

Exemple inspiré du projet YOLOv4 Object Detection:

<https://colab.research.google.com/drive/1xdjyBiY75MAVRSjgmigI7pbRLn58VrbE?usp=sharing>

```
# import dependencies
from IPython.display import display, Javascript, Image
from google.colab.output import eval_js
from google.colab.patches import cv2_imshow
from base64 import b64decode, b64encode
import cv2
import numpy as np
import PIL
import io
import html
import time
import matplotlib.pyplot as plt
%matplotlib inline
```

## ▼ Cloning and Setting Up Darknet for YOLOv4

We will be using the famous AlexeyAB's darknet repository in this tutorial to perform YOLOv4 detections.

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

Cloning into 'darknet'...
remote: Enumerating objects: 14730, done.
remote: Total 14730 (delta 0), reused 0 (delta 0), pack-reused 14730
Receiving objects: 100% (14730/14730), 13.27 MiB | 8.19 MiB/s, done.
Resolving deltas: 100% (10020/10020), done.

# change makefile to have GPU, OPENCV and LIBSO enabled
%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/darknet

# make darknet (builds darknet so that you can then use the darknet.py file and have
!make

./src/network_network.cu(304): warning: variable 'x' was declared but never referenced
./src/network_kernels.cu: In function 'float train_network_datum_gpu(network, fl
```

```

./src/network_kernels.cu:364:7: warning: variable 'l' set but not used [-Wunused
    layer l = net.layers[net.n - 1];
    ^
nvcc -gencode arch=compute_35,code=sm_35 -gencode arch=compute_50,code=[sm_50,cc
nvcc warning : The 'compute_35', 'compute_37', 'compute_50', 'sm_35', 'sm_37' an
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -DOPENCV `pkg-config
g++ -std=c++11 -shared -std=c++11 -fvisibility=hidden -DLIB_EXPORTS -Iinclude/ -
In file included from src/yolo_v2_class.cpp:2:0:
include/yolo_v2_class.hpp: In member function 'void track_kalman_t::clear_old_st
include/yolo_v2_class.hpp:878:50: warning: comparison between signed and unsigne
    if ((result_vec_pred[state_id].x > img_size.width) ||
include/yolo_v2_class.hpp:879:50: warning: comparison between signed and unsigne
    (result_vec_pred[state_id].y > img_size.height))
include/yolo_v2_class.hpp: In member function 'track_kalman_t::tst_t track_kalma
include/yolo_v2_class.hpp:899:30: warning: comparison between signed and unsigne
    for (size_t i = 0; i < max_objects; ++i)
        ~^~~~~~
include/yolo_v2_class.hpp: In member function 'std::vector<bbox_t> track_kalman_
include/yolo_v2_class.hpp:989:30: warning: comparison between signed and unsigne
    for (size_t i = 0; i < max_objects; ++i)
        ~^~~~~~
include/yolo_v2_class.hpp: In member function 'std::vector<bbox_t> track_kalman_
include/yolo_v2_class.hpp:1024:30: warning: comparison between signed and unsign
    for (size_t i = 0; i < max_objects; ++i)
        ~^~~~~~
src/yolo_v2_class.cpp: In member function 'std::vector<bbox_t> Detector::trackin
src/yolo_v2_class.cpp:370:40: warning: comparison between signed and unsigned in
    if (prev_bbox_vec_deque.size() > frames_story) prev_bbox_vec_deque.pop_
        ~~~~~~^~~~~~
src/yolo_v2_class.cpp:385:34: warning: comparison between signed and unsigned in
    if (cur_dist < max_dist && (k.track_id == 0 || dist_vec[m]
        ~~~~~~^~~~~~
src/yolo_v2_class.cpp:409:40: warning: comparison between signed and unsigned in
    if (prev_bbox_vec_deque.size() > frames_story) prev_bbox_vec_deque.pop_
        ~~~~~~^~~~~~
g++ -std=c++11 -std=c++11 -Iinclude/ -I3rdparty/stb/include -DOPENCV `pkg-config
In file included from src/yolo_console_dll.cpp:23:0:
include/yolo_v2_class.hpp: In member function 'void track_kalman_t::clear_old_st
include/yolo_v2_class.hpp:878:50: warning: comparison between signed and unsigne
    if ((result_vec_pred[state_id].x > img_size.width) ||
include/yolo_v2_class.hpp:879:50: warning: comparison between signed and unsigne
    (result_vec_pred[state_id].y > img_size.height))
include/yolo_v2_class.hpp: In member function 'track_kalman_t::tst_t track_kalma
include/yolo_v2_class.hpp:899:30: warning: comparison between signed and unsigne
    for (size_t i = 0; i < max_objects; ++i)
        ~^~~~~~
include/yolo_v2_class.hpp: In member function 'std::vector<bbox_t> track_kalman_
include/yolo_v2_class.hpp:989:30: warning: comparison between signed and unsigne
    for (size_t i = 0; i < max_objects; ++i)
        ~^~~~~~
include/yolo_v2_class.hpp: In member function 'std::vector<bbox_t> track_kalman_
include/yolo_v2_class.hpp:1024:30: warning: comparison between signed and unsign
    for (size_t i = 0; i < max_objects; ++i)
        ~^~~~~~
src/yolo_console_dll.cpp: In function 'void draw_boxes(cv::Mat, std::vector<bbox
src/yolo_console_dll.cpp:192:46: warning: comparison between signed and unsigned
    int max width = (text size.width > i.w + 2) ? text size.width : (i.

```

```
# get bthe scaled yolov4 weights file that is pre-trained to detect 80 classes (objec
!wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&con
```

```
--2021-03-17 18:32:10-- https://docs.google.com/uc?export=download&confirm=Gfxc
Resolving docs.google.com (docs.google.com)... 108.177.97.113, 108.177.97.102, 1
Connecting to docs.google.com (docs.google.com)|108.177.97.113|:443... connected
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://doc-10-68-docs.googleusercontent.com/docs/securesc/puvukhuq599
--2021-03-17 18:32:10-- https://doc-10-68-docs.googleusercontent.com/docs/secu
Resolving doc-10-68-docs.googleusercontent.com (doc-10-68-docs.googleusercontent
Connecting to doc-10-68-docs.googleusercontent.com (doc-10-68-docs.googleusercontent
HTTP request sent, awaiting response... 302 Found
Location: https://docs.google.com/nonceSigner?nonce=44g4c6ba7hoh8&continue=https
--2021-03-17 18:32:10-- https://docs.google.com/nonceSigner?nonce=44g4c6ba7hoh8
Connecting to docs.google.com (docs.google.com)|108.177.97.113|:443... connected
HTTP request sent, awaiting response... 302 Found
Location: https://doc-10-68-docs.googleusercontent.com/docs/securesc/puvukhuq599
--2021-03-17 18:32:11-- https://doc-10-68-docs.googleusercontent.com/docs/secu
Connecting to doc-10-68-docs.googleusercontent.com (doc-10-68-docs.googleusercontent
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/octet-stream]
Saving to: 'yolov4-csp.weights'
```

```
yolov4-csp.weights      [          <=>          ] 202.13M  103MB/s  in 2.0s
```

```
2021-03-17 18:32:14 (103 MB/s) - 'yolov4-csp.weights' saved [211944840]
```

## ▼ Darknet for Python

```
# import darknet functions to perform object detections
from darknet import *
# load in our YOLOv4 architecture network
network, class_names, class_colors = load_network("cfg/yolov4-csp.cfg", "cfg/coco.dat
width = network_width(network)
height = network_height(network)

# darknet helper function to run detection on image
def darknet_helper(img, width, height):
    darknet_image = make_image(width, height, 3)
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img_resized = cv2.resize(img_rgb, (width, height),
                                interpolation=cv2.INTER_LINEAR)

    # get image ratios to convert bounding boxes to proper size
    img_height, img_width, _ = img.shape
    width_ratio = img_width/width
    height_ratio = img_height/height

    # run model on darknet style image to get detections
    copy_image from bytes(darknet_image, img_resized.tobytes())
```

```

def detect_image(network, class_names, darknet_image):
    detections = detect_image(network, class_names, darknet_image)
    free_image(darknet_image)
    return detections, width_ratio, height_ratio

```

## ▼ Helper Functions

```

# function to convert the JavaScript object into an OpenCV image
def js_to_image(js_reply):
    """
    Params:
        js_reply: JavaScript object containing image from webcam
    Returns:
        img: OpenCV BGR image
    """
    # decode base64 image
    image_bytes = b64decode(js_reply.split(',')[1])
    # convert bytes to numpy array
    jpg_as_np = np.frombuffer(image_bytes, dtype=np.uint8)
    # decode numpy array into OpenCV BGR image
    img = cv2.imdecode(jpg_as_np, flags=1)

    return img

# function to convert OpenCV Rectangle bounding box image into base64 byte string to
def bbox_to_bytes(bbox_array):
    """
    Params:
        bbox_array: Numpy array (pixels) containing rectangle to overlay on video s
    Returns:
        bytes: Base64 image byte string
    """
    # convert array into PIL image
    bbox_PIL = PIL.Image.fromarray(bbox_array, 'RGBA')
    iobuf = io.BytesIO()
    # format bbox into png for return
    bbox_PIL.save(iobuf, format='png')
    # format return string
    bbox_bytes = 'data:image/png;base64,{}'.format((str(b64encode(iobuf.getvalue()))), 'u

    return bbox_bytes

```

## ▼ YOLOv4 on Webcam Videos

```

# JavaScript to properly create our live video stream using our webcam as input
def video_stream():
    js = Javascript('')

```

```

var video;
var div = null;
var stream;
var captureCanvas;
var imgElement;
var labelElement;

var pendingResolve = null;
var shutdown = false;

function removeDom() {
    stream.getVideoTracks()[0].stop();
    video.remove();
    div.remove();
    video = null;
    div = null;
    stream = null;
    imgElement = null;
    captureCanvas = null;
    labelElement = null;
}

function onAnimationFrame() {
    if (!shutdown) {
        window.requestAnimationFrame(onAnimationFrame);
    }
    if (pendingResolve) {
        var result = "";
        if (!shutdown) {
            captureCanvas.getContext('2d').drawImage(video, 0, 0, 640, 480);
            result = captureCanvas.toDataURL('image/jpeg', 0.8)
        }
        var lp = pendingResolve;
        pendingResolve = null;
        lp(result);
    }
}

async function createDom() {
    if (div !== null) {
        return stream;
    }

    div = document.createElement('div');
    div.style.border = '2px solid black';
    div.style.padding = '3px';
    div.style.width = '100%';
    div.style.maxWidth = '600px';
    document.body.appendChild(div);

    const modelOut = document.createElement('div');
    modelOut.innerHTML = "<span>Status:</span>";

```

```

labelElement = document.createElement('span');
labelElement.innerText = 'No data';
labelElement.style.fontWeight = 'bold';
modelOut.appendChild(labelElement);
div.appendChild(modelOut);

video = document.createElement('video');
video.style.display = 'block';
video.width = div.clientWidth - 6;
video.setAttribute('playsinline', '');
video.onclick = () => { shutdown = true; };
stream = await navigator.mediaDevices.getUserMedia(
    {video: { facingMode: "environment"}});
div.appendChild(video);

imgElement = document.createElement('img');
imgElement.style.position = 'absolute';
imgElement.style.zIndex = 1;
imgElement.onclick = () => { shutdown = true; };
div.appendChild(imgElement);

const instruction = document.createElement('div');
instruction.innerHTML =
    '<span style="color: red; font-weight: bold;">' +
    'When finished, click here or on the video to stop this demo</span>';
div.appendChild(instruction);
instruction.onclick = () => { shutdown = true; };

video.srcObject = stream;
await video.play();

captureCanvas = document.createElement('canvas');
captureCanvas.width = 640; //video.videoWidth;
captureCanvas.height = 480; //video.videoHeight;
window.requestAnimationFrame(onAnimationFrame);

return stream;
}
async function stream_frame(label, imgData) {
    if (shutdown) {
        removeDom();
        shutdown = false;
        return '';
    }

    var preCreate = Date.now();
    stream = await createDom();

    var preShow = Date.now();
    if (label !== "") {
        labelElement.innerHTML = label;
    }

```

```

    }

    if (imgData != "") {
        var videoRect = video.getClientRects()[0];
        imgElement.style.top = videoRect.top + "px";
        imgElement.style.left = videoRect.left + "px";
        imgElement.style.width = videoRect.width + "px";
        imgElement.style.height = videoRect.height + "px";
        imgElement.src = imgData;
    }

    var preCapture = Date.now();
    var result = await new Promise(function(resolve, reject) {
        pendingResolve = resolve;
    });
    shutdown = false;

    return { 'create': preShow - preCreate,
            'show': preCapture - preShow,
            'capture': Date.now() - preCapture,
            'img': result };
}
''')

```

display(js)

```

def video_frame(label, bbox):
    data = eval_js('stream_frame("{}","{}").format(label, bbox)')
    return data

```

## ▼ Running on phone camera as webcam

```

# start streaming video from webcam
video_stream()
# label for video
label_html = 'Capturing...'
# initialize bounding box to empty
bbox = ''
count = 0
allowed_items = ['person', 'backpack', 'tie', 'suitcase', 'bottle', 'cup', 'bowl', 'b
while True:
    js_reply = video_frame(label_html, bbox)
    if not js_reply:
        break

# convert JS response to OpenCV Image
frame = js_to_image(js_reply["img"])

# create transparent overlay for bounding box
bbox_array = np.zeros([480,640,4], dtype=np.uint8)

```

```

# call our darknet helper on video frame
detections, width_ratio, height_ratio = darknet_helper(frame, width, height)

# loop through detections and draw them on transparent overlay image
for label, confidence, bbox in detections:
    if label not in allowed_items:
        print('Un article interdit a été détecté: ', label)
        left, top, right, bottom = bbox2points(bbox)
        left, top, right, bottom = int(left * width_ratio), int(top * height_ratio), int(right * width_ratio), int(bottom * height_ratio)
        bbox_array = cv2.rectangle(frame, (left, top), (right, bottom), class_color, 2)
        bbox_array = cv2.putText(bbox_array, "{} {:.2f}".format(label, confidence),
                                (left, top - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                                class_colors[label], 2)

bbox_array[:, :, 3] = (bbox_array.max(axis = 2) > 0 ).astype(int) * 255
# convert overlay of bbox into bytes
bbox_bytes = bbox_array[:, :, 3].tobytes()
# update bbox so next frame gets new overlay
bbox = bbox_bytes

Un article interdit a été détecté: cell phone
Un article interdit a été détecté: cell phone
Un article interdit a été détecté: cell phone
Un article interdit a été détecté: toothbrush
Un article interdit a été détecté: toothbrush
Un article interdit a été détecté: toothbrush

```