1 == ( fd_from_yolo = open( FIFO_FROM_YOLO, O_RDWR)))
    {
        if ( -1 == mkfifo( FIFO_FROM_YOLO, 0666))
        {
            perror( "mkfifo() run error");
            exit( 1);
        }
        if ( -1 == ( fd_from_yolo = open( FIFO_FROM_YOLO, O_RDWR)))
        {
            perror( "open() error");
            exit( 1);
        }
    }

    // to wifi thread
    if ( -1 == ( fd_to_yolo = open( FIFO_TO_YOLO, O_RDWR)))
    {
        if ( -1 == mkfifo( FIFO_TO_YOLO, 0666))
        {
            perror( "mkfifo() run error");
            exit( 1);
        }
        if ( -1 == ( fd_to_yolo = open( FIFO_TO_YOLO, O_RDWR)))
        {
            perror( "open() error");
            exit( 1);
        }
    }
    // code is added <--

    printf("Demo\n");
    net = load_network(cfgfile, weightfile, 0);
```

물체 추적 기능 – demo 파일 수정

\src\demo.c 파일 수정하기

영상인식을 한 경우 객체 위치에
따라서 모터 제어 명령을 보낸다.

a → left
b → right
c → up
d → down
i → stop

void *detect_in_thread(void *ptr) : 함수에서 수정

```
{
 //code is added -->
#if 0
   draw_detections(display, dets, nboxes, demo_thresh, demo_names, demo_alphabet, demo_classes);
#else
   if(draw_detections(display, dets, nboxes, demo_thresh, demo_names, demo_alphabet, demo_classes,
           target_class_a, &target_xval)) {
                       printf("target class(%d), xval = %f\n", target_class_a, target_xval);

                       if(target_xval > 0.6){
                        buff_a[0] = 'a';
                        write( fd_from_yolo, buff_a, 1 );
                        printf("%c\n", buff_a[0]);
                       }
                       else if(target_xval < 0.4){
                        buff_a[0] = 'b';
                        write( fd_from_yolo, buff_a, 1 );
                        printf("%c\n", buff_a[0]);
                       }
     else{
                        buff_a[0] = 'i';
                        write( fd_from_yolo, buff_a, 1 );
                        printf("%c\n", buff_a[0]);
                      }
   }
   else if(buff_a[0]=='a' || buff_a[0]=='b'){
                     buff_a[0] = 'i';
                        write( fd_from_yolo, buff_a, 1 );
                        printf("%c\n", buff_a[0]);
   }
#endif
   //code is added <--

   free_detections(dets, nboxes);
```

물체 추적 기능 – detector 파일 수정

\examples\detector.c 파일 수정하기

```
//code is added -->
#include <fcntl.h>
#include <sys/stat.h>
#include <pthread.h>
#define  FIFO_FROM_YOLO   "/tmp/from_yolo_fifo“
#define  FIFO_TO_YOLO    "/tmp/to_yolo_fifo“
#define  BUFF_SIZE  1024
int  target_class = -1;
int  counter = 0;

int  fd_from_yolo;
int  fd_to_yolo;
char  buff[BUFF_SIZE];
//code is added <--
```

t_function 스레드 함수 추가하기

```
// code is added -->
void *t_function_a(void *data)
{
  int id;
  int i = 0;
  id = *((int *)data);

  while(1)
  {
    while(read(fd_to_yolo, buff_a, BUFF_SIZE) != NULL)
    {
      if(buff_a[0] == 'A')              // apple  39 - bottle
      {
        target_class_a = 39;
        printf("class %d\n", target_class_a);
        buff_a[0]=0;
      }

    }
  }
}
// code is added <--

void test_detector(char *datacfg, char *cfgfile, ....  ){

}
```

물체 추적 기능 – detector 파일 수정

\examples\detector.c 파일 수정하기

void test_detector(char *datacfg, char *cfgfile, ….) : 함수에서 수정

```
{

 // code is added -->
 pthread_t p_thread[2];
 int thr_id;
 int status;
 int a = 1;
 int b = 2;

 thr_id = pthread_create(&p_thread[0], NULL, t_function_a, (void *)&a);
 if (thr_id < 0)
 {
   perror("thread create error : ");
   exit(0);
 }
 // from wifi thread
 if ( -1 == ( fd_from_yolo = open( FIFO_FROM_YOLO, O_RDWR)))
 {
   if ( -1 == mkfifo( FIFO_FROM_YOLO, 0666))
   {
     perror( "mkfifo() run error");
     exit( 1);
   }
   if ( -1 == ( fd_from_yolo = open( FIFO_FROM_YOLO, O_RDWR)))
   {
     perror( "open() error");
     exit( 1);
   }
 }
 // to wifi thread
 if ( -1 == ( fd_to_yolo = open( FIFO_TO_YOLO, O_RDWR)))
 {
   if ( -1 == mkfifo( FIFO_TO_YOLO, 0666))
   {
     perror( "mkfifo() run error");
     exit( 1);
   }
   if ( -1 == ( fd_to_yolo = open( FIFO_TO_YOLO, O_RDWR)))
   {
     perror( "open() error");
     exit( 1);
   }
 }
printf("wait ...%d\n", target_class);
while(target_class == -1)
  printf("%d\b\b", target_class);
printf("receive class ...\n");
 // code is added <--

while(1){
    if(filename){
```

물체 추적 기능 – detector 파일 수정

\examples\detector.c 파일 수정하기

void test_detector(char *datacfg, char *cfgfile, ….) : 함수에서 수정

```
{
 //code is added -->
#if 0
   draw_detections(display, dets, nboxes, demo_thresh, demo_names, demo_alphabet, demo_classes);
#else
   if(draw_detections(im, dets, nboxes, thresh, names, alphabet, l.classes, target_class, &target_xval)) {
                       printf("target class(%d), xval = %f\n", target_class_a, target_xval);

                       if(target_xval > 0.6){
                        buff[0] = ‘b’;
                        write( fd_from_yolo, buff, 1 );
                        printf("%c\n", buff[0]);
                       }
                       else if(target_xval < 0.4){
                        buff[0] = ‘a’;
                        write( fd_from_yolo, buff, 1 );
                        printf("%c\n", buff[0]);
                       }
     else{
                        buff[0] = 'i';
                        write( fd_from_yolo, buff, 1 );
                        printf("%c\n", buff[0]);
                      }
   }
   else if(buff[0]=='a' || buff[0]=='b'){
                     buff[0] = 'i';
                        write( fd_from_yolo, buff, 1 );
                        printf("%c\n", buff[0]);
   }
#endif
   //code is added <--

free_detections(dets, nboxes);
```

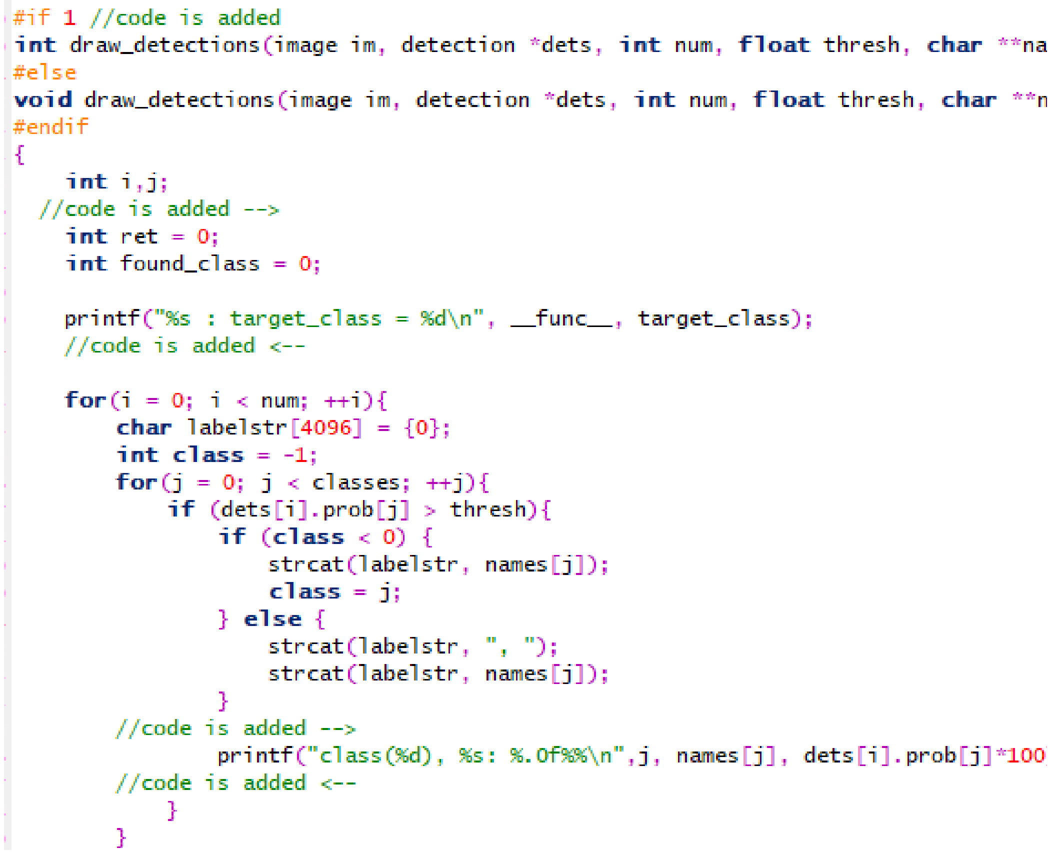
영상인식을 한 경우 객체 위치에
따라서 모터 제어 명령을 보낸다.

a → left
b → right
i → stop

물체추적-image 파일수정

\src\image.c 파일 수정하기

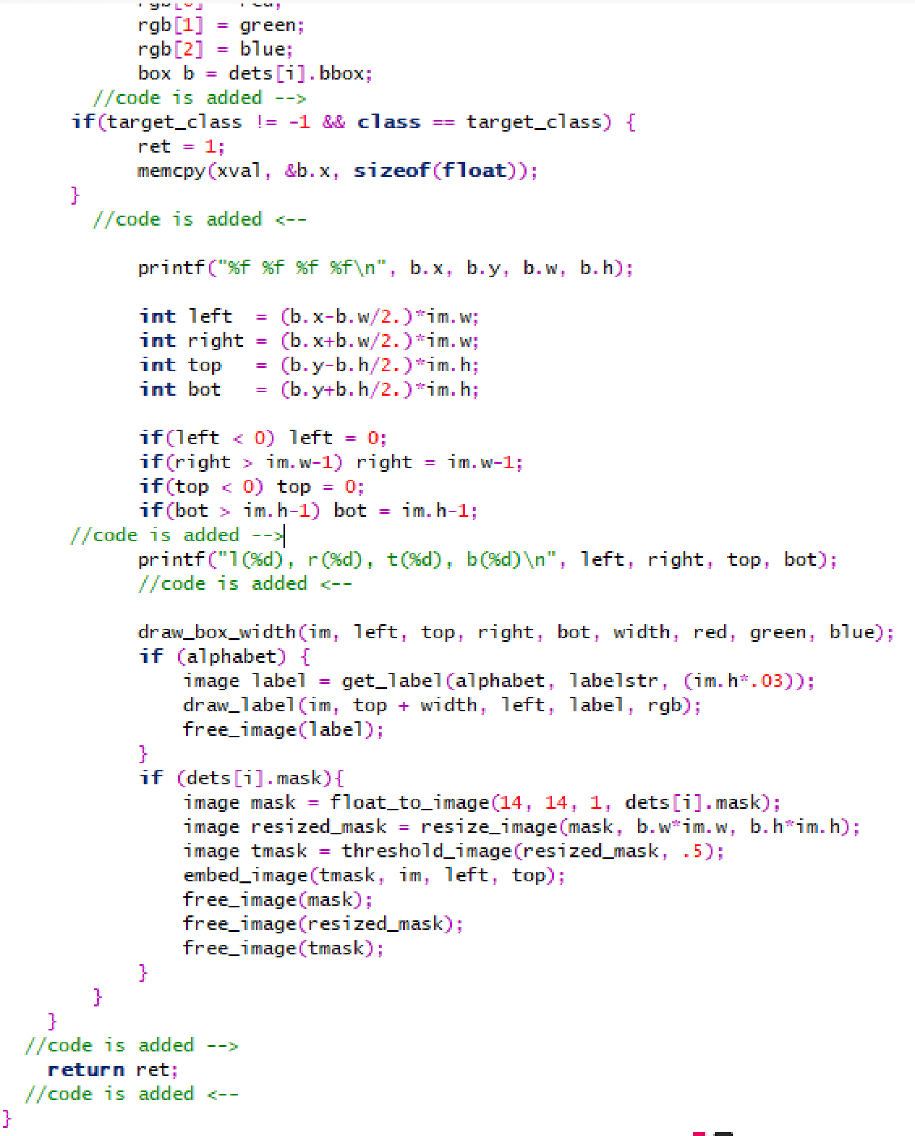
#if 1 //code is added
int draw_detections(image im, detection *dets, int num, float thresh, char **name
target_class, float *xval)
#else
void draw_detections(image im, detection *dets, int num, float thresh, char **nan
#endif



```
#if 1 //code is added
int draw_detections(image im, detection *dets, int num, float thresh, char **na
#else
void draw_detections(image im, detection *dets, int num, float thresh, char **n
#endif
{
    int i,j;
  //code is added -->
    int ret = 0;
    int found_class = 0;

    printf("%s : target_class = %d\n", __func__, target_class);
    //code is added <--

    for(i = 0; i < num; ++i){
        char labelstr[4096] = {0};
        int class = -1;
        for(j = 0; j < classes; ++j){
            if (dets[i].prob[j] > thresh){
                if (class < 0) {
                    strcat(labelstr, names[j]);
                    class = j;
                } else {
                    strcat(labelstr, ", ");
                    strcat(labelstr, names[j]);
                }
        //code is added -->
                printf("class(%d), %s: %.0f%%\n",j, names[j], dets[i].prob[j]*100
        //code is added <--
            }
        }
```



```
      rgb[0] = red;
      rgb[1] = green;
      rgb[2] = blue;
      box b = dets[i].bbox;
    //code is added -->
   if(target_class != -1 && class == target_class) {
      ret = 1;
      memcpy(xval, &b.x, sizeof(float));
   }
    //code is added <--

      printf("%f %f %f %f\n", b.x, b.y, b.w, b.h);

      int left  = (b.x-b.w/2.)*im.w;
      int right = (b.x+b.w/2.)*im.w;
      int top   = (b.y-b.h/2.)*im.h;
      int bot   = (b.y+b.h/2.)*im.h;

      if(left < 0) left = 0;
      if(right > im.w-1) right = im.w-1;
      if(top < 0) top = 0;
      if(bot > im.h-1) bot = im.h-1;
   //code is added -->
      printf("l(%d), r(%d), t(%d), b(%d)\n", left, right, top, bot);
      //code is added <--

      draw_box_width(im, left, top, right, bot, width, red, green, blue);
      if (alphabet) {
        image label = get_label(alphabet, labelstr, (im.h*.03));
        draw_label(im, top + width, left, label, rgb);
        free_image(label);
      }
      if (dets[i].mask){
        image mask = float_to_image(14, 14, 1, dets[i].mask);
        image resized_mask = resize_image(mask, b.w*im.w, b.h*im.h);
        image tmask = threshold_image(resized_mask, .5);
        embed_image(tmask, im, left, top);
        free_image(mask);
        free_image(resized_mask);
        free_image(tmask);
      }
    }
  }
 //code is added -->
  return ret;
 //code is added <--
}
```

물체추적기능-coco,yolo 파일수정

\include\darknet.h 파일 수정하기

```
#if 0 //code is added -->
void draw_detections(image im, detection *dets, int num, float thresh, char **names, image **alphabet, int classes);
#else
int draw_detections(image im, detection *dets, int num, float thresh, char **names, image **alphabet, int classes, int target_class, float *xval);
#endif //code is added <--
```

\examples\coco.c 파일 수정하기

```
void test_coco(char *cfgfile, char *weightfile, char *filename, float thresh)
{

    //code is added -->
    #if 0
    draw_detections(im, dets, l.side*l.side*l.n, thresh, coco_classes, alphabet, 80);
    #else
    draw_detections(im, dets, l.side*l.side*l.n, thresh, coco_classes, alphabet, 80, -1, NULL);
    #endif
    //code is added <--
}
```

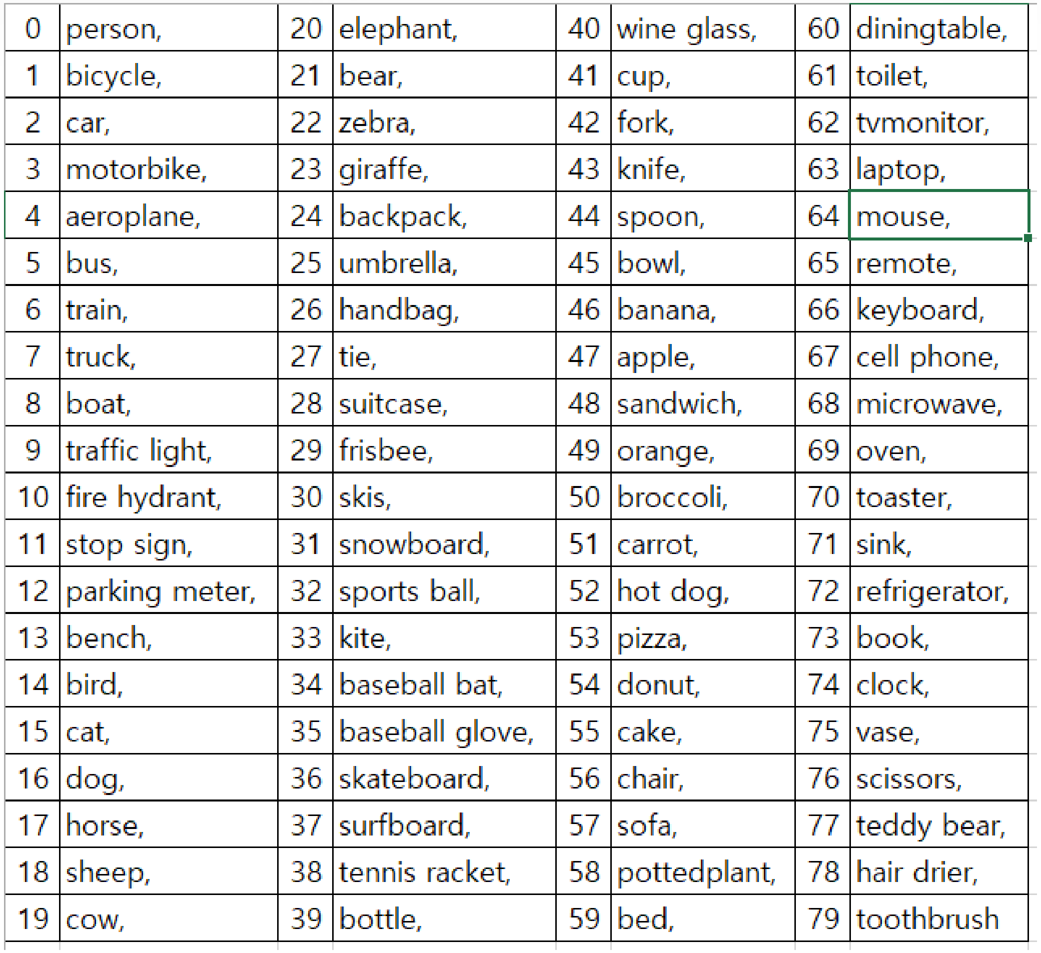
\examples\yolo.c 파일 수정하기

```
void test_yolo(char *cfgfile, char *weightfile, char *filename, float thresh)
{

    #if 0 // code is added -->
    draw_detections(im, dets, l.side*l.side*l.n, thresh, voc_names, alphabet, 20);
    #else
    draw_detections(im, dets, l.side*l.side*l.n, thresh, voc_names, alphabet, 20, -1, NULL);
    #endif // code is added <--
}
```

YOLO V3 객체명과 커맨드

YOLO 에서 사용하는 객체명은 classlist 버퍼에 아래와 같이 정의 되어 있다.



0	person,	20	elephant,	40	wine glass,	60	diningtable,
1	bicycle,	21	bear,	41	cup,	61	toilet,
2	car,	22	zebra,	42	fork,	62	tvmonitor,
3	motorbike,	23	giraffe,	43	knife,	63	laptop,
4	aeroplane,	24	backpack,	44	spoon,	64	mouse,
5	bus,	25	umbrella,	45	bowl,	65	remote,
6	train,	26	handbag,	46	banana,	66	keyboard,
7	truck,	27	tie,	47	apple,	67	cell phone,
8	boat,	28	suitcase,	48	sandwich,	68	microwave,
9	traffic light,	29	frisbee,	49	orange,	69	oven,
10	fire hydrant,	30	skis,	50	broccoli,	70	toaster,
11	stop sign,	31	snowboard,	51	carrot,	71	sink,
12	parking meter,	32	sports ball,	52	hot dog,	72	refrigerator,
13	bench,	33	kite,	53	pizza,	73	book,
14	bird,	34	baseball bat,	54	donut,	74	clock,
15	cat,	35	baseball glove,	55	cake,	75	vase,
16	dog,	36	skateboard,	56	chair,	76	scissors,
17	horse,	37	surfboard,	57	sofa,	77	teddy bear,
18	sheep,	38	tennis racket,	58	pottedplant,	78	hair drier,
19	cow,	39	bottle,	59	bed,	79	toothbrush

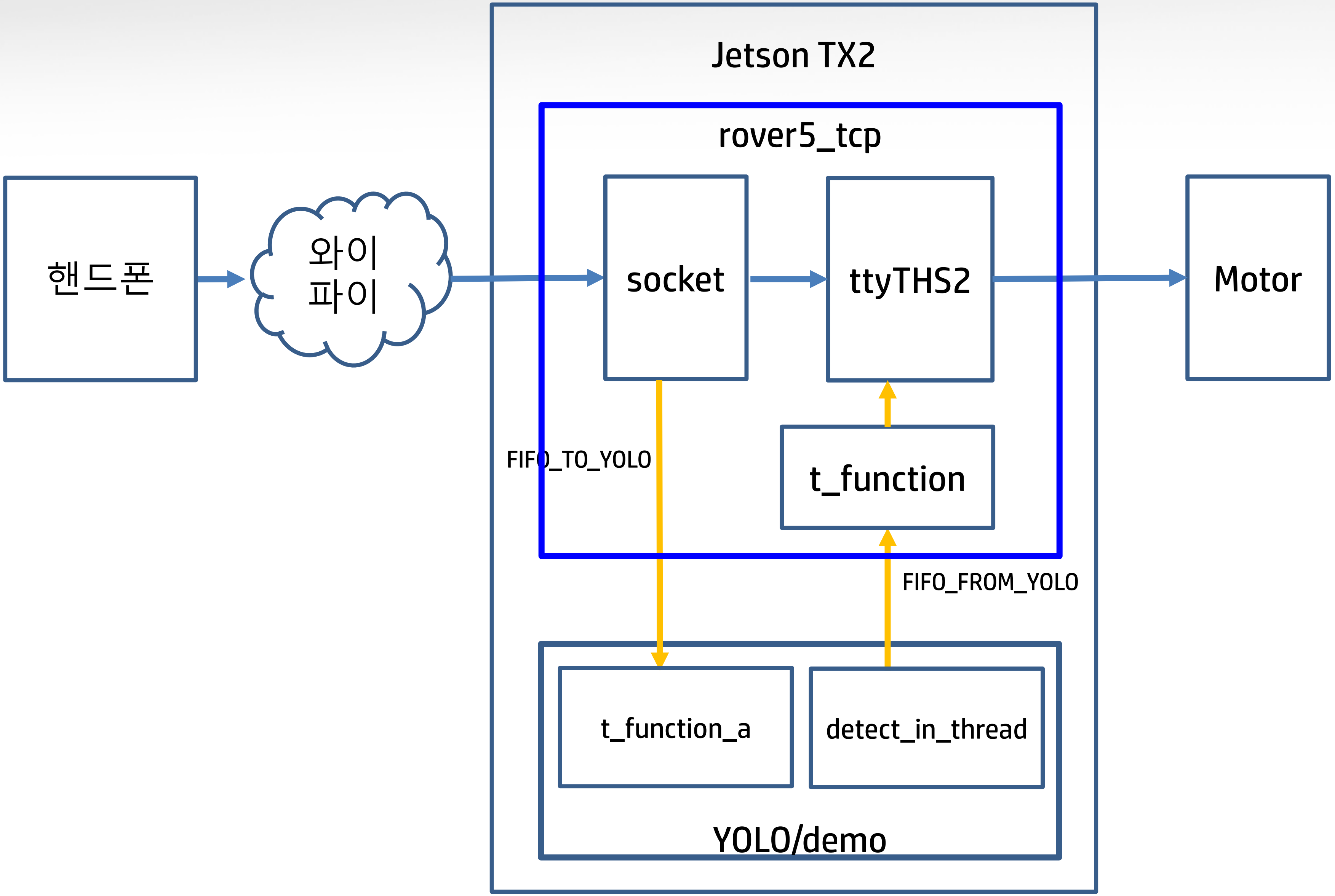
영상인식 객체명 전달

앱->TX2(Wifi)	TX2(Wifi)->YOLO	Class index
apple	A	47
banana	B	46
bicycle	C	1
dog	D	16

모터제어 명령 전달

	YOLO>TX2(WiFi)	TX2(WiFi)->STM32
left	a	a
right	b	b
up	c	c
down	d	d
stop	i	i

명령어 전달 과정


Jetson TX2
rover5_tcp
핸드폰
와이
파이
socket
ttyTHS2
Motor
FIFO_TO_YOLO
t_function
FIFO_FROM_YOLO
t_function_a
detect_in_thread
YOLO/demo