

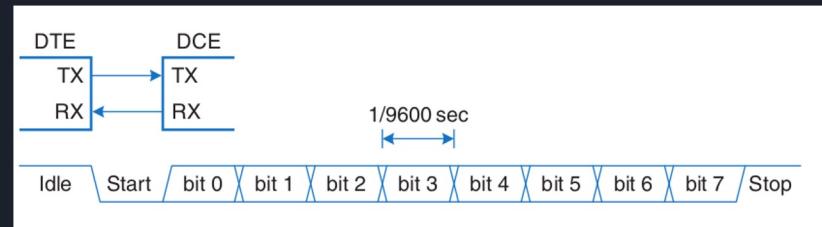


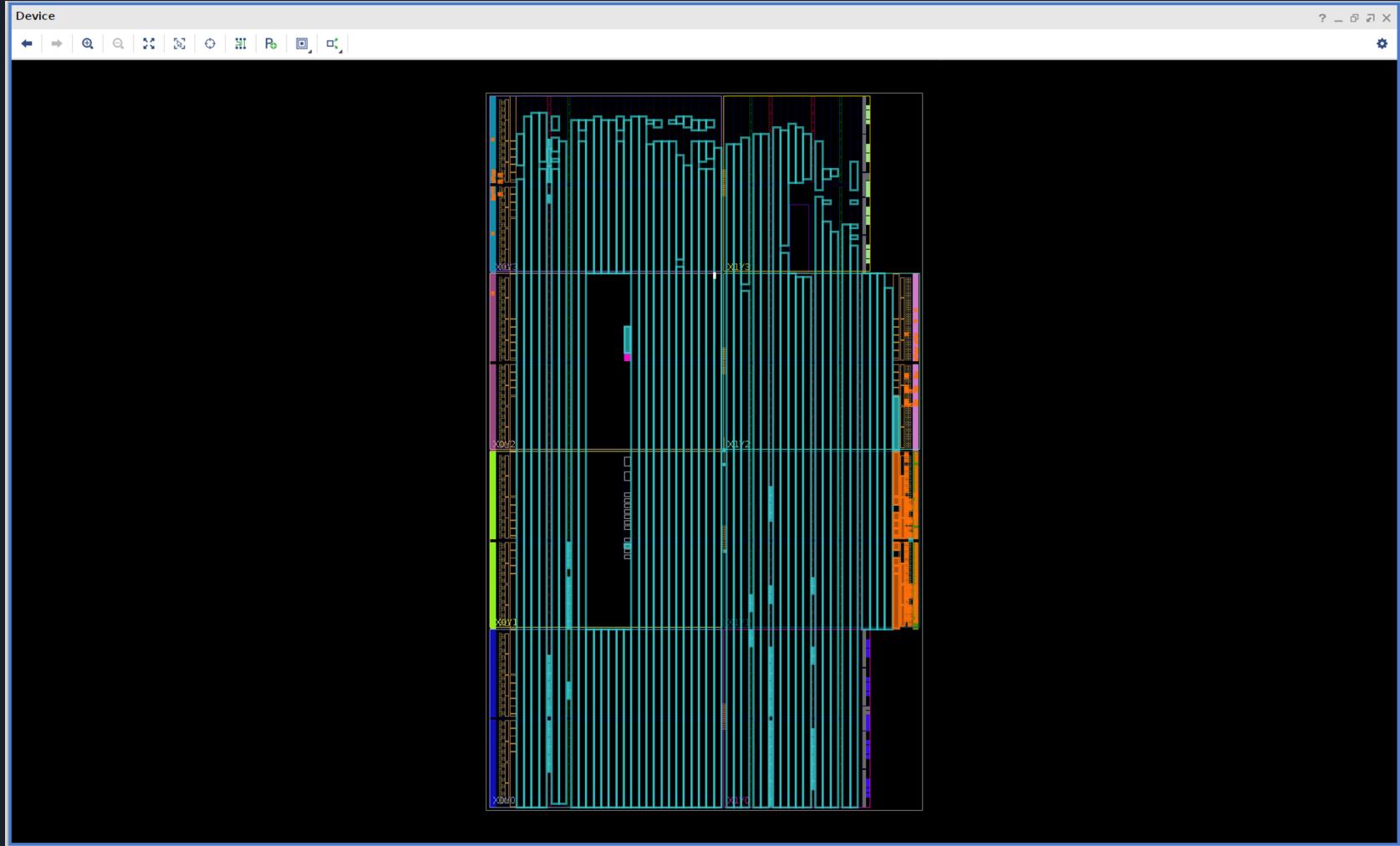
# ECE 4300 Project: Object Detection

Group B:  
Ryan Agatep, Jose Hernandez, Daniel Pedro, Gloria Tovar

# Vivado RISC-V

- Loading RISC-V architecture onto Digilent Nexys A7 100t hardware
  - Communication had to be through serial connections: UART
  - Tried in Ubuntu 22.04 on VM and WSL2
  - Software used was AMD/Xilinx Vitis, which includes Vivado
  - Had to install OS image onto SD, then boot in Vivado







## Summary

Settings

Summary (1.276)

Power Supply

## Utilization Details

Hierarchical (1)

Clocks (0.068)

## Signals (0.07)

Data (0.0)

Clock Enab.

Set/Reset

Logic (0.052)

BRAM (0.034)

DSP (0.006 W)

Clock Manage

PHASER (0.08)

I/O (0.526 W)

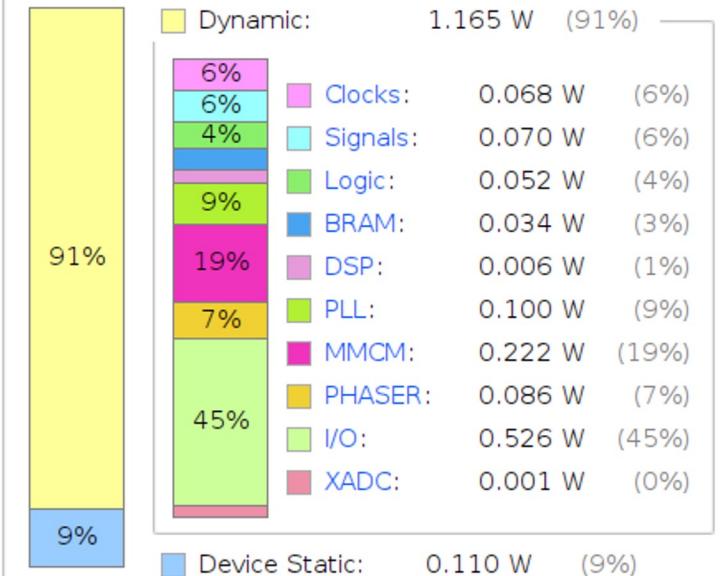
XADC (0.001)

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

<b>Total On-Chip Power:</b>	<b>1.276 W</b>
<b>Design Power Budget:</b>	<b>Not Specified</b>
<b>Process:</b>	<b>typical</b>
<b>Power Budget Margin:</b>	<b>N/A</b>
<b>Junction Temperature:</b>	<b>30.8°C</b>
Thermal Margin:	54.2°C (11.8 W)
Ambient Temperature:	25.0 °C
Effective θJA:	4.6°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	<b>Low</b>

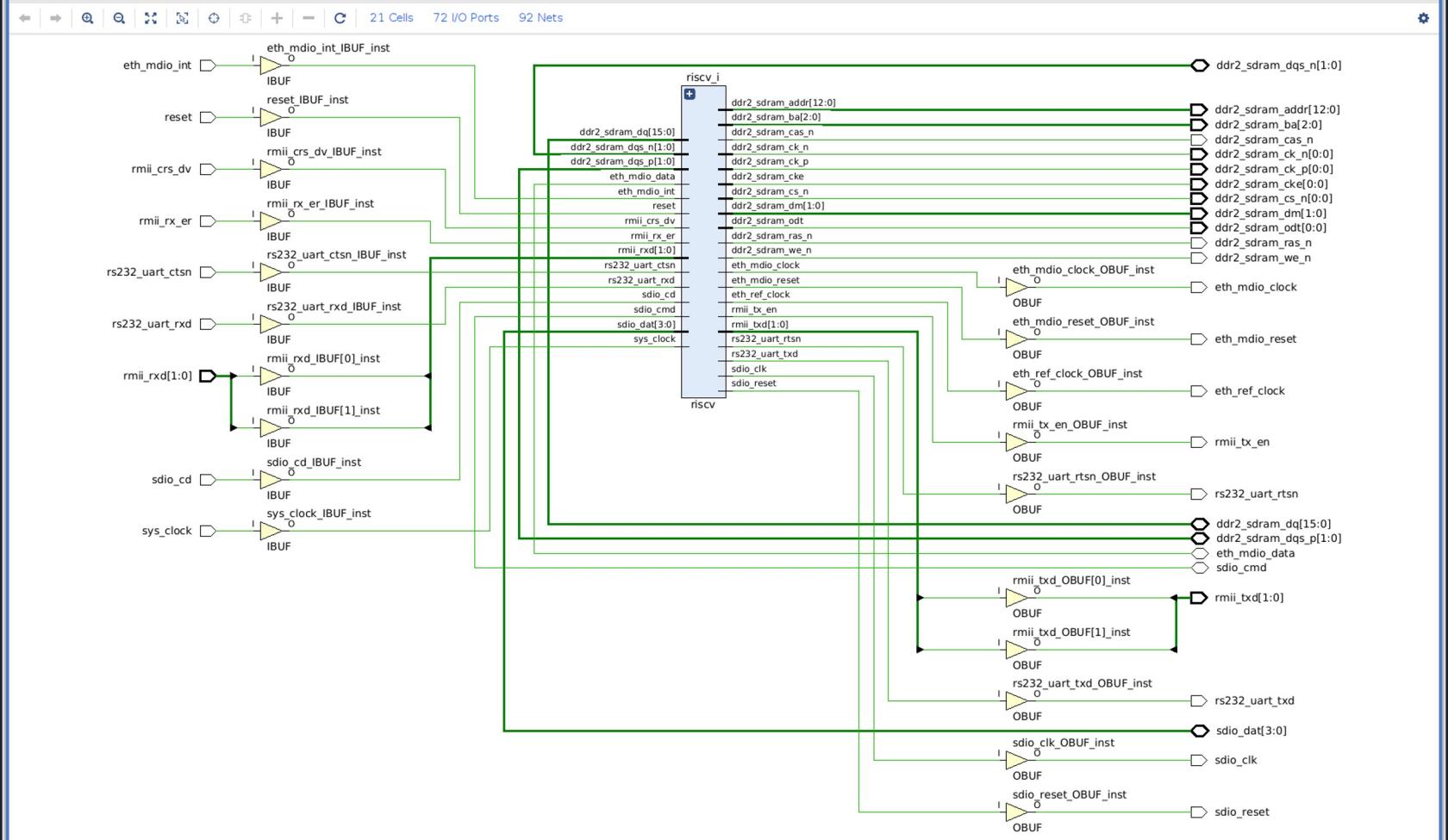
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

## On-Chip Power



impl\_1 (saved)

## Schematic



# Emulation (Qemu)

- RISC-V was emulated (same image as the one loaded onto FPGA)
  - Worked, but limited resources (storage, data)
  - Had to install GCC, basic OS toolchains

```
[...]
[ 8] Started rpc-statd-notify.svc - Notify service for a restart.
[ 9] Started serial-ptytysvc.service - Serial Getty on ttyS0.
[ 9] Started chrony.service - chrony, an NTP Client/Server.
[ 9] Started autofs.service - System Time Synchronization.
[ 9] Started ntpdate-sync-tst - System Time Synchronization.
[ 9] Started dmsgd.service - Dmsgd All-time-redacted Check for All Timer.
[ 9] Started fattr.timer - Discard unused blocks once a week.
[ 9] Started dhcpcd@.service - dhcpcd daemon for interface eth0.
[ 9] Reached target timers.target - Timer units.
[ 9] Reached target graphical.target - Graphical Interface.
[ 9] Reached target default.target - Default Unit.
[ 9] Finished system-update-ut - Record Runlevel Change in UTMP.
[ 9] Finished system-update-ut - Record Runlevel Change in UTMP.

debian:~# lsb_release -a
No LSB modules are available.

debian:~# uname -a
Linux debian 5.10.0-1135-riscv64 #1 SMP Mon Oct 16 11:45:18 PDT 2023 riscv64 GNU/Linux

debjan:~# cat /etc/lsb-release
DISTRIB_ID=Debian
DISTRIB_RELEASE=11.3 (bullseye)
DISTRIB_CODENAME=bullseye
DISTRIB_DESCRIPTION="Debian GNU/Linux 11.3 (bullseye)"

debjan:~# cat /etc/riscv64/lsb-release
DISTRIB_ID=Debian
DISTRIB_RELEASE=11.3 (bullseye)
DISTRIB_CODENAME=bullseye
DISTRIB_DESCRIPTION="Debian GNU/Linux 11.3 (bullseye)"

debjan:~# cat /etc/riscv64/debian-release
Debian 11.3 (bullseye) - riscv64
Copyright (C) 2023 The RISC-V Foundation, Inc.
This software is released under the terms of the GNU General Public License version 2 or later.
You can redistribute it and/or modify it under the terms of the GNU General Public License version 2 or later, or the terms of the Apache License, Version 2.0.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the GNU General Public License for more details.
You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

debjan:~# ls -l /etc/riscv64/debian-release
-rw-r--r-- 1 root root 1024 Oct 16 11:45 /etc/riscv64/debian-release

debjan:~# cat /etc/riscv64/debian-release | head -n 1
Debian 11.3 (bullseye) - riscv64

debjan:~# cat /etc/riscv64/debian-release | tail -n 1
Copyright (C) 2023 The RISC-V Foundation, Inc.

debjan:~# cat /etc/riscv64/debian-release | grep -i copyright
Copyright (C) 2023 The RISC-V Foundation, Inc.

debjan:~# cat /etc/riscv64/debian-release | grep -i license
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the GNU General Public License for more details.

debjan:~# cat /etc/riscv64/debian-release | grep -i gpl
You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

debjan:~# cat /etc/riscv64/debian-release | grep -i apache
This program is released under the terms of the Apache License, Version 2.0.
You may obtain a copy of the Apache License at <http://www.apache.org/licenses/LICENSE-2.0>.

debjan:~# cat /etc/riscv64/debian-release | grep -i gpl OR apache
This program is released under the terms of the Apache License, Version 2.0.
You may obtain a copy of the Apache License at <http://www.apache.org/licenses/LICENSE-2.0>.

debjan:~# cat /etc/riscv64/debian-release | grep -i gpl OR apache | head -n 1
This program is released under the terms of the Apache License, Version 2.0.
```

```
u-installation-0@think:~/Development/qemu_emulation/risc-v/b64
```

```
tclsh8.6      timedatectl      tset
tcpdump       timeout        tsort
tee           times          tty
telinit       tipc           tune2fs
telnet        tkconch3      twist3
tempfile      tload          twistd3
test          tmux           type
then          tnftp          typeset
thin_check    toe            tzselect
thin_delta   top             top

[0[          0.0%] Tasks: 22, 26 thr, 80 kthr; 1 running
[1[          0.0%] Load average: 0.04 0.02 0.02
[2[          0.0%] Uptime: 00:35:40
[3**         5.1%]
Mem[||||#*@@@@@@@@@ 137M/1.91G]
Swp[          0K/0K]
```

```
[Main] [I/O]
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2605 ubuntu 20 0 6540 2688 2176 R 4.4 0.1 0:00.95 htop
  1 root 20 0 160M 11424 6688 S 0.0 0.6 0:12.55 /sbin/init
  291 root 19 -1 44220 9860 8964 S 0.0 0.5 0:02.09 /lib/systemd/
  329 root RT 0 407M 24448 5760 S 0.0 1.2 0:02.45 /sbin/multipa
  334 root 20 0 407M 24448 5760 S 0.0 1.2 0:00.00 /sbin/multipa
  335 root RT 0 407M 24448 5760 S 0.0 1.2 0:00.00 /sbin/multipa
  336 root RT 0 407M 24448 5760 S 0.0 1.2 0:00.00 /sbin/multipa
  337 root RT 0 407M 24448 5760 S 0.0 1.2 0:00.07 /sbin/multipa
  338 root RT 0 407M 24448 5760 S 0.0 1.2 0:01.38 /sbin/multipa
  339 root RT 0 407M 24448 5760 S 0.0 1.2 0:00.00 /sbin/multipa
  340 root 20 0 23300 5348 3556 S 0.0 0.3 0:01.45 /lib/systemd/
  430 systemd-re 20 0 15016 9856 7808 S 0.0 0.5 0:01.35 /lib/systemd/
  435 systemd-ti 20 0 84928 6016 5248 S 0.0 0.3 0:00.85 /lib/systemd/
ubuntu@ubuntu:~$ df
Filesystem 1K-blocks Used Available Use% Mounted on
tmpfs     200192     820 199372  1% /run
/dev/vda1  4382908 3042776 1323748 70% /
tmpfs     1000960     0 1000960  0% /dev/shm
tmpfs      5120      0  5120  0% /run/lock
/dev/vda15  106858      2 106857  1% /boot/efi
tmpfs     200192      4 200188  1% /run/user/1000
ubuntu@ubuntu:~$ uname -a
Linux ubuntu 6.2.0-19-generic #19.1-Ubuntu SMP Fri Mar 31 12:41:53 UTC 2023 riscv64 riscv64 riscv64 GNU/Linux
ubuntu@ubuntu:~$ python3
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello world')
Hello world
>>> 
```

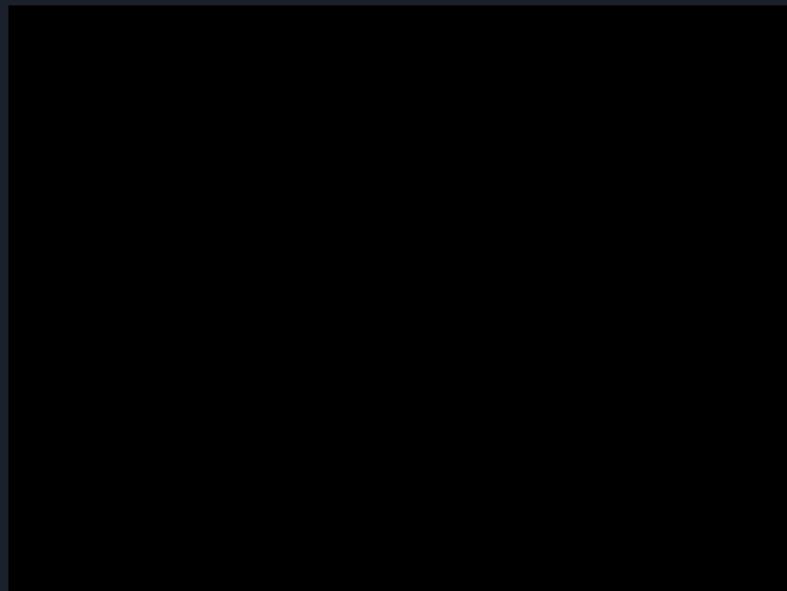
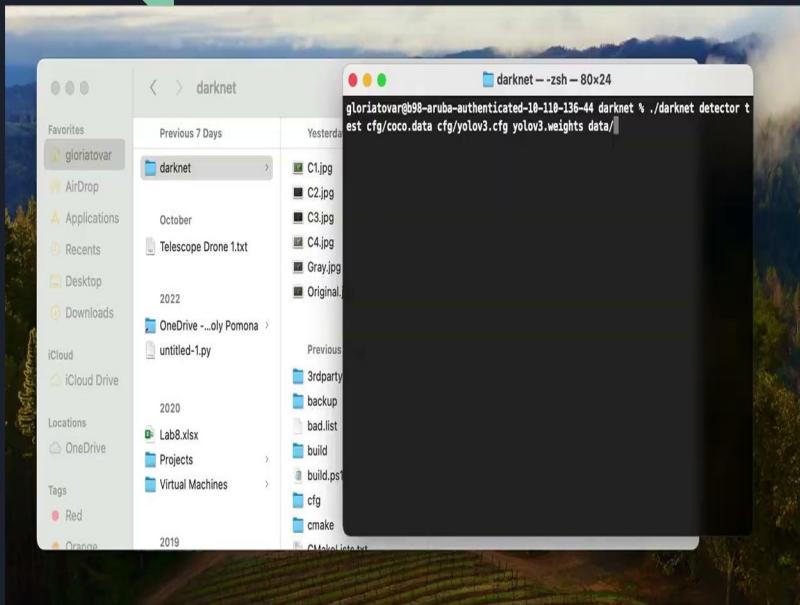
# YOLO

- Specifically used YOLOv3
- Darknet
  - Open source neural network framework written in C
  - Helps to support CPU and GPU computation
  - What's in the Darknet folder?
    - Prediction.jpg to include the results YOLO has provided
    - Data folder includes all images that were uploaded
    - Yolo weights
- First tried on Mac Terminal
- Then tried on Ubuntu 23.10.01

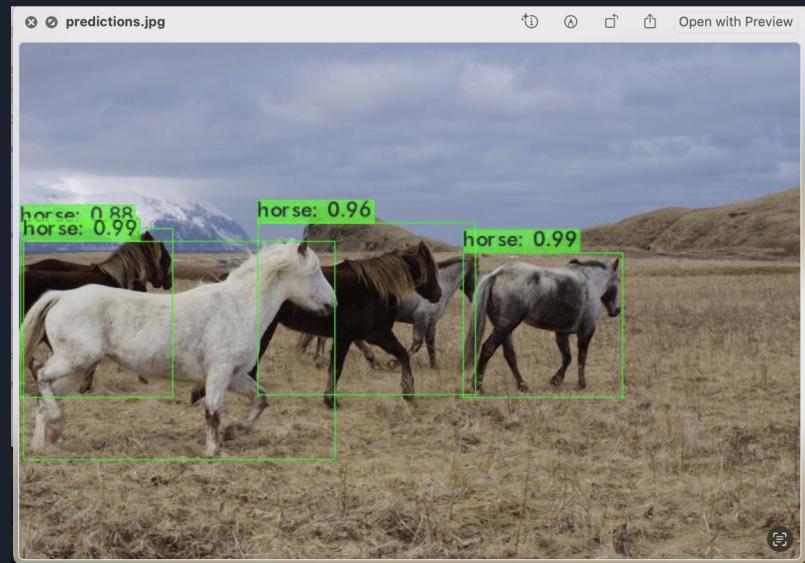
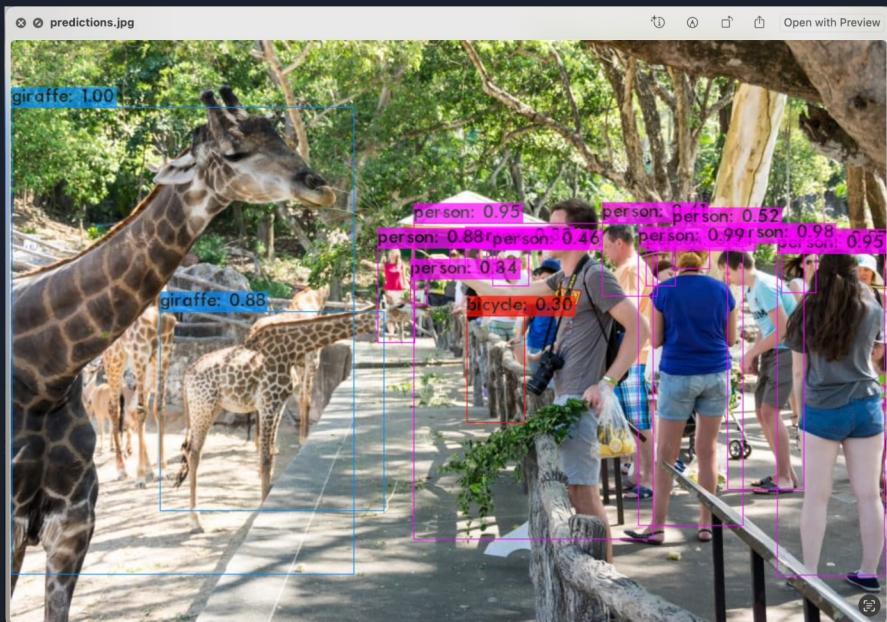
```
gloriatovar@b98-aruba-authenticated-10-110-136-44 darknet % ls
3rdparty           darknet          predictions.jpg
CMakeLists.txt     darknet.py       results
DarknetConfig.cmake.in darknet_images.py    scripts
Dockerfile.cpu      darknet_video.py   src
Dockerfile.gpu     data
LICENSE            docker-compose.yml vcpkg.json
Makefile           image_yolov3.sh  vcpkg.json.opencv23
README.md          image_yolov4.sh  video_yolov3.sh
backup             include
bad.list           json_mjpeg_streams.sh  yolov3.weights
build              net_cam_y3.sh    yolov3.weights.1
build.ps1          net_cam_v4.sh   yolov3.weights.2
cfg                obj
cmake              package.xml    yolov3.weights.3
gloriatovar@b98-aruba-authenticated-10-110-136-44 darknet %
```

```
gloriatovar@b98-aruba-authenticated-10-110-136-44 darknet % cd data
gloriatovar@b98-aruba-authenticated-10-110-136-44 data % ls
9k.tree           imagenet_shortnames.list
cat.jpg           labels
coco.names        openimages.names
coco9k.map        person.jpg
dog.jpg           scream.jpg
eagle.jpg         strawberries.jpg
giraffe.jpg       vegetables.jpg
goal.txt          voc.names
horses.jpg        zoo.jpg
imagenet.labels.list
gloriatovar@b98-aruba-authenticated-10-110-136-44 data %
```

# MAC Demo



# YOLO Results in MAC



# Ubuntu Demo in Mac Computer



Ubuntu terminal window showing the command to load weights from a file named yellow3.weights.

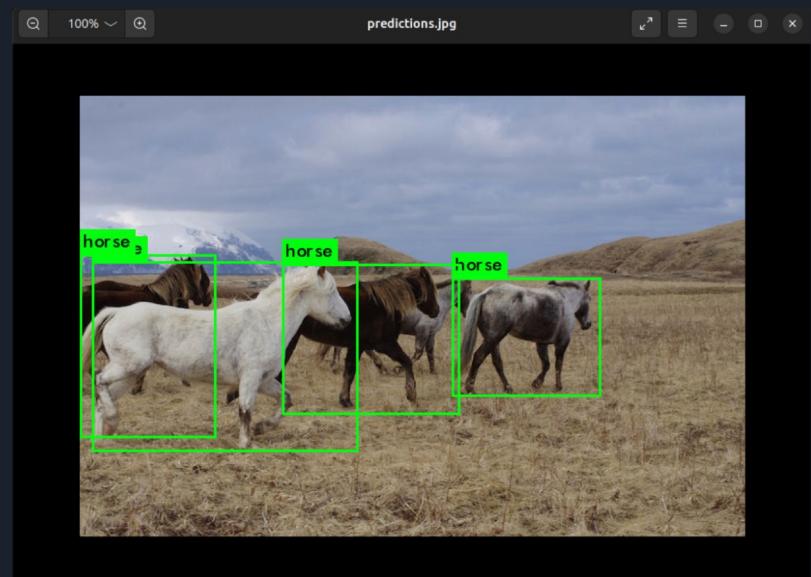
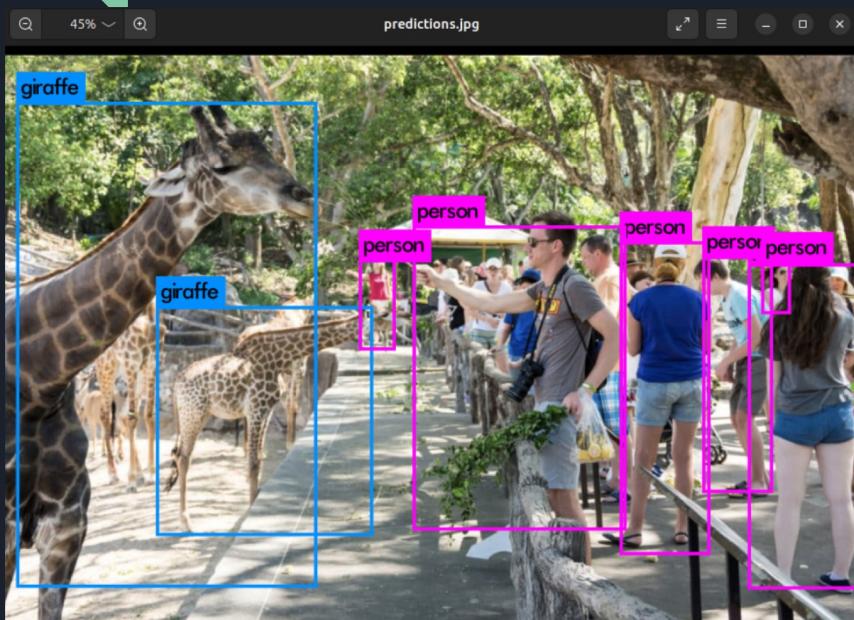
```
gloria@gloria-MacBook-Pro:~/Desktop$ python3 predict.py --model yellow3 --weights yellow3.weights --image test.jpg --output output.jpg
[...]
95 conv 320 3 x 1, 1, 1 76 x 76 x 384 -> 76 x 76 x
128 8,548 BPLOPs
160 conv 256 3 x 3, 2, 2 76 x 76 x 128 -> 76 x 76 x
256 3,487 BPLOPs
180 conv 320 3 x 1, 1, 1 76 x 76 x 256 -> 76 x 76 x
128 8,379 BPLOPs
192 conv 256 3 x 3, 2, 2 76 x 76 x 128 -> 76 x 76 x
256 3,487 BPLOPs
183 conv 320 3 x 1, 1, 1 76 x 76 x 256 -> 76 x 76 x
128 8,379 BPLOPs
184 conv 256 3 x 3, 2, 2 76 x 76 x 128 -> 76 x 76 x
256 3,487 BPLOPs
185 conv 256 3 x 1, 1, 1 76 x 76 x 256 -> 76 x 76 x
255 8,754 BPLOPs
186 yolo
>Loading weights from yellow3.weights...Done!
```



Ubuntu terminal window showing the command to load weights from a file named yellow3.weights.

```
gloria@gloria-MacBook-Pro:~/Desktop$ python3 predict.py --model yellow3 --weights yellow3.weights --image test.jpg --output output.jpg
[...]
95 conv 320 3 x 1, 1, 1 76 x 76 x 384 -> 76 x 76 x
128 8,548 BPLOPs
160 conv 256 3 x 3, 2, 2 76 x 76 x 128 -> 76 x 76 x
256 3,487 BPLOPs
180 conv 320 3 x 1, 1, 1 76 x 76 x 256 -> 76 x 76 x
128 8,379 BPLOPs
192 conv 256 3 x 3, 2, 2 76 x 76 x 128 -> 76 x 76 x
256 3,487 BPLOPs
183 conv 320 3 x 1, 1, 1 76 x 76 x 256 -> 76 x 76 x
128 8,379 BPLOPs
184 conv 256 3 x 3, 2, 2 76 x 76 x 128 -> 76 x 76 x
256 3,487 BPLOPs
185 conv 256 3 x 1, 1, 1 76 x 76 x 256 -> 76 x 76 x
255 8,754 BPLOPs
186 yolo
>Loading weights from yellow3.weights...Done!
```

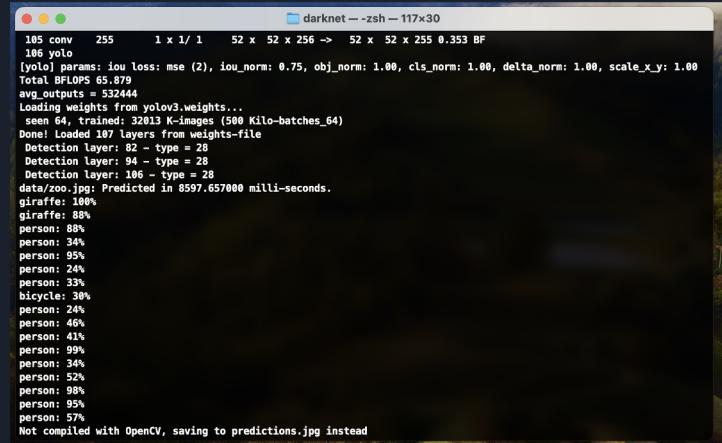
# YOLO Results in Ubuntu



# Performance (Execution Time) on MAC:

- Execution Time on Mac
  - To get results, ran ./darknet detect cfg/yolov3-tiny.cfg yolov3-tiny.weights data/**insert the name of jpg**
  - For zoo.jpg,
    - It took around 8597.657 ms
  - For horses.jpg
    - It took around 8515.933 ms

```
darknet --zsh -- 117x30
94 yolo
[yolo] params: iou_loss: mse (2), iou_norm: 0.75, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.00
95 route 91      1 x 1/ 1      26 x  26 x 256 -> 26 x  26 x 256
96 conv 128      1 x 1/ 1      26 x  26 x 256 -> 26 x  26 x 128  0.044 BF
97 yolo2nogap 96      2x      26 x  26 x 128 -> 52 x  52 x 128
98 route 97 36      2x      52 x  52 x 128 -> 52 x  52 x 384
99 conv 128      1 x 1/ 1      52 x  52 x 384 -> 52 x  52 x 128  0.266 BF
100 conv 256      3 x 3/ 1      52 x  52 x 128 -> 52 x  52 x 256  1.595 BF
101 conv 128      1 x 1/ 1      52 x  52 x 256 -> 52 x  52 x 128  0.177 BF
102 conv 256      3 x 3/ 1      52 x  52 x 128 -> 52 x  52 x 256  1.595 BF
103 conv 128      1 x 1/ 1      52 x  52 x 256 -> 52 x  52 x 128  0.177 BF
104 conv 256      3 x 3/ 1      52 x  52 x 128 -> 52 x  52 x 256  1.595 BF
105 conv 255      1 x 1/ 1      52 x  52 x 256 -> 52 x  52 x 255  0.353 BF
106 yolo
[yolo] params: iou_loss: mse (2), iou_norm: 0.75, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.00
Total BFLOPS: 68.579
avg_bflops = 68.579
avg_outputs = 532444
Loading weights from yolov3.weights...
seen 64, trained: 32013 K-Images (500 Kilo-batches_64)
Done! Loaded 107 layers from weights-file
Detection layer: 82 - type = 28
Detection layer: 94 - type = 28
Detection layer: 106 - type = 28
data/zoo.jpg: Predicted in 8597.657000 milli-seconds.
giraffe: 100%
giraffe: 88%
person: 88%
person: 34%
person: 95%
person: 24%
person: 33%
bicycle: 38%
person: 24%
person: 46%
person: 41%
person: 99%
person: 34%
person: 52%
person: 98%
person: 95%
person: 37%
Not compiled with OpenCV, saving to predictions.jpg instead
gloriatoverg090-aruba-authenticated-10-110-136-44 darknet % ||
```



```
darknet --zsh -- 117x30
105 conv 255      1 x 1/ 1      52 x  52 x 255 -> 52 x  52 x 255  0.353 BF
106 yolo
[yolo] params: iou_loss: mse (2), iou_norm: 0.75, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.00
Total BFLOPS: 65.879
avg_bflops = 532444
Loading weights from yolov3.weights...
seen 64, trained: 32013 K-Images (500 Kilo-batches_64)
Done! Loaded 107 layers from weights-file
Detection layer: 82 - type = 28
Detection layer: 94 - type = 28
Detection layer: 106 - type = 28
data/zoo.jpg: Predicted in 8597.657000 milli-seconds.
giraffe: 100%
giraffe: 88%
person: 88%
person: 34%
person: 95%
person: 24%
person: 33%
bicycle: 38%
person: 24%
person: 46%
person: 41%
person: 99%
person: 34%
person: 52%
person: 98%
person: 95%
person: 37%
Not compiled with OpenCV, saving to predictions.jpg instead
gloriatoverg090-aruba-authenticated-10-110-136-44 darknet % ||
```

# Performance (Execution Time) on Ubuntu:

- Running Time on Ubuntu
  - To get results, ran `./darknet detect cfg/yolov3-tiny.cfg yolov3-tiny.weights data/insert the name of jpg`
  - For `zoo.jpg`,
    - It took around 29.19 seconds
  - For `horses.jpg`
    - It took around 30.18 seconds

```
gloria@gloria-None:~/darknet [Q] [≡] [-] [□] [x]
104 conv 256 3 x 3 / 1    76 x 76 x 128 -> 76 x 76 x
256 3.407 BFLOPs
105 conv 255 1 x 1 / 1    76 x 76 x 256 -> 76 x 76 x
255 0.754 BFLOPs
106 yolo
Loading weights from yolov3.weights...Done!
data/zoo.jpg: Predicted in 29.198622 seconds.
giraffe: 100%
giraffe: 99%
person: 100%
person: 99%
person: 99%
person: 97%
person: 79%
person: 59%
gloria@gloria-None:~/darknet$ ./darknet detect cfg/yolov3.cfg yolov3.weights data/horses.jpg
```

```
gloria@gloria-None:~/darknet [Q] [≡] [-] [□] [x]
128 0.379 BFLOPs
102 conv 256 3 x 3 / 1    76 x 76 x 128 -> 76 x 76 x
256 3.407 BFLOPs
103 conv 128 1 x 1 / 1    76 x 76 x 256 -> 76 x 76 x
128 0.379 BFLOPs
104 conv 256 3 x 3 / 1    76 x 76 x 128 -> 76 x 76 x
256 3.407 BFLOPs
105 conv 255 1 x 1 / 1    76 x 76 x 256 -> 76 x 76 x
255 0.754 BFLOPs
106 yolo
Loading weights from yolov3.weights...Done!
data/horses.jpg: Predicted in 30.183500 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
gloria@gloria-None:~/darknet$
```

# Performance (Execution Time) on Windows Ubuntu WSL2:

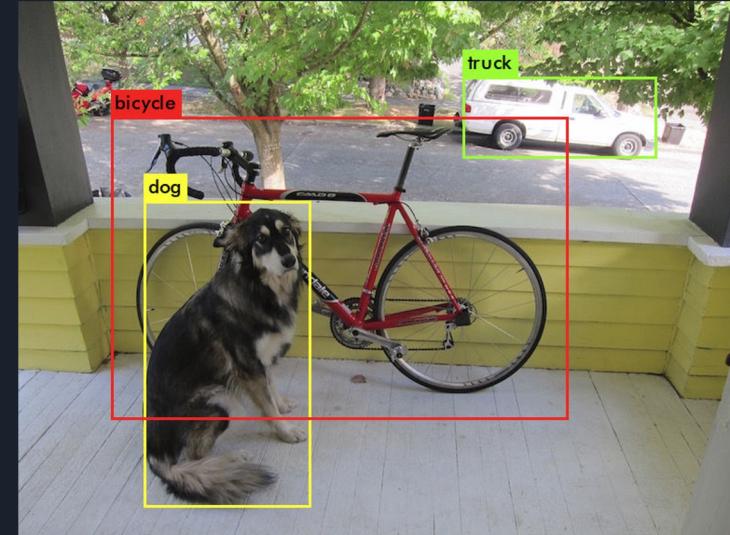
- From the previous slides on the images (on Windows),
  - To get results, ran `./darknet detect cfg/yolov3-tiny.cfg yolov3-tiny.weights data/insert the name of jpg`
  - For `zoo.jpg`,
    - It took **16.650312 s**
  - For `horses.jpg`,
    - It took **16.917211 s**

```
ryaga@LAPTOP-H800E35N: ~/darknet
85 upsample      2x    19 x  19 x 256 ->   38 x  38 x 256
86 route  85 61
87 conv   256 1 x 1 / 1   38 x  38 x 768 ->   38 x  38 x 256  0.568 BFLOPs
88 conv   512 3 x 3 / 1   38 x  38 x 256 ->   38 x  38 x 512  3.407 BFLOPs
89 conv   256 1 x 1 / 1   38 x  38 x 512 ->   38 x  38 x 256  0.379 BFLOPs
90 conv   512 3 x 3 / 1   38 x  38 x 256 ->   38 x  38 x 512  3.407 BFLOPs
91 conv   256 1 x 1 / 1   38 x  38 x 512 ->   38 x  38 x 256  0.379 BFLOPs
92 conv   512 3 x 3 / 1   38 x  38 x 256 ->   38 x  38 x 512  3.407 BFLOPs
93 conv   255 1 x 1 / 1   38 x  38 x 512 ->   38 x  38 x 255  0.377 BFLOPs
94 yolo
95 route  91
96 conv   128 1 x 1 / 1   38 x  38 x 256 ->   38 x  38 x 128  0.095 BFLOPs
97 upsample      2x    38 x  38 x 128 ->   76 x  76 x 128
98 route  97 36
99 conv   128 1 x 1 / 1   76 x  76 x 384 ->   76 x  76 x 128  0.568 BFLOPs
100 conv   256 3 x 3 / 1   76 x  76 x 128 ->   76 x  76 x 256  3.407 BFLOPs
101 conv   128 1 x 1 / 1   76 x  76 x 256 ->   76 x  76 x 128  0.379 BFLOPs
102 conv   256 3 x 3 / 1   76 x  76 x 128 ->   76 x  76 x 256  3.407 BFLOPs
103 conv   128 1 x 1 / 1   76 x  76 x 256 ->   76 x  76 x 128  0.379 BFLOPs
104 conv   256 3 x 3 / 1   76 x  76 x 128 ->   76 x  76 x 256  3.407 BFLOPs
105 conv   255 1 x 1 / 1   76 x  76 x 256 ->   76 x  76 x 255  0.754 BFLOPs
106 yolo
Loading weights from yolov3.weights...Done!
Enter Image Path: data/zoo.jpg
data/zoo.jpg: Predicted in 16.650312 seconds.
giraffe: 100%
giraffe: 99%
person: 100%
person: 100%
person: 99%
person: 98%
person: 85%
person: 65%
person: 55%
106 yolo
Loading weights from yolov3.weights...Done!
Enter Image Path: data/horses.jpg
data/horses.jpg: Predicted in 16.917211 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
```

```
ryaga@LAPTOP-H800E35N: ~/darknet
90 conv   512 3 x 3 / 1   38 x  38 x 256 ->   38 x  38 x 512  3.407 BFLOPs
91 conv   256 1 x 1 / 1   38 x  38 x 512 ->   38 x  38 x 256  0.379 BFLOPs
92 conv   512 3 x 3 / 1   38 x  38 x 256 ->   38 x  38 x 512  3.407 BFLOPs
93 conv   255 1 x 1 / 1   38 x  38 x 512 ->   38 x  38 x 255  0.377 BFLOPs
94 yolo
95 route  91
96 conv   128 1 x 1 / 1   38 x  38 x 256 ->   38 x  38 x 128  0.095 BFLOPs
97 upsample      2x    38 x  38 x 128 ->   76 x  76 x 128
98 route  97 36
99 conv   128 1 x 1 / 1   76 x  76 x 384 ->   76 x  76 x 128  0.568 BFLOPs
100 conv   256 3 x 3 / 1   76 x  76 x 128 ->   76 x  76 x 256  3.407 BFLOPs
101 conv   128 1 x 1 / 1   76 x  76 x 256 ->   76 x  76 x 128  0.379 BFLOPs
102 conv   256 3 x 3 / 1   76 x  76 x 128 ->   76 x  76 x 256  3.407 BFLOPs
103 conv   128 1 x 1 / 1   76 x  76 x 256 ->   76 x  76 x 128  0.379 BFLOPs
104 conv   256 3 x 3 / 1   76 x  76 x 128 ->   76 x  76 x 256  3.407 BFLOPs
105 conv   255 1 x 1 / 1   76 x  76 x 256 ->   76 x  76 x 255  0.754 BFLOPs
106 yolo
Loading weights from yolov3.weights...Done!
Enter Image Path: data/zoo.jpg
data/zoo.jpg: Predicted in 16.650312 seconds.
giraffe: 100%
giraffe: 99%
person: 100%
person: 100%
person: 99%
person: 98%
person: 85%
person: 65%
person: 55%
106 yolo
Loading weights from yolov3.weights...Done!
Enter Image Path: data/horses.jpg
data/horses.jpg: Predicted in 16.917211 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
```

# Performance Execution Time (x86):

- 12th Gen Intel(R) Core(TM) i7-12800HX
- Ran “./darknet detect cfg/yolov3.cfg yolov3.weights data/dog.jpg” several times
- Average: 11.905seconds
- Median: ~11.86
- Mode: N/A always <12s



# Performance Execution Time (x86) on V.M. running Ubuntu:

- AMD Ryzen 9 5900HS
- Ran “./darknet detect cfg/yolov3.cfg yolov3.weights data/dog.jpg” 5 times
- Ran “./darknet detect cfg/yolov3.cfg yolov3.weights data/horses.jpg” 5 times
- Average: 12.880 & 12.451 seconds

```
Enter Image Path: data/dog.jpg
data/dog.jpg: Predicted in 12.820173 seconds.
dog: 100%
truck: 92%
bicycle: 99%
Enter Image Path: data/dog.jpg
data/dog.jpg: Predicted in 13.409342 seconds.
dog: 100%
truck: 92%
bicycle: 99%
Enter Image Path: data/dog.jpg
data/dog.jpg: Predicted in 13.298548 seconds.
dog: 100%
truck: 92%
bicycle: 99%
Enter Image Path: data/dog.jpg
data/dog.jpg: Predicted in 12.406311 seconds.
dog: 100%
truck: 92%
bicycle: 99%
Enter Image Path: data/dog.jpg
data/dog.jpg: Predicted in 12.468326 seconds.
dog: 100%
truck: 92%
bicycle: 99%
```

```
Enter Image Path: data/horses.jpg
data/horses.jpg: Predicted in 12.279558 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
Enter Image Path: data/horses.jpg
data/horses.jpg: Predicted in 12.270935 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
Enter Image Path: data/horses.jpg
data/horses.jpg: Predicted in 12.513005 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
Enter Image Path: data/horses.jpg
data/horses.jpg: Predicted in 12.485204 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
Enter Image Path: data/horses.jpg
data/horses.jpg: Predicted in 12.707352 seconds.
horse: 100%
horse: 100%
horse: 96%
horse: 95%
Enter Image Path: ■
```

Thank you

