

```
%cd ..
from google.colab import drive
drive.mount('/content/gdrive')
```

```
↗ /
Mounted at /content/gdrive
```

```
!ln -s /content/gdrive/My Drive/ /mydrive
```

```
%cd /mydrive/yolov4
```

```
↗ /content/gdrive/My Drive/yolov4
```

```
!git clone https://github.com/AlexeyAB/darknet
```

```
↗ fatal: destination path 'darknet' already exists and is not an empty directory.
```

```
%cd darknet/
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
!sed -i 's/LIBSO=0/LIBSO=1/' Makefile
```

```
↗ /content/gdrive/My Drive/yolov4/darknet
```

```
!make
```

```
↗ chmod +x *.sh
g++ -std=c++11 -std=c++11 -include/ -I3rdparty/stb/include -DOPENCV `pkg-config --cflags opencv4 2> /dev/null || pkg-config --cflags opencv`
./src/image_opencv.cpp: In function 'void draw_detections_cv_v3(void**, detection*, int, float, char**, image**, int, int)':
./src/image_opencv.cpp:946:23: warning: variable 'rgb' set but not used [-Wunused-but-set-variable]
  946 |         float rgb[3];
      |         ^~~
./src/image_opencv.cpp: In function 'void draw_train_loss(char*, void**, int, float, float, int, int, float, int, char*, float, int, int, double)':
./src/image_opencv.cpp:1147:13: warning: this 'if' clause does not guard... [-Wmisleading-indentation]
 1147 |         if (iteration_old == 0)
      |         ^~
./src/image_opencv.cpp:1150:17: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the 'if'
 1150 |         if (iteration_old != 0){
      |         ^~
./src/image_opencv.cpp: In function 'void cv_draw_object(image, float*, int, int, int*, float*, int*, int, char**)' :
./src/image_opencv.cpp:1444:14: warning: unused variable 'buff' [-Wunused-variable]
 1444 |         char buff[100];
      |         ^~~~~
./src/image_opencv.cpp:1420:9: warning: unused variable 'it_tb_res' [-Wunused-variable]
 1420 |         int it_tb_res = cv::createTrackbar(it_trackbar_name, window_name, &it_trackbar_value, 1000);
      |         ^~~~~~
./src/image_opencv.cpp:1424:9: warning: unused variable 'lr_tb_res' [-Wunused-variable]
 1424 |         int lr_tb_res = cv::createTrackbar(lr_trackbar_name, window_name, &lr_trackbar_value, 20);
      |         ^~~~~~
./src/image_opencv.cpp:1428:9: warning: unused variable 'cl_tb_res' [-Wunused-variable]
 1428 |         int cl_tb_res = cv::createTrackbar(cl_trackbar_name, window_name, &cl_trackbar_value, classes-1);
      |         ^~~~~~
./src/image_opencv.cpp:1431:9: warning: unused variable 'bo_tb_res' [-Wunused-variable]
 1431 |         int bo_tb_res = cv::createTrackbar(bo_trackbar_name, window_name, boxonly, 1);
      |         ^~~~~~
g++ -std=c++11 -std=c++11 -include/ -I3rdparty/stb/include -DOPENCV `pkg-config --cflags opencv4 2> /dev/null || pkg-config --cflags opencv`
./src/http_stream.cpp: In member function 'bool JSON_sender::write(const char*)':
./src/http_stream.cpp:253:21: warning: unused variable 'n' [-Wunused-variable]
  253 |         int n = _write(client, outputbuf, outlen);
      |         ^
./src/http_stream.cpp: In member function 'bool MJPG_sender::write(const cv::Mat&)':
./src/http_stream.cpp:511:95: warning: format '%zu' expects argument of type 'size_t', but argument 3 has type 'int' [-Wformat=]
  511 |         sprintf(head, "--mjpegstreamWrWnContent-Type: image/jpegWrWnContent-Length: %zuWrWnWrWn", outlen);
      |                                     ~~~~~^~~~~~
      |                                     |         |
      |                                     |         int
      |                                     long unsigned int
      |                                     %u
./src/http_stream.cpp: In function 'void set_track_id(detection*, int, float, float, float, int, int, int)':
./src/http_stream.cpp:867:27: warning: comparison of integer expressions of different signedness: 'int' and 'std::vector<detection_t>::size_t'
  867 |         for (int i = 0; i < v.size(); ++i) {
      |                        ~~~~~^~~~~~
./src/http_stream.cpp:875:33: warning: comparison of integer expressions of different signedness: 'int' and 'std::vector<detection_t>::size_t'
  875 |         for (int old_id = 0; old_id < old_dets.size(); ++old_id) {
      |                        ~~~~~^~~~~~
./src/http_stream.cpp:894:31: warning: comparison of integer expressions of different signedness: 'int' and 'std::vector<detection_t>::size_t'
  894 |         for (int index = 0; index < new_dets_num*old_dets.size(); ++index) {
      |                        ~~~~~^~~~~~
./src/http_stream.cpp:930:28: warning: comparison of integer expressions of different signedness: 'std::deque<std::vector<detection_t>>::size_t' and 'std::vector<detection_t>::size_t'
  930 |         if (old_dets_dq.size() > deque_size) old_dets_dq.pop_front();
```

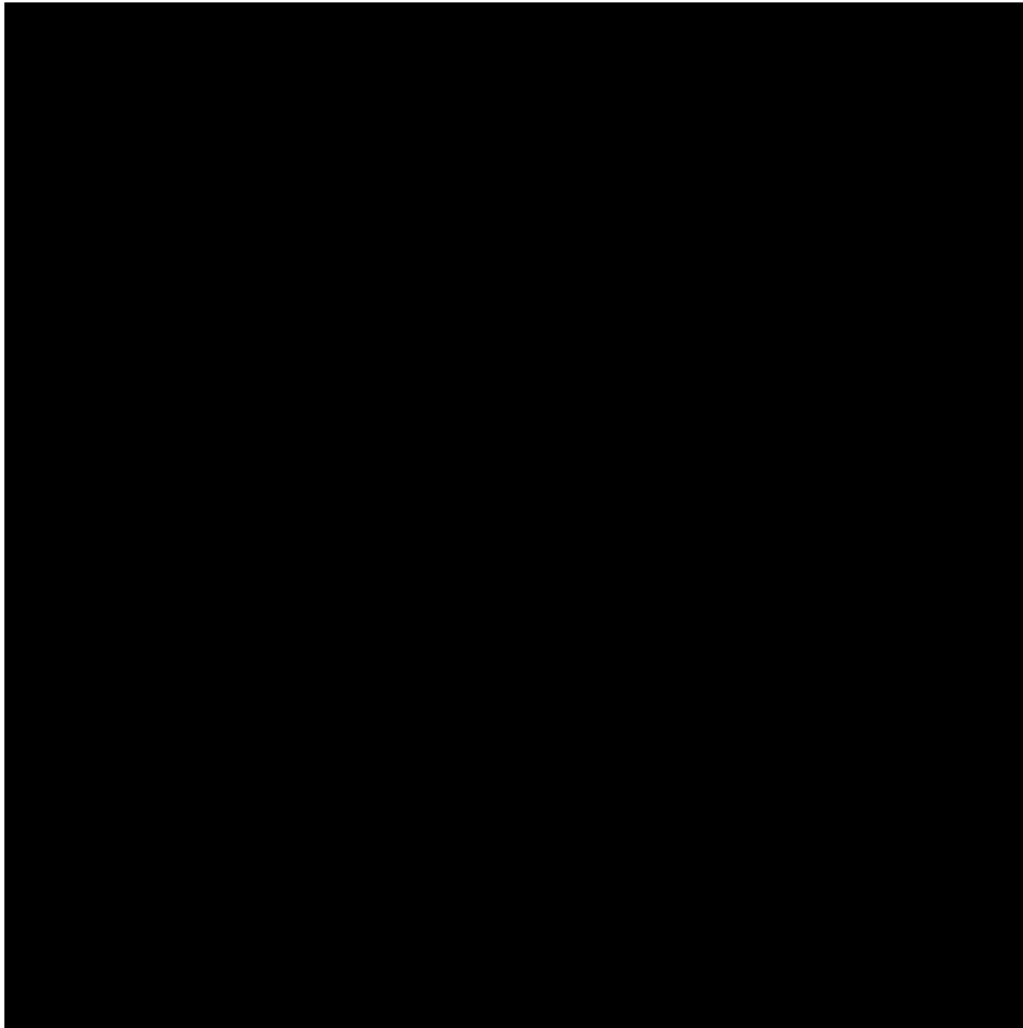
```
!cp /mydrive/yolov4/yolov4-custom.cfg cfg

!cp /mydrive/yolov4/obj.names data
!cp /mydrive/yolov4/obj.data data

!cp /mydrive/yolov4/process.py .

!python process.py
```

```
!./darknet detector test data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom_best.weights /mydrive/food_test_images/image1.jpg
imshow('predictions.jpg')
```

 /bin/bash: line 1: ./darknet: Permission denied

```
listResult = []  
recipeBook = {}  
posMax = 0  
recipe = []  
recipeProcess = [""- 소스
```

진간장 2큰술(20g)  
황설탕 2큰술(20g)  
케찹 4큰술(60g)  
식초 1큰술(10g)  
간생강 1/3큰술(5g)

- 볶음

비엔나소시지 250g  
식용유 3큰술(20g)  
마늘 10개(40g)  
양파 1/2개(80g)  
양송이 2개(50g)  
파프리카 1/4개(80g)  
당근 1/5개(20g)  
후춧가루 적당량  
깨소금 적당량

[만드는 법]

1. 양파, 파프리카는 한입 크기로 잘라 준비한다.
2. 당근은 반으로 갈라 편 썰고 마늘은 반으로 자른다.
3. 소시지는 모양을 내어 준비한다.
4. 진간장, 황설탕, 케첩, 식초, 간 생강을 넣어 소스를 만들어 준다.
5. 팬에 기름을 두르고 마늘을 넣어 튀기듯이 볶는다.
6. 마늘이 익기 시작하면 양파, 후춧가루를 넣어 함께 볶는다.
7. 양파가 투명해지면 소시지를 넣어 칼집이 벌어질 정도까지 볶는다.
8. 파프리카, 양송이, 당근을 넣어 볶아준다.
9. 만들어 둔 소스를 넣어 볶아준다.
10. 완성 그릇에 담은 후 깨소금을 뿌려 완성한다."

두부 1모(290g)

새우젓 1큰술(20g)  
 굵은 고춧가루 2큰술(16g)  
 진간장 2큰술(20g)  
 황설탕 1/3큰술(4g)  
 정수 물 1컵(180ml)  
 간 마늘 1/2큰술(10g)  
 간 생강 약간  
 대파 1컵(60g)  
 청양고추 1개(10g)  
 들기름 1큰술(7g)

[만드는 법]

1. 대파, 청양고추를 두께 0.3cm로 송송 썬다.
2. 두부를 반으로 갈라 두께 1cm로 자른다.
3. 얇은 팬에 썰어놓은 두부를 깎는다.
4. 두부가 반쯤 잠기도록 정수물을 붓는다.
5. 새우젓, 굵은 고춧가루, 진간장, 황설탕, 간 마늘, 간 생강, 청양고추, 대파를 넣고 졸여준다.  
 (두부가 끓어오르는 양념에 살짝 잠길 수 있게 해준다.)
6. 양념장이 반 정도 졸아들면 들기름을 넣어 졸인다.
7. 양념장이 자작할 정도까지 졸으면 불을 끄고 접시에 담아 완성한다. """, ""Wn[재료]

식빵 2장  
 달걀 2개  
 양배추 1/3컵(30g)  
 대파 1/3컵(30g)  
 당근 1/2큰술(5g)  
 슬라이스햄 2장  
 슬라이스치즈 2장  
 황설탕 2큰술 (20g)  
 꽃소금 적당량  
 마가린 적당량

[만드는 법]

1. 양배추는 가늘게 채 썰고 대파는 송송 썰어 준비한다.
2. 당근은 잘게 썰어 준비한다.
3. 컵에 달걀 2개를 잘 풀어준 후 양배추, 당근, 대파, 소금을 넣어 섞는다.
4. 프라이팬에 마가린을 두르고 달걀물을 부은 후 식빵 2장을 올려 익힌다.
5. 달걀이 익으면 뒤집어 프라이팬에 마가린을 둘러 가며 식빵을 구워준다.
6. 프라이팬 한 쪽에서 슬라이스 햄을 노릇하게 구워 달걀 위에 올려준다.
7. 슬라이스 햄 위에 슬라이스 치즈를 올린다.
8. 황설탕을 치즈 위에 골고루 뿌려 준 후 반으로 접어준다.
9. 빵이 노릇해지면 완성 그릇에 옮겨 담아 반으로 자른다. """, ""Wn[재료]

된장 1/3컵(60g)  
 물 약4컵(750mL)  
 간마늘 1/2큰술(8g)  
 육수용멸치 10마리(20g)  
 애호박 1/2개(130g)  
 양파 1/2개(80g)  
 느타리버섯 1컵(80g)  
 대파 2/3컵(60g)  
 청양고추 2개(14g)  
 홍고추 1/2개(7g)  
 두부 1/2모(180g)

[만드는 법]

1. 애호박, 양파, 두부는 먹기 좋은 크기로 잘라 준비한다.
2. 느타리버섯은 찢어서 준비한다.
3. 청양고추, 홍고추, 대파는 1cm 정도 두께로 썰어 준비한다.
4. 멸치는 머리, 내장을 제거하고 3등분 정도로 찢어 준비한다.
5. 냄비에 손질한 멸치, 물, 양파를 넣어 끓인다.
6. 육수가 끓으면 느타리버섯, 애호박, 간마늘, 된장을 넣는다.
7. 된장 육수가 끓어오르면 대파, 청양고추, 홍고추를 넣어 끓인다.
8. 찌개가 끓으면 두부를 넣고 1분 정도 끓여 완성한다. """]

```
i = 0
for result in listResult:
    resultSet = ()
    resultSet.append(result)
resultSet = { "red pepper", "tofu", "garlic"}

recipeSet1 = {"sausage", "garlic", "onion", "mushroom", "pepper", "carrot"}
recipeSet2 = {"tofu", "red pepper", "garlic", "leek"}
recipeSet3 = {"bread", "egg", "cabbage", "leek", "mushroom", "carrot", "sausage"}
recipeSet4 = {"garlic", "squash", "leek", "mushroom", "onion", "red pepper", "tofu"}
recipeTotal = [list(recipeSet1), list(recipeSet2), list(recipeSet3), list(recipeSet4)]
recipeName = ["소세지 야채볶음", "두부조림", "토스트", "된장찌개"]
for key in recipeName:
```

```

recipeBook[key] = recipeTotal[i]
i += 1

inter1 = resultSet.intersection(recipeSet1)
inter2 = resultSet.intersection(recipeSet2)
inter3 = resultSet.intersection(recipeSet3)
inter4 = resultSet.intersection(recipeSet4)
possibility1 = len(inter1)
possibility2 = len(inter2)
possibility3 = len(inter3)
possibility4 = len(inter4)
possible = [possibility1, possibility2, possibility3, possibility4]
for pos in possible:
    if posMax <= pos:
        posMax = pos
    else:
        continue
count = 0
for pos in possible:
    if pos == posMax:
        recipe.append(count)
        count += 1
    else:
        count += 1
menuNum=1
print("내가 가진 재료 :", 'W033[38;2;0;215;0m' + "{}".format(list(resultSet)) + 'W033[0m')
for i in recipe:
    print("{} . 메뉴이름:".format(menuNum), 'W033[38;2;215;50;50m' + "{}".format(recipeName[i]) + 'W033[0m', ", 필요한 재료:", 'W033[38;2;135;175;255m' + "{}"
    menuNum += 1
menuChoice = int(input("몇번째 메뉴를 고르시겠습니까?: "))

print(recipeProcess[recipe[menuChoice-1]])

```

🔄 내가 가진 재료 : ['tofu', 'garlic', 'red pepper']

1. 메뉴이름: **두부조림** , 필요한 재료: ['leek', 'tofu', 'garlic', 'red pepper'] , 부족한 재료: ['leek']
2. 메뉴이름: **된장찌개** , 필요한 재료: ['leek', 'squash', 'mushroom', 'red pepper', 'onion', 'tofu', 'garlic'] , 부족한 재료: ['leek', 'onion', 'squash', 'mushroom']

몇번째 메뉴를 고르시겠습니까?: 2

[재료]

된장 1/3컵(60g)  
 물 약4컵(750mL)  
 간마늘 1/2큰술(8g)  
 육수용멸치 10마리(20g)  
 애호박 1/2개(130g)  
 양파 1/2개(80g)  
 느타리버섯 1컵(80g)  
 대파 2/3컵(60g)  
 청양고추 2개(14g)  
 홍고추 1/2개(7g)  
 두부 1/2모(180g)

[만드는 법]

1. 애호박, 양파, 두부는 먹기 좋은 크기로 잘라 준비한다.
2. 느타리버섯은 찢어서 준비한다.
3. 청양고추, 홍고추, 대파는 1cm 정도 두께로 썰어 준비한다.
4. 멸치는 머리, 내장을 제거하고 3등분 정도로 찢어 준비한다.
5. 냄비에 손질한 멸치, 물, 양파를 넣어 끓인다.
6. 육수가 끓으면 느타리버섯, 애호박, 간마늘, 된장을 넣는다.
7. 된장 육수가 끓어오르면 대파, 청양고추, 홍고추를 넣어 끓인다.
8. 찌개가 끓으면 두부를 넣고 1분 정도 끓여 완성한다.

```

def make_dish(self, recipe):
    if not self.check_condition(recipe):
        return False
    for idx in range(1, len(self.data) - 1):
        item = self.data[idx]
        if item.name in recipe.ingredients:
            if not item.reduce_stock():
                self.remove_idx_by_date(idx)

    if recipe.name in self.kinds_of_items():
        for item in self.data[1:]:
            if item.name == recipe.name:
                item.stock += 1
    else:
        self.ingredient_create(recipe.name, str(datetime.date.today()), 1)
    return True

```

...

```
def reduce_stock(self, idx):
```

```
    if not self.data[idx].reduce_stock():
        self.remove_idx_by_date(idx)
        return False
    return True

def kinds_of_items(self):
    kinds = set()
    for item in self.data[1:]:
        kinds.add(item.name)
    return kinds

def check_condition(self, recipe):
    for item in recipe.ingredients:
        if item not in self.kinds_of_items():
            return False
    return True
```



```
File "<tokenize>", line 26
    def kinds_of_items(self):
        ^
```

**IndentationError**: unindent does not match any outer indentation level