Week 8
Introduction to Deep Learning-Based Object Detection
Tuesday, February 27th 5:30 PM – 6:45 PM
DUE: Tuesday, March 5th

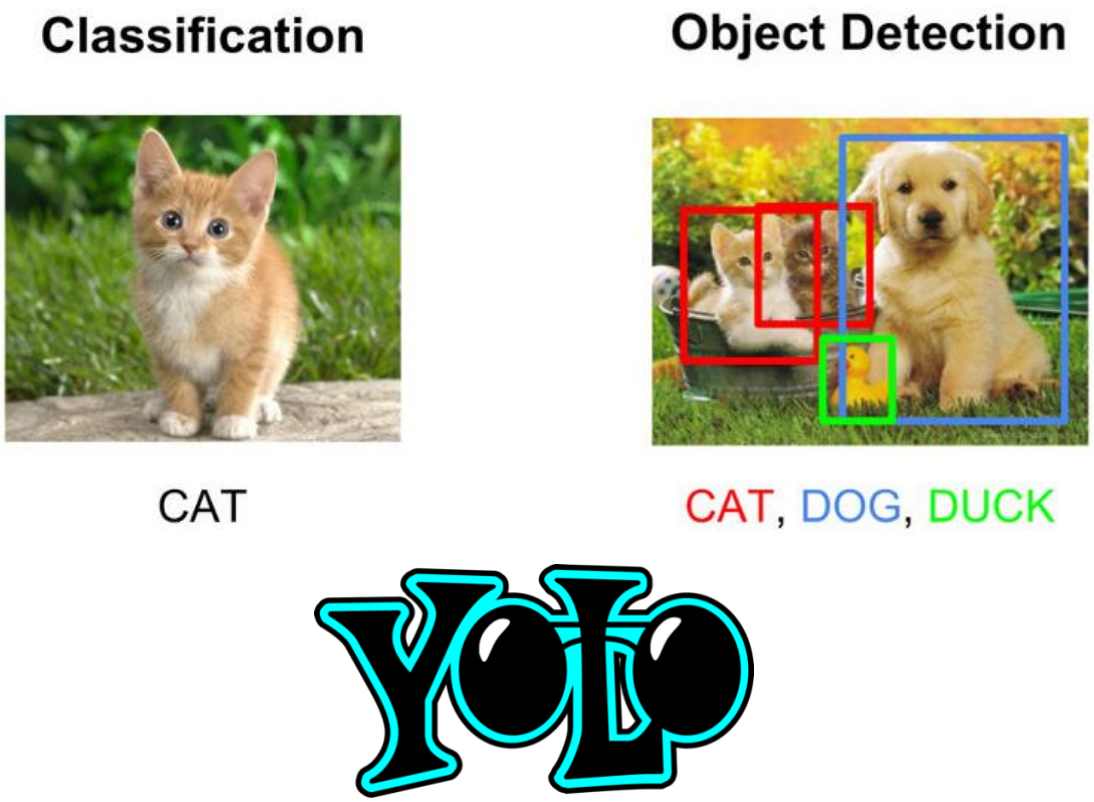

Figure 1. Image Classification vs Object Detection (https://tinyurl.com/3usmmytd, https://tinyurl.com/mrfkssee)

DESCRIPTION

As deep learning applications have gained popularity over the past decade, various different techniques have been proposed. Two common deep learning approaches that have been utilized within academia and industry are image classification and object detection. Image classification is a basic deep learning technique that helps classify individual images by assigning probabilities to different classes the image likely corresponds to. Each student in this class was introduced to image classification in Lab 4 where you trained the custom and pre-trained models on the MNIST and Fashion MNIST datasets. Although this is a popular technique that has been used for identification, image classification is unable to identify multiple instances of objects or multiple different objects per image. Additionally, image classification also lacks localization information. Therefore, in this lab, you will be introduced to another technique of deep learning called object detection. Object detection helps accurately locate and identify multiple objects within images and videos by creating bounding boxes around the objects of interest. Therefore, object detection can also aid in real-time identification tasks, for example, autonomous driving.

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

PREREQUISITES

1. Complete Labs 1, 2, 3, 4, 5, and 6

LEARNING OBJECTIVES

By the end of this lab, students will complete the following:

1. Setup the edge device by installing relevant libraries for object detection
2. Modify the YOLO object detection files
3. Perform object detection in real-time using the CSI camera
4. Perform object detection on Images
5. Homework Assignment

ASSIGNMENT

Objective 1: Setup the edge device by installing relevant libraries for object detection

1. Before you power ON the edge device, insert your CSI cameras as you had done in Lab 2
2. Power ON the edge device
3. Now launch the terminal
4. Activate the anaconda environment and navigate to the correct folder:
    a. Activate the anaconda environment used in the last three labs by typing conda activate dl and press "enter"
    b. Create a new directory called lab 7 by typing mkdir lab7 and press "enter"
    c. Navigate to the new directory by typing cd lab7 and press "enter"
5. Install the dependencies and Darknet for YOLOv4 deep learning model:
    a. Within the terminal in the lab7 directory, type sudo apt update and press "enter"
    b. Then type sudo apt upgrade and press "enter"
    c. Now type sudo apt install -y git wget build-essential gcc g++ make binutils libcanberra-gtk-module and press "enter"
    d. To export the CUDA and GPU driver, type export PATH="/usr/local/cuda-10.2/bin:$PATH" and press "enter"
    e. Now type export LD_LIBRARY_PATH="/usr/local/cuda-10.2/lib64:$LD_LIBRARY_PATH" and press "enter"
    f. Clone the YOLOv4 GitHub repository by typing git clone https://github.com/AlexeyAB/darknet.git and press enter.

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

g. Now enter the directory by typing cd darknet

Objective 2: Modify the YOLO object detection files

1. A Makefile is used for defining a set of dependencies that are utilized for training and testing the models. Open the Makefile using visual studio code (**figure 2**).

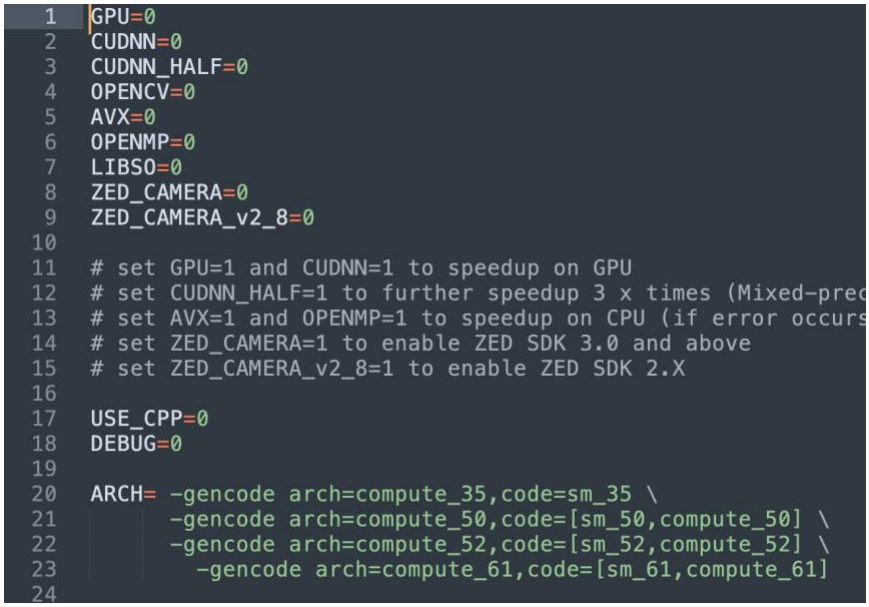

Figure 2. Original Makefile

2. You will have to change 4 lines (**figure 3**):
    a. On line 1, change GPU=0 to GPU=1 to enable GPU usage
    b. On line 2, change CUDNN=0 to CUDNN=1 to enable GPU usage
    c. On line 4, change OpenCV=0 to OpenCV=1 to use OpenCV library
    d. Change line 20 to ARCH= -gencode arch=compute_53,code=[sm_53,compute_53]
    e. Save the file

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

```
GPU=1
CUDNN=1
CUDNN_HALF=0
OPENCV=1
AVX=0
OPENMP=0
LIBSO=0
ZED_CAMERA=0
ZED_CAMERA_v2_8=0

# set GPU=1 and CUDNN=1 to speedup on GPU
# set CUDNN_HALF=1 to further speedup 3 x times (Mixed-pr
# set AVX=1 and OPENMP=1 to speedup on CPU (if error occu
# set ZED_CAMERA=1 to enable ZED SDK 3.0 and above
# set ZED_CAMERA_v2_8=1 to enable ZED SDK 2.X

USE_CPP=0
DEBUG=0

ARCH= -gencode arch=compute_53,code=[sm_53,compute_53]
```

Figure 3. Makefile after making the modifications

3. Make the file by typing make -j4 in the terminal and press "enter"
4. Obtain the YOLOv4 pre-trained weights that were trained on the benchmark COCO dataset by typing wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.weights and press "enter"
5. Open the file located in the cfg folder called yolov4.cfg (/cfg/yolov4.cfg) in Visual Studio Code.
6. Modify the width and height to 224 and 224 on lines 7 and 8, respectively (**figure 4**).

```
[net]
batch=64
subdivisions=8
# Training
#width=512
#height=512
width=608
height=608
channels=3
momentum=0.949
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1
```

Figure 4. The yolov4.cfg file

7. The YOLO model files are now ready for detection.

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

8. **Restart your edge device (You may type sudo reboot in the terminal and press "enter")**

Objective 3: Perform object detection in real-time using CSI Camera

1. Confirm that the CSI camera is working correctly by typing gst-launch-1.0 nvarguscamerasrc ! 'video/x-raw(memory:NVMM),width=3820, height=2464, framerate=21/1, format=NV12' ! nvvidconv flip-method=0 ! 'video/x-raw,width=1280, height=720' ! nvvidconv ! nvegltransform ! nveglglessink -e and press "enter"
2. Type "ctrl+c" in the terminal to end the program
3. Now type ./darknet detector demo cfg/coco.data cfg/yolov4.cfg yolov4.weights "nvarguscamerasrc ! video/x-raw(memory:NVMM),width=1280, height=720, framerate=30/1, format=NV12 ! nvvidconv flip_method=0 ! video/x-raw, format=BGRx, width=640, height=480 ! videoconvert ! video/x-raw, format=BGR ! appsink" and press "enter"
4. Type "ctrl+c" in the terminal to end the program
5. Wait for a few seconds until the object detection starts in real-time.
6. The terminal will continuously show the different objects being detected (circled in **RED**) and their confidence scores (circled in **GREEN**) as shown in **figure 5**.
7. Take a screenshot or an image of the live object detection for submission.

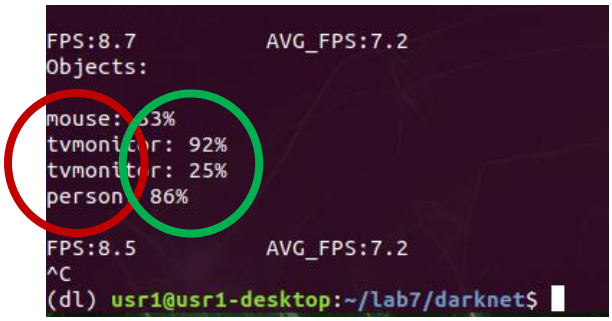

Figure 5. Output of the live object detection

Objective 4: Perform object detection on Images

1. For running the detection on images, type ./darknet detect cfg/yolov4.cfg yolov4.weights data/dog.jpg and press "enter" as shown in **figure 6**.

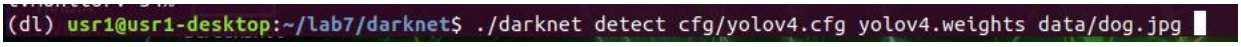

Figure 6. Running the object detection on images

2. The output is shown in **figure 7**.

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

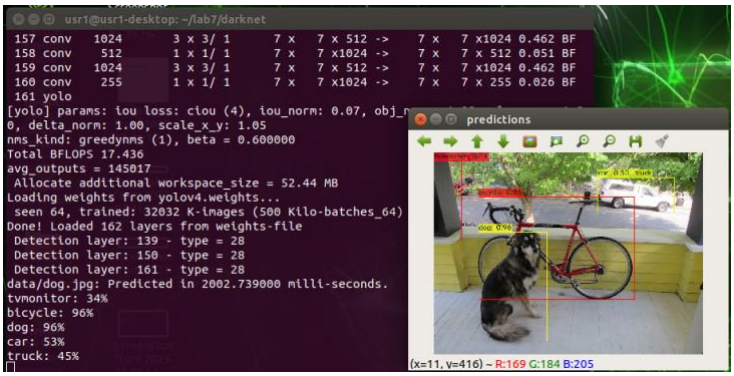

Figure 7. Output of running the object detection on images

Objective 5: Homework Tasks

1. Display the object detection information onto your web applications from Lab 6:
    a. Launch the web application created in lab 6
    b. Create a new page on the web application from lab 6 called “Object Detection”.
    c. From the main.py file used in lab 6, modify the web application task = st.sidebar.selectbox() function to add an additional page called “Object Detection”
    d. You will also have to modify the pages() function by adding the line elif task == “Object Detection” at the end of the function in the same way the “Deep Learning”, “Mapping” and “Homework” pages were created.
    e. Obtain 5 images of everyday objects from Google.
    f. Feed each image to the object detection model as you did in Objective 4 (**figure 6**).
    g. Save the output of the image (you may take a screenshot)
    h. Display each image with the detections onto the web application page using the st.image() function.
    i. Submit:
        i. The main.py file from lab 6 after adding the “Object Detection” page
        ii. Upload the screenshot / image of the live detection from objective 3
        iii. Upload the 5 images after performing object detection on them

REFERENCES / ADDITIONAL RESOURCES

1. Reference: https://medium.com/@thundo/yolov4-on-jetson-nano-672c1d38aed2

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)