Computer vision in the new era of Artificial Intelligence and Deep Learning

Visión por computador en la nueva era de la Inteligencia Artificial y el Deep Learning

Rubén Usamentiaga*, Alberto Fernández°
*** University of Oviedo**
° TSK

Gijón (Spain)
5 – 16 April 2021

https://github.com/albertofernandezvillan/computer-vision-and-deep-learning-course

OpenCV

Configure OpenCV with GPU on Colab and benchmarking inference speed

- configure_opencv_with_gpu_on_colab.ipynb
- benchmarking_inference_speed_gpu_vs_cpu_opencv_on_colab.ipynb
- yolo_v4_opencv_dnn.ipynb

Open in Colab

- configure_opencv_with_gpu_on_colab.ipynb
- benchmarking_inference_speed_gpu_vs_cpu_opencv_on_colab.ipynb
- yolo_v4_opencv_dnn.ipynb

https://github.com/albertofernandezvillan/computer-vision-and-deep-learning-course

Configure OpenCV with GPU on Colab

- ❑ Ref 1describes how to install and compile OpenCV with GPU support. Installing OpenCV for GPU support is easier in Colab because this environment has many dependencies onboard.
- ❑ I have followed the instructions given here: "how-to-make-opencv-use-gpu-on-google-colab"

```
%cd /content
!git clone https://github.com/opencv/opencv
!git clone https://github.com/opencv/opencv_contrib
!mkdir /content/build
%cd /content/build
!cmake -DOPENCV_EXTRA_MODULES_PATH=/content/opencv_contrib/modules  -
DBUILD_SHARED_LIBS=OFF  -DBUILD_TESTS=OFF  -DBUILD_PERF_TESTS=OFF -
DBUILD_EXAMPLES=OFF -DWITH_OPENEXR=OFF -DWITH_CUDA=ON -
DWITH_CUBLAS=ON -DWITH_CUDNN=ON -DOPENCV_DNN_CUDA=ON
/content/opencv
!make -j8 install
```

Ref 1: How to use OpenCV's "dnn" module with NVIDIA GPUs, CUDA, and cuDNN

Configure OpenCV with GPU on Colab

cv2.getBuildInformation(): To check how the current installation of OpenCV was built

```
import cv2
import re

cv_info = [re.sub('\s+', ' ', ci.strip()) for ci in cv2.getBuildInformation().strip().split('\n')
           if len(ci) > 0 and re.search(r'(nvidia*:?)|(cuda*:)|(cudnn*:)', ci.lower()) is not None]
print(cv_info)
```

```
['NVIDIA CUDA: YES (ver 10.1, CUFFT CUBLAS)', 'NVIDIA GPU
arch: 30 35 37 50 52 60 61 70 75', 'NVIDIA PTX archs:',
'cuDNN: YES (ver 7.6.5)']
```

You can also download the OpenCV 4.5.1 library compiled with GPU support from this link, or using the following command:

```
!gdown --id 1-Ze3zkdzA_kDsakY_hGAZRh3aK3p5lHk
```

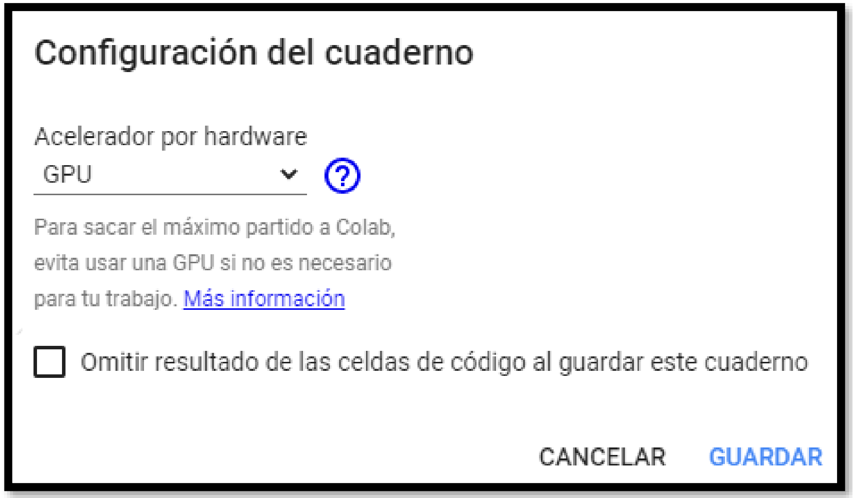
Configure OpenCV with GPU on Colab

```
# Test everything:
import cv2
import re

cv_info = [re.sub('\s+', ' ', ci.strip()) for ci in cv2.getBuildInformation().strip().split('\n')
               if len(ci) > 0 and re.search(r'(nvidia*:?)|(cuda*:)|(cudnn*:)', ci.lower()) is not None]
print(cv_info)

print('You need OpenCV 4.2 or above to use DNN_BACKEND_CUDA & DNN_TARGET_CUDA')
print('Lets check it')
print("Current OpenCV installation: '{}'".format(cv2.__version__))
try:
  print("cv2.dnn.DNN_BACKEND_CUDA: '{}'".format(cv2.dnn.DNN_BACKEND_CUDA))
  print("cv2.dnn.DNN_TARGET_CUDA: '{}'".format(cv2.dnn.DNN_TARGET_CUDA))
except AttributeError:
  print("It seems like your current OpenCV version is < 4.2 with no GPU support")
```

```
['NVIDIA CUDA: YES (ver 10.1, CUFFT CUBLAS)', 'NVIDIA GPU arch: 30 35 37 50
52 60 61 70 75', 'NVIDIA PTX archs:', 'cuDNN: YES (ver 7.6.5)']
You need OpenCV 4.2 or above to use DNN_BACKEND_CUDA & DNN_TARGET_CUDA
Lets check it
Current OpenCV installation: '4.5.1-dev'
cv2.dnn.DNN_BACKEND_CUDA: '5'
cv2.dnn.DNN_TARGET_CUDA: '6'
```

Benchmarking gpu vs cpu Opencv on Colab

- ❑ Check that the GPU is activated (and check it in your notebook)



Configuración del cuaderno

Acelerador por hardware: GPU

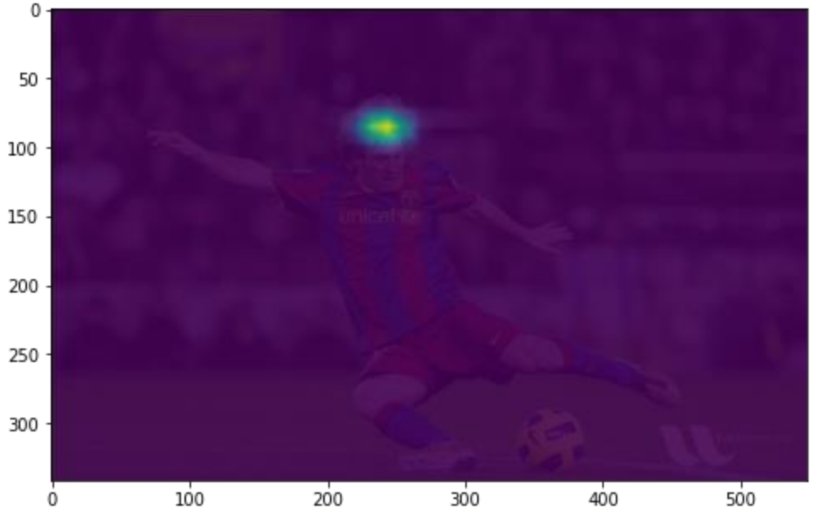
Para sacar el máximo partido a Colab, evita usar una GPU si no es necesario para tu trabajo. Más información

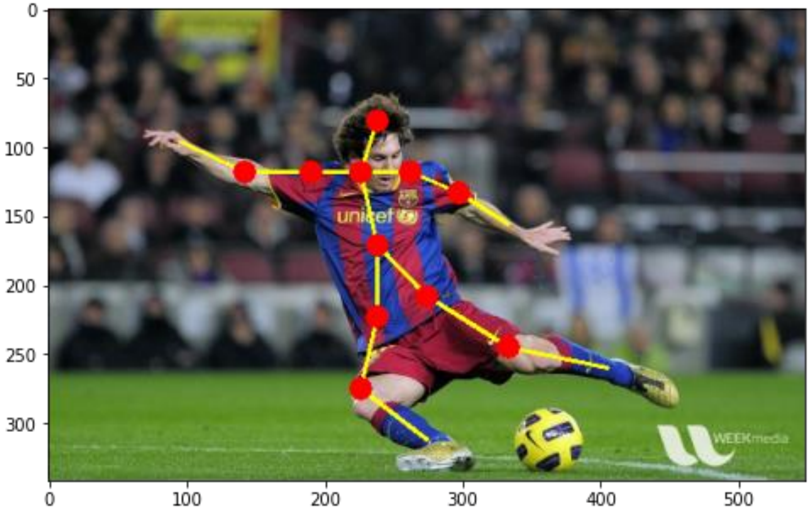
□ Omitir resultado de las celdas de código al guardar este cuaderno

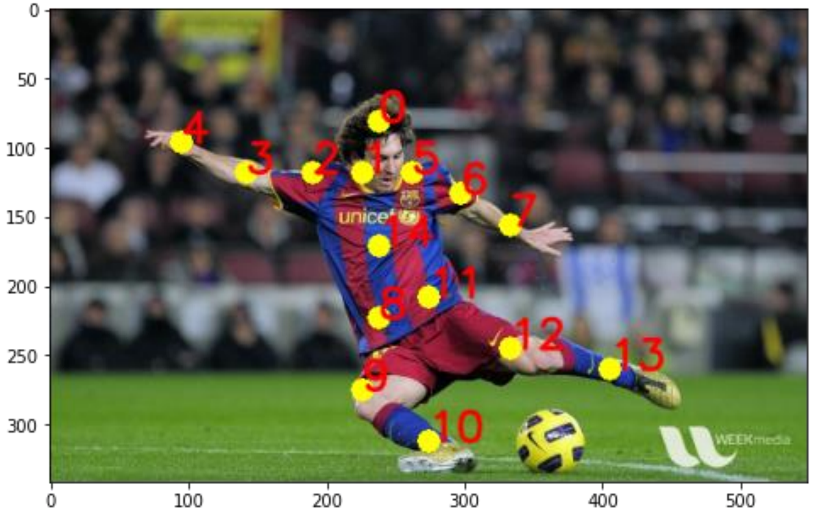
```
!nvidia-smi -L
```

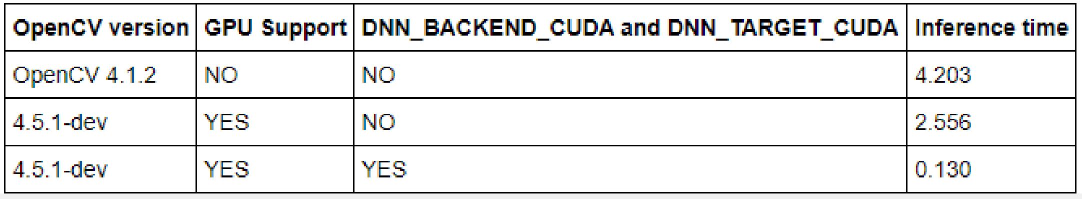
```
GPU 0: Tesla T4 (UUID: GPU-ec920886-668d-ad63-e0df-cc77e5673b16)
```

pre-trained model for human pose estimation



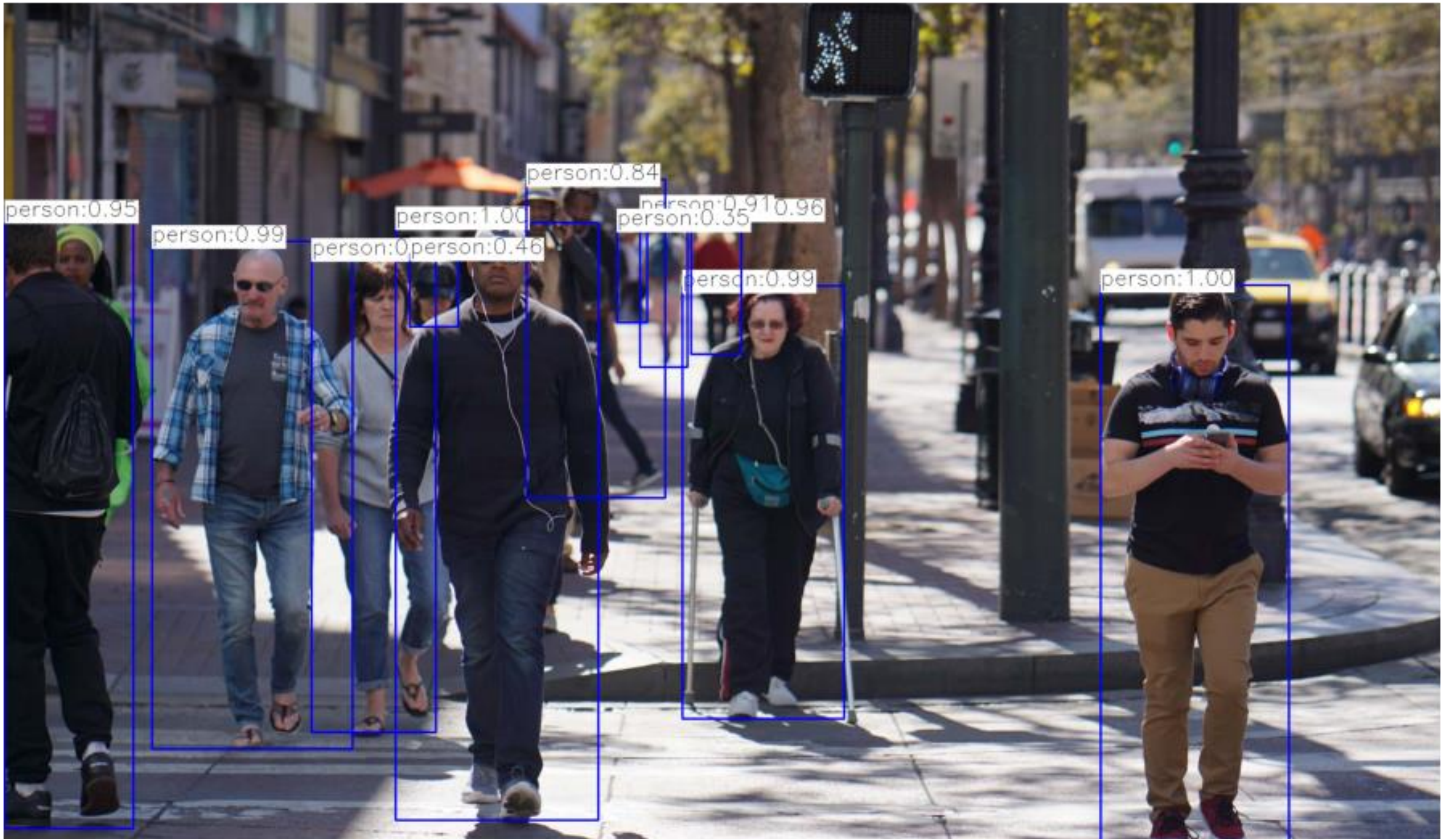


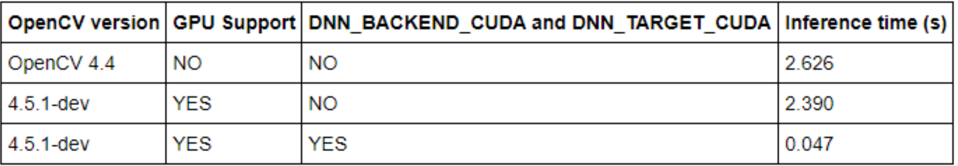




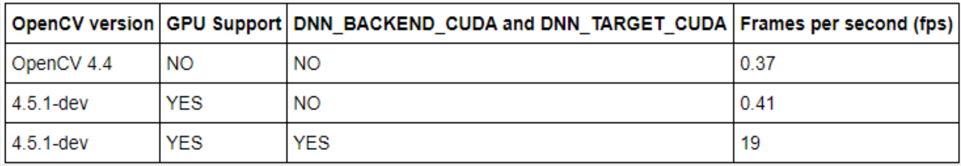
OpenCV version	GPU Support	DNN_BACKEND_CUDA and DNN_TARGET_CUDA	Inference time
OpenCV 4.1.2	NO	NO	4.203
4.5.1-dev	YES	NO	2.556
4.5.1-dev	YES	YES	0.130

pre-trained model for object detection (YOLO V4)





OpenCV version	GPU Support	DNN_BACKEND_CUDA and DNN_TARGET_CUDA	Inference time (s)
OpenCV 4.4	NO	NO	2.626
4.5.1-dev	YES	NO	2.390
4.5.1-dev	YES	YES	0.047



OpenCV version	GPU Support	DNN_BACKEND_CUDA and DNN_TARGET_CUDA	Frames per second (fps)
OpenCV 4.4	NO	NO	0.37
4.5.1-dev	YES	NO	0.41
4.5.1-dev	YES	YES	19

OpenCV

Configure OpenCV with GPU on Colab and benchmarking inference speed

https://github.com/albertofernandezvillan/computer-vision-and-deep-learning-course