Jetson TX2 기반 YOLO 응용 과정

- Day 3 -

2020.00

양재 R&CD 혁신허브
Yangjae Innovation Hub

모두의연구소

목 차

01. Jetson TX2 Darknet 셋업

[1] Darknet 설치

① Darknet 소스코드를 Github에서 다운로드한다.

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

② Makefile을 gedit으로 열어서 아래와 같이 수정 _ TX2의 CUDA 아키텍처는 "62" 이다.

```
$ gedit Makefile
```

▶ (참조: https://developer.nvidia.com/cuda-gpus)

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

ARCH= -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?
```

01. Jetson TX2 Darknet 셋업

[1] Darknet 설치

③ Make 명령어로 컴파일하면, darknet 파일이 생성된다.

```
$ make -j4
$ sudo ldconfig
```

01. Jetson TX2 Darknet 셋업

[2] YOLOv3 실행 준비

- 일반적으로 물체 인식을 위해서는 두개의 파일이 필요하다

  1) 신경망 레이어 정보가 담긴 .cfg 파일

  2) 가중치 파라미터 정보가 담긴 .weight 파일

- darknet 소스 코드에서 cfg 파일이 존재하나, weights 파일이 없다.

  따라서, weights 이름의 경로를 만들어 yolov3-tiny.weights 파일을 저장한다.

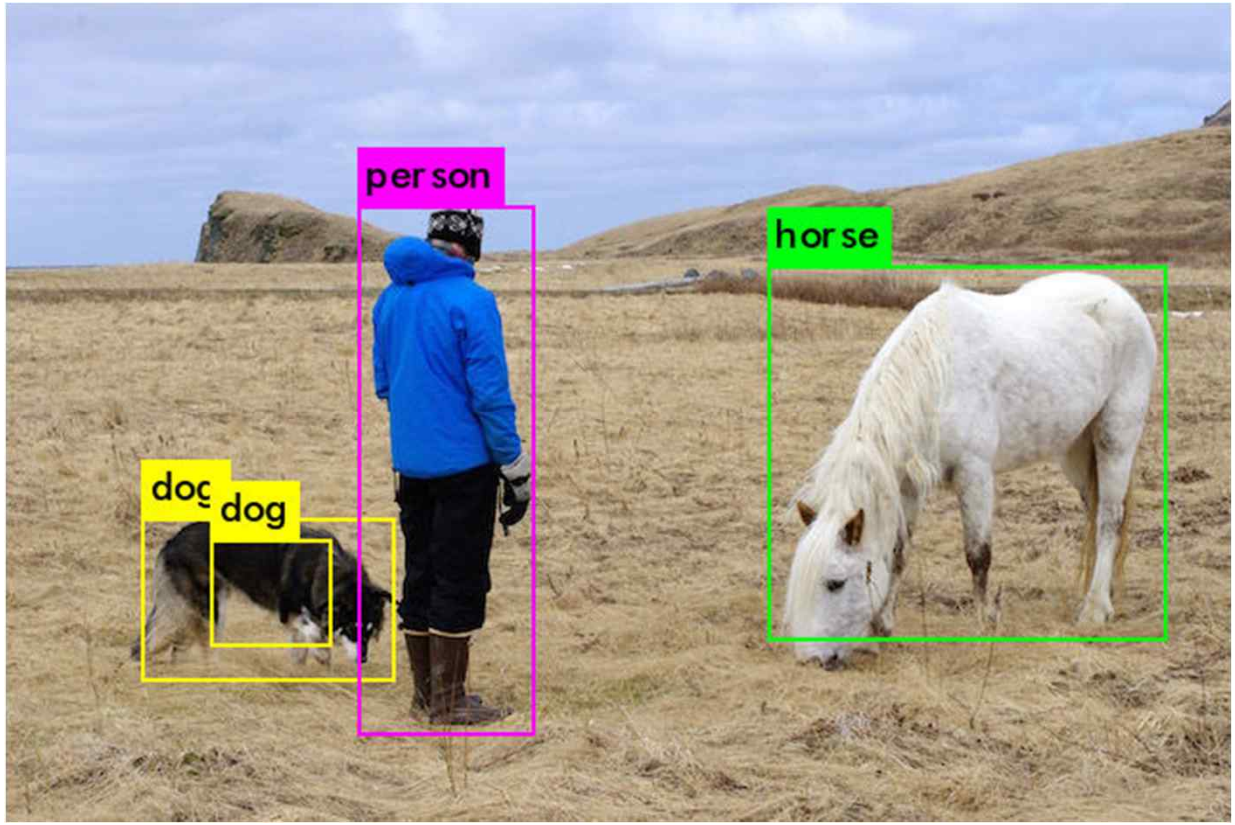
```
$ mkdir ~/Darknet/darknet/weights
$ wget https://pjreddie.com/media/files/yolov3-tiny.weights -P./weights
```

02. YOLOv3 실행 및 예제

[1] YOLOv3 실행

- .cfg 파일과 .weight 파일을 불러와 YOLOv3로 물체 인식을 한다.
- 맨 뒤에는 원하는 사진을 인터넷에서 다운로드하여 다양하게 직접 시험해 볼 수 있다.

```
$ ./darknet detect cfg/yolov3-tiny.cfg weights/yolov3-tiny.weights data/person.jpg
```

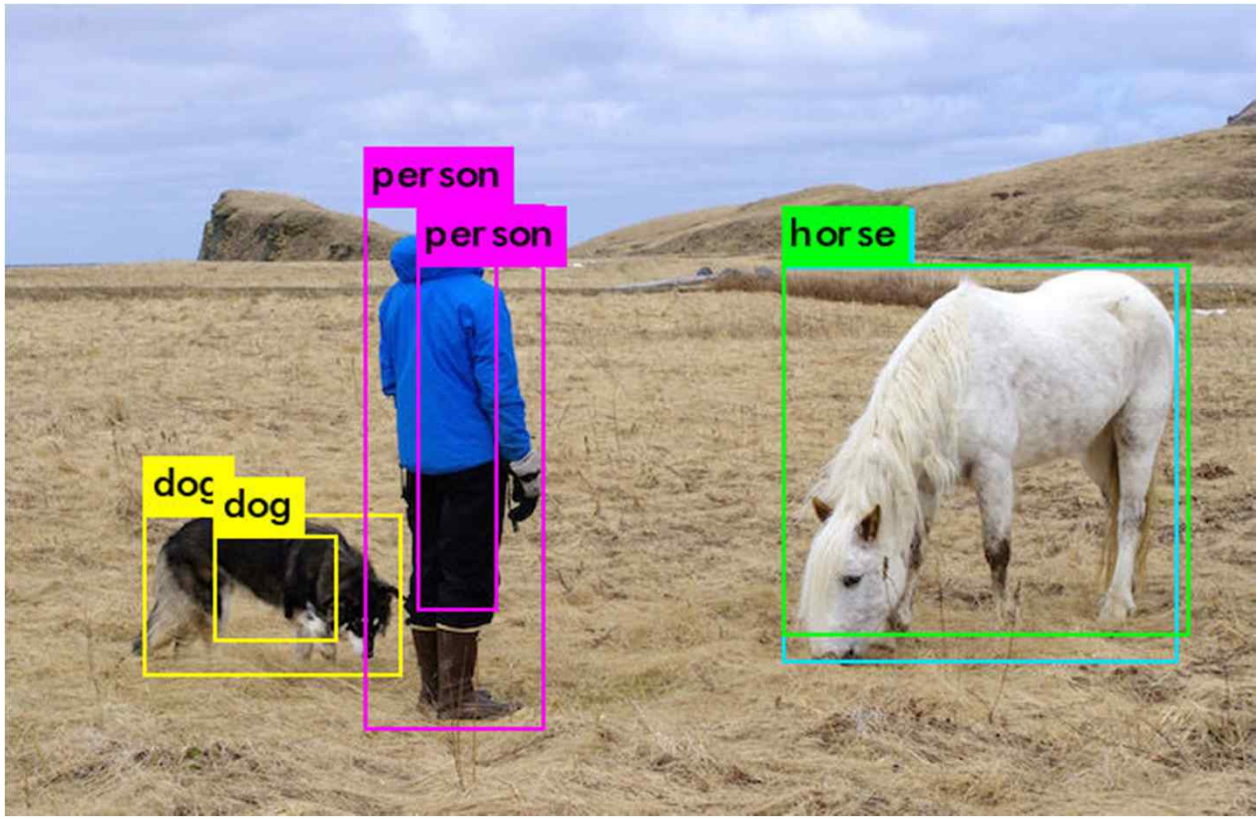


02. YOLOv3 실행 및 예제

[2] YOLOv3 실행 _ 임계치 설정

- 임계치 기본값은 " -thresh 0.5 " 이다. 이를 " –thresh 0.1"로 수정 후 실행 해본다.
- thresh 0.1 : 인식률이 10% 이상인 것을 바운딩 박스로 보여준다.

```
$ ./darknet detect cfg/yolov3-tiny.cfg weights/yolov3-tiny.weights data/person.jpg -thresh 0.1
```

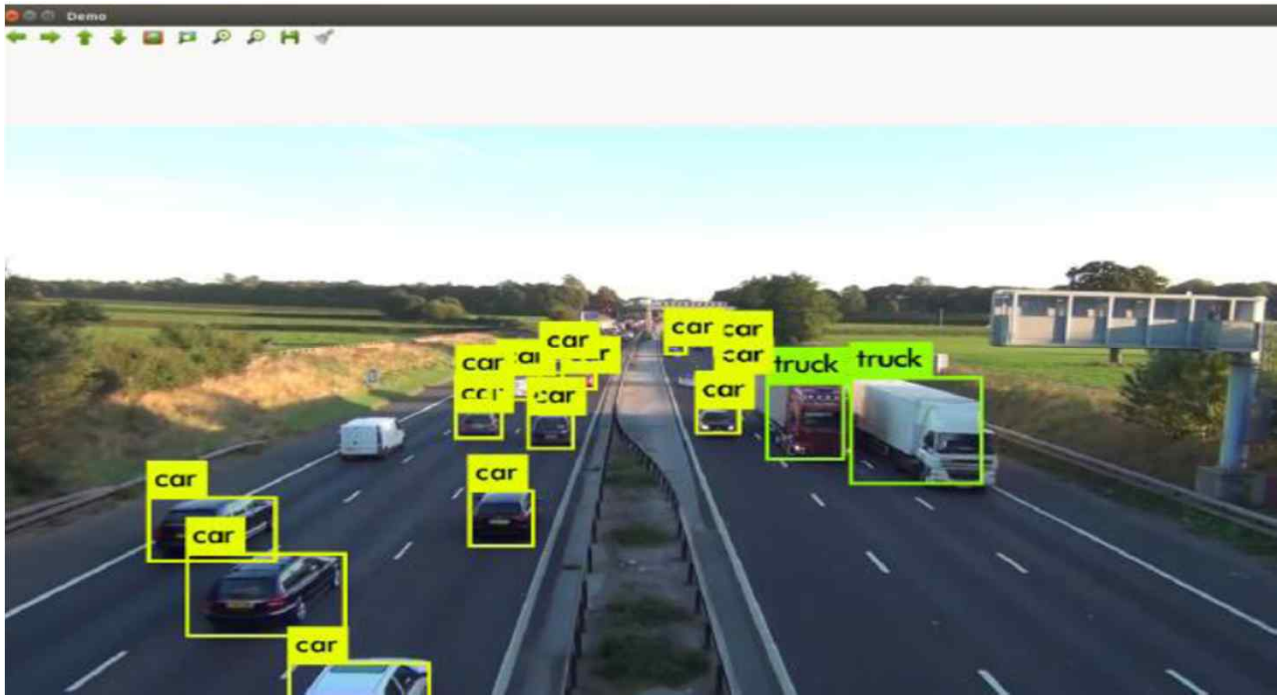


02. YOLOv3 실행 및 예제

[2] YOLOv3 실행 _ 웹캠 사용

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

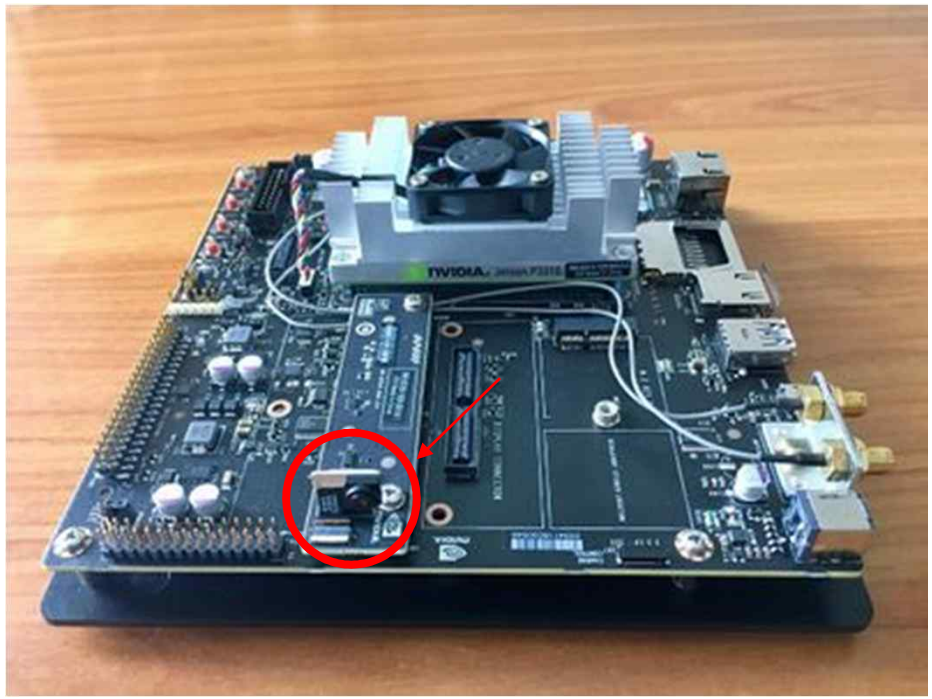
- 웹캠을 사용하여 실시간 영상 물체인식을 한다.
- -c 1 은 USB Camera 장치에 해당하는 /dev/video1 설정하는 것을 의미한다.
- 만일 작동하지 않으면, USB Camera 가 /dev/video0 이나 /dev/video2로 설정되어 있을 수 있다.
  (비디오 장치 확인은 < $ ls /dev/video* > 커맨드로 확인할 수 있다.



02. YOLOv3 실행 및 예제

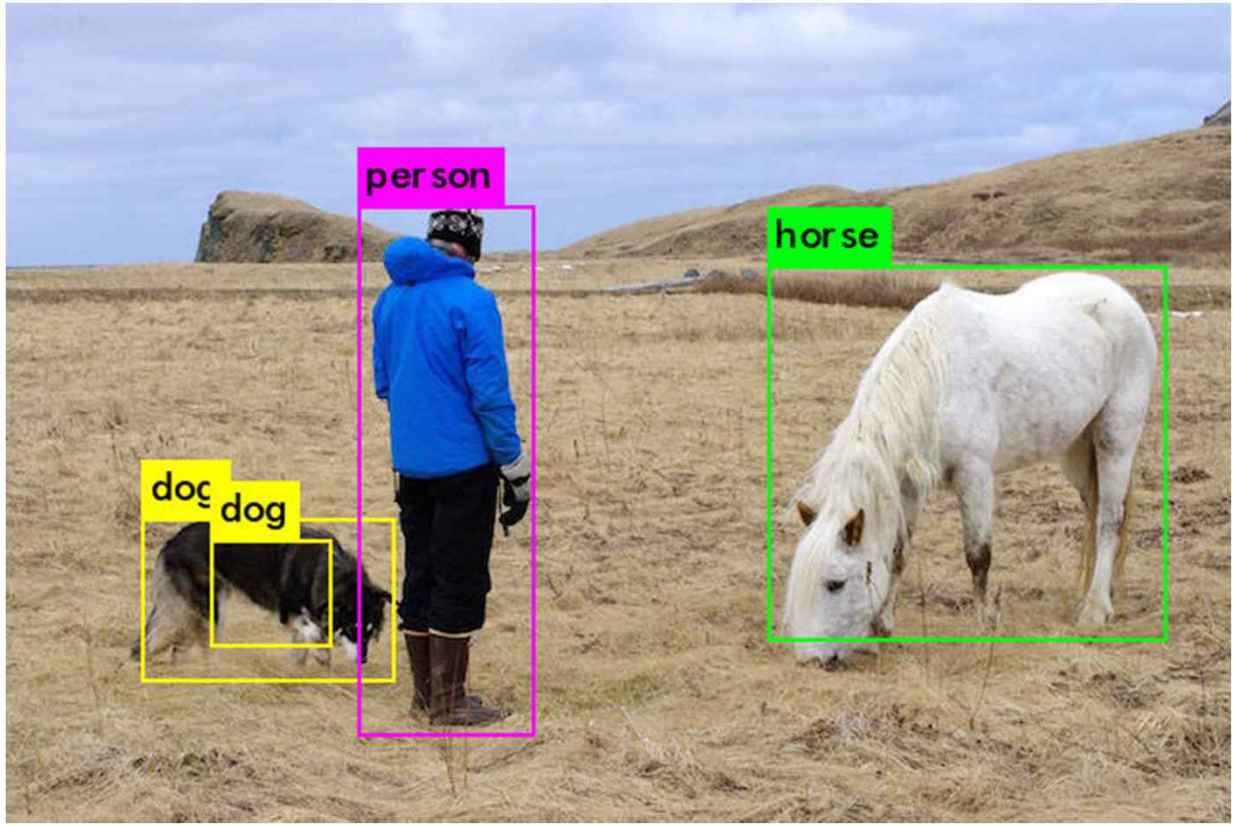
[2] YOLOv3 실행 _ Jetson TX2 온보드 카메라 사용

```
$ ./darknet detector demo cfg/coco.data cfg/yolov3-tiny.cfg weights/yolov3-tiny.weights "'nvarguscamerasrc
! video/x-raw(memory:NVMM), width=1920, height=1080, format=(string)NV12, framerate=(fraction)30/1 ! nvtee
! nvvidconv flip-method=0 ! video/x-raw, width=(int)1280, height=(int)720, format=(string)BGRx !
videoconvert ! appsink'"
```



03. YOLOv3를 통한 물체추적

- Bounding Box 위치 = 물체의 위치
- Bounding Box 위치 값을 이용하여, 물체를 추적하는 알고리즘을 생성.
- Darknet 에서 Bounding Box 위치 값을 나타내는 함수는 **draw_detections( )**
- 따라서, draw_detections( ) 함수를 확장하여, 물체추적 시스템 제작.



03. YOLOv3를 통한 물체추적

▶ draw_detections( ) 함수가 프로토 타입 선언, 정의, 호출되고 있는 위치

선언	함수를 사용할 수 있도록 미리 알리는 부분	
	darknet/include/darknet.h	

정의	함수가 수행하는 기능이 작성된 부분	
	darknet/src/image.c	

호출	함수를 사용하는 부분	
	darknet/src/demo.c	TCP 프로세스와 통신을 위해서 thread 및 fifo 생성 및 사용
	darknet/examples/detector.c	단순히 인자만 확장
	darknet/examples/coco.c	단순히 인자만 확장
	darknet/examples/yolo.c	단순히 인자만 확장

03. YOLOv3를 통한 물체추적

▶ 요약

1. thread 및 fifo 생성 - src/demo.c

2. draw_detections() 함수 프로토 타입 수정 - include/darknet.h
   - return : void -> int (target class가 있는 경우 1)
   - tartget class와 좌표값 인자로 추가

3. draw_detections() 함수 원형 수정 - src/image.c
   - tartget class가 있는 경우 좌표값 인자로 전달

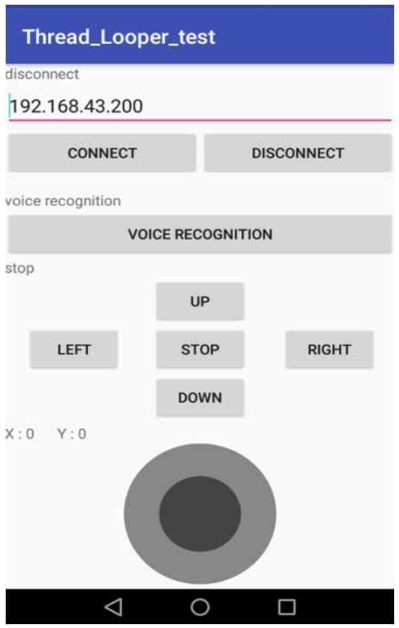
4. draw_detections() 함수 호출 후 모터 구동 - src/demo.c
   - return이 1인 경우 모터 구동 명령을 fifo를 통하여 tcp thread에 전달

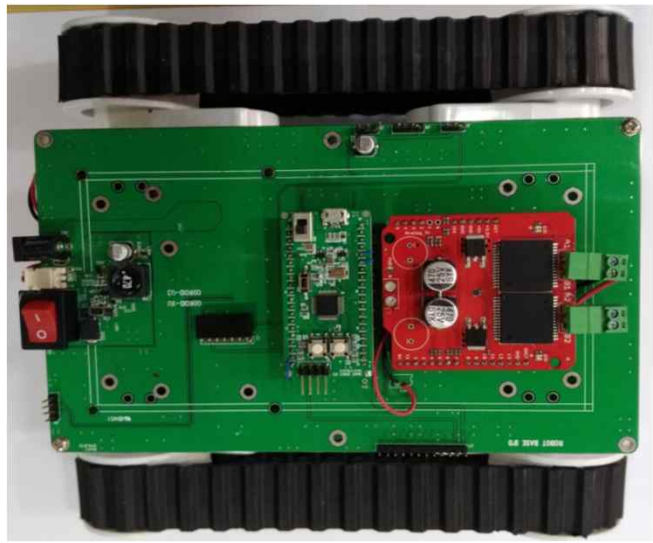
examples/detector.c, examples/coco.c, examples/yolo.c : draw_detections() 호출 함수 수정

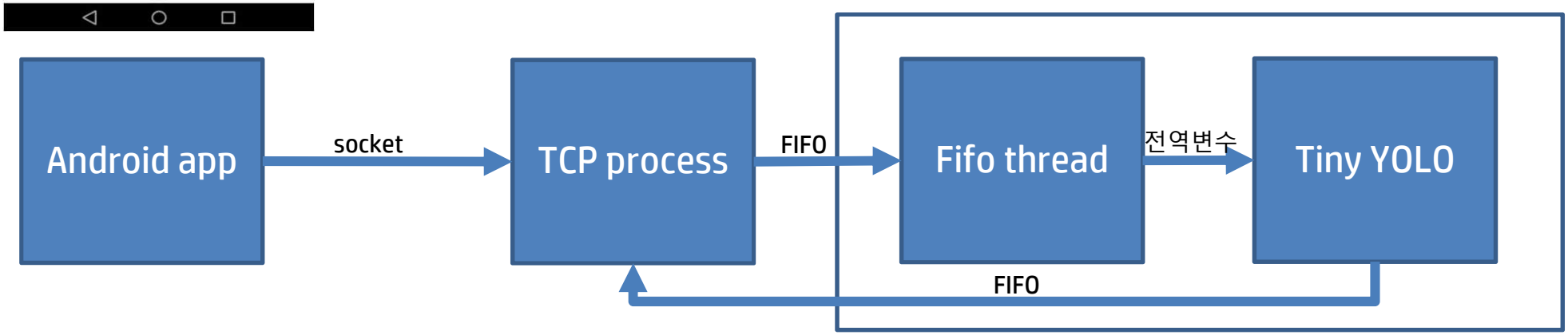
```
draw_detections(im, dets, nboxes, thresh, names, alphabet, l.classes, -1, NULL, NULL, NULL);
```

03. YOLOv3를 통한 물체추적

▶ 개요







03. YOLOv3를 통한 물체추적

src/demo.c 에서 socket과 통신하기 위한 thread 추가, fifo 2개 추가

./src/demo.c

12 line

```
//SECTION 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


//temp human
//int   target_class_a = -1;
int   target_class_a = 0;
char  buff_a[BUFF_SIZE];


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

100 line 근처

238번 줄 부터

```
//SECTION code is added -->
void *t_function_a(void *data)
{
  int id;
  id = *((int *)data);
  while(1)  {
    while(read(fd_to_yolo, buff_a, BUFF_SIZE) != 0)  {
      if(buff_a[0] == 'A')  {            // apple  39 - bottle
        target_class_a = 39;
        printf("class %d\n", target_class_a);
        buff_a[0]=0;
      }
      if(buff_a[0] == 'B')  {            // banana 0 - person
        target_class_a = 0;
        printf("class %d\n", target_class_a);
        buff_a[0]=0;
      }
      if(buff_a[0] == 'C')  {            // bicycle
        target_class_a = 1;
        printf("class %d\n", target_class_a);
        buff_a[0]=0;
      }
      if(buff_a[0] == 'D')  {            // dog
        target_class_a = 16;
        printf("class %d\n", target_class_a);
        buff_a[0]=0;
      }
      if(buff_a[0] == 'E')  {            // truck
        target_class_a = 7;
```

03. YOLOv3를 통한 물체추적

src/demo.c 에서 socket과 통신하기 위한 thread 추가, fifo 2개 추가

./src/demo.c

238번 줄 부터

130line

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


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);
   }
}
```

03. YOLOv3를 통한 물체추적

include/darknet.h 에서 draw_detections() proto type 선언

./include/darknet.h

736번 줄 부터

```
//SECTION code is added -->
#if 0
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,
float *wval, float *hval);
#endif
//!SECTION code is added <--
```

draw_detections proto type 선언

03. YOLOv3를 통한 물체추적

src/image.c 에서 draw_detections() 함수원형 정의

./src/image.c

238번 줄 부터

```
#if 1 //SECTION code is added
int draw_detections(image im, detection *dets, int num, float thresh, char **names, image
**alphabet, int classes, int target_class, float *xval, float *wval, float *hval)
#else
void draw_detections(image im, detection *dets, int num, float thresh, char **names,
image **alphabet, int classes)
#endif
{
   int i,j;
   //code is added -->
      int ret = 0;
      int found_class = 0;


      printf("[image.c] %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[i]):
```

03. YOLOv3를 통한 물체추적

src/demo.c 에서 draw_detections() 사용 및 후 처리

./src/demo.c

238번 줄 부터

130line

```
//지역 변수 추가
  //code is added -->
  float target_xval = .0f;
  float target_wval = .0f;
  float target_hval = .0f;
  float distance_val = .0f;
  //code is added <--
```

```
  //SECTION code is added -->
  #if 0
    draw_detections(display, dets, nboxes, demo_thresh, demo_names,
demo_alphabet, demo_classes);
  #else
  #else
    if(draw_detections(display, dets, nboxes, demo_thresh, demo_names,
demo_alphabet, demo_classes, target_class_a, &target_xval,
&target_wval, &target_hval)) {
      distance_val = target_wval*target_hval;
      printf("[demo.c] target class(%d), xval = %f, wval = %f, hval = %f
distance_val = %f \n", target_class_a, target_xval, target_wval,
```

03. YOLOv3를 통한 물체추적

example/yolo.c , coco.c, detector.c 에 적용

./src/demo.c

238번 줄 부터

```
// code is added -->
#if 0
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, NULL, NULL);
#endif
// code is added <--
```

03. YOLOv3를 통한 물체추적 -

./src/image.c

238번 줄 부터

```
232             sprintf(buff, "data/labels/%d_%d.png", i, j);
233             alphabets[j][i] = load_image_color(buff, 0, 0);
234         }
235     }
236     return alphabets;
237 }
238
239 #if 0 //SECTION code is added
240 void draw_detections(image im, detection *dets, int num, float thresh, char **names, image **alphabet, int classes)
241 #else
242 int  draw_detections(image im, detection *dets, int num, float thresh, char **names, image **alphabet, int classes, int target_class, float
243 #endif
244 {
245     int i,j;
246     //code is added -->
247 #if 0
248 #else
249         int ret = 0;
250         //int found_class = 0;
251
252         printf("[image.c] %s : target_class = %d \n",__func__,target_class);
253 #endif
254     //code is added <--
255
256     for(i = 0; i < num; ++i){
257         char labelstr[4096] = {0};
```

03. YOLOv3를 통한 물체추적

./src/image.c

269번 줄 부터

```
264                 } else {
265                     strcat(labelstr, ", ");
266                     strcat(labelstr, names[j]);
267                 }
268                 //code is added -->
269 #if 0
270                 printf("%s: %.0f%%\n", names[j], dets[i].prob[j]*100);
271 #else
272                 printf("class(%d), %s : %.0f%%\n", j, names[j], dets[i].prob[j]*100);
273 #endif
274                 //code is added <--
275             }
276         }
277         if(class >= 0){
278             int width = im.h * .006;
```

03. YOLOv3를 통한 물체추적

./src/image.c

302번 줄 부터

```
298             rgb[2] = blue;
299             box b = dets[i].bbox;
300
301             //code is added -->
302 #if 0
303 #else
304             if(target_class != -1 && class == target_class){
305                 ret = 1;
306                 memcpy(xval, &b.x, sizeof(float));
307                 memcpy(yval, &b.y, sizeof(float));
308                 memcpy(wval, &b.w, sizeof(float));
309                 memcpy(hval, &b.h, sizeof(float));
310             }
311             printf("%f %f %f %f\n", b.x, b.y, b.w, b.h);
312 #endif
313             //code is added <--
314
315             int left  = (b.x-b.w/2.)*im.w;
316             int right = (b.x+b.w/2.)*im.w;
```

03. YOLOv3를 통한 물체추적

./src/image.c

326번 줄 부터

```
322             if(top < 0) top = 0;
323             if(bot > im.h-1) bot = im.h-1;
324
325             //code is added -->
326 #if 0
327 #else
328             printf("left(%d), right(%d), top(%d), bottom(%d)\n",left, right,top,bot);
329 #endif
330             //code is added <--
331
332             draw_box_width(im, left, top, right, bot, width, red, green, blue);
333             if (alphabet) {
334                 image label = get_label(alphabet, labelstr, (im.h*.03));
335                 draw_label(im, top + width, left, label, rgb);
336                 free_image(label);
337             }
338             if (dets[i].mask){
339                 image mask = float_to_image(14, 14, 1, dets[i].mask);
```

03. YOLOv3를 통한 물체추적

./src/image.c

350번 줄 부터

```
342                 embed_image(tmask, im, left, top);
343                 free_image(mask);
344                 free_image(resized_mask);
345                 free_image(tmask);
346             }
347         }
348     }
349     //code is added -->
350 #if 0
351 #else
352     return ret;
353 #endif
354     //code is added <--
355 }
356
357 void transpose_image(image im)
358 {
359     assert(im.w == im.h);
360     int n, m;
361     int c;
```

03. YOLOv3를 통한 물체추적

./include/darknet.h

736번 줄 부터

```
730  void visualize_network(network *net);
731  float box_iou(box a, box b);
732  data load_all_cifar10();
733  box_label *read_boxes(char *filename, int *n);
734  box float_to_box(float *f, int stride);
735
736  //SECTION code is added -->
737  #if 0
738  void draw_detections(image im, detection *dets, int num, float thresh, char **names, image **alphabet, int classes);
739  #else
740  int  draw_detections(image im, detection *dets, int num, float thresh, char **names, image **alphabet, int classes, int target_class, float
741  #endif
742  //!SECTION code is added <--
743
744  matrix network_predict_data(network *net, data test);
745  image **load_alphabet();
746  image get_network_image(network *net);
747  float *network_predict(network *net, float *input);
```

03. YOLOv3를 통한 물체추적

./src/demo.c

11번 줄 부터

```
#include "demo.h"
#include <sys/time.h>
//SECTION code is added -->
#include <stdio.h>
#include <stdlib.h>

int target_class_a = 0;
//SECTION code is added <--
#define DEMO 1
```

93번 줄 부터

```
void *detect_in_thread(void *ptr)
{
    running = 1;
    float nms = .4;

    //code is added -->
    float target_xval = .0f;
    float target_wval = .0f;
    float target_hval = .0f;
    float distance_val = .0f;
    //code is added <--

    layer l = net->layers[net->n-1];
    float *X = buff_letter[(buff_index+2)%3].data;
    network_predict(net, X);
```

03. YOLOv3를 통한 물체추적

./src/demo.c

143번 줄 부터

```
image display = buff[(buff_index+2) % 3];
//SECTION 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, &target_wval, &target_hval))
{
    distance_val = target_wval*target_hval;
    printf("[demo.c] target class(%d), xval = %f, wval = %f, hval = %f distance_val = %f \n", target_class_a, target_xval, target_wval, target_hval, distance_val);
    if(target_xval > 0.6){
        printf("r\n");
        system("sudo echo 0 > /sys/class/gpio/gpio298/value");
        system("sudo echo 1 > /sys/class/gpio/gpio396/value");
    }else if(target_xval < 0.4){
        printf("l\n");
        system("sudo echo 1 > /sys/class/gpio/gpio298/value");
        system("sudo echo 0 > /sys/class/gpio/gpio396/value");
    }else{
        printf("s\n");
        system("sudo echo 1 > /sys/class/gpio/gpio298/value");
        system("sudo echo 1 > /sys/class/gpio/gpio396/value");
    }
}else {
    printf("i\n");
    system("sudo echo 0 > /sys/class/gpio/gpio392/value");
    system("sudo echo 0 > /sys/class/gpio/gpio396/value");
}
#endif
//!SECTION code is added <--
```

03. YOLOv3를 통한 물체추적

./src/demo.c

226번 줄 부터

```
void demo(char *cfgfile, char *weightfile, float thresh, int cam_index, const char *filename, char **names, int classes, int delay, char *prefix, int
avg_frames, float hier, int w, int h, int frames, int fullscreen)
{
//SECTION code is added -->
   system("sudo echo 298 > /sys/class/gpio/export");
   system("sudo echo 396 > /sys/class/gpio/export");
   system("sudo echo out > /sys/class/gpio/gpio298/direction");
   system("sudo echo out > /sys/class/gpio/gpio396/direction");
//SECTION code is added <--
   //demo_frame = avg_frames;
   image **alphabet = load_alphabet();
   demo_names = names;
```

03. YOLOv3를 통한 물체추적

./examples/detector.c

604번 줄 부터

```
//printf("%d\n", nboxes);
//if (nms) do_nms_obj(boxes, probs, l.w*l.h*l.n, l.classes, nms);
if (nms) do_nms_sort(dets, nboxes, l.classes, nms);
draw_detections(im, dets, nboxes, thresh, names, alphabet, l.classes, -1, NULL, NULL, NULL);
free_detections(dets, nboxes);
if(outfile){
   save_image(im, outfile);
}
```

03. YOLOv3를 통한 물체추적

./examples/yolo.c

296번 줄 부터

```
int nboxes = 0;
detection *dets = get_network_boxes(net, 1, 1, thresh, 0, 0, 0, &nboxes);
if (nms) do_nms_sort(dets, l.side*l.side*l.n, l.classes, nms);
// code is added -->
#if 0
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, NULL, NULL);
#endif
// code is added <--
save_image(im, "predictions");
show_image(im, "predictions", 0);
```

03. YOLOv3를 통한 물체추적

./examples/coco.c

326번 줄 부터

```
detection *dets = get_network_boxes(net, 1, 1, thresh, 0, 0, 0, &nboxes);
if (nms) do_nms_sort(dets, l.side*l.side*l.n, l.classes, nms);

//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, NULL, NULL);
#endif
//code is added <--

save_image(im, "prediction");
show_image(im, "predictions", 0);
```

03. YOLOv3를 통한 물체추적

▶ YOLO에서 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

- 영상 인식 객체명 전달

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

- 모터 제어 명령 전달

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