



# UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732



Acreditación Institucional  
**Internacional**

OTORGADA POR EL IAC CINDE ACUERDO 55 DEL 6 DE MAYO-VIGENCIA 5 AÑOS



Vigencia por seis años



SC4289-1







UNIVERSIDAD SANTO TOMÁS  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

**Faculty:** systems engineer

**Course:** Deep Learning

**Topic:** CNN-visión por computadora (ftlite and testing)

---

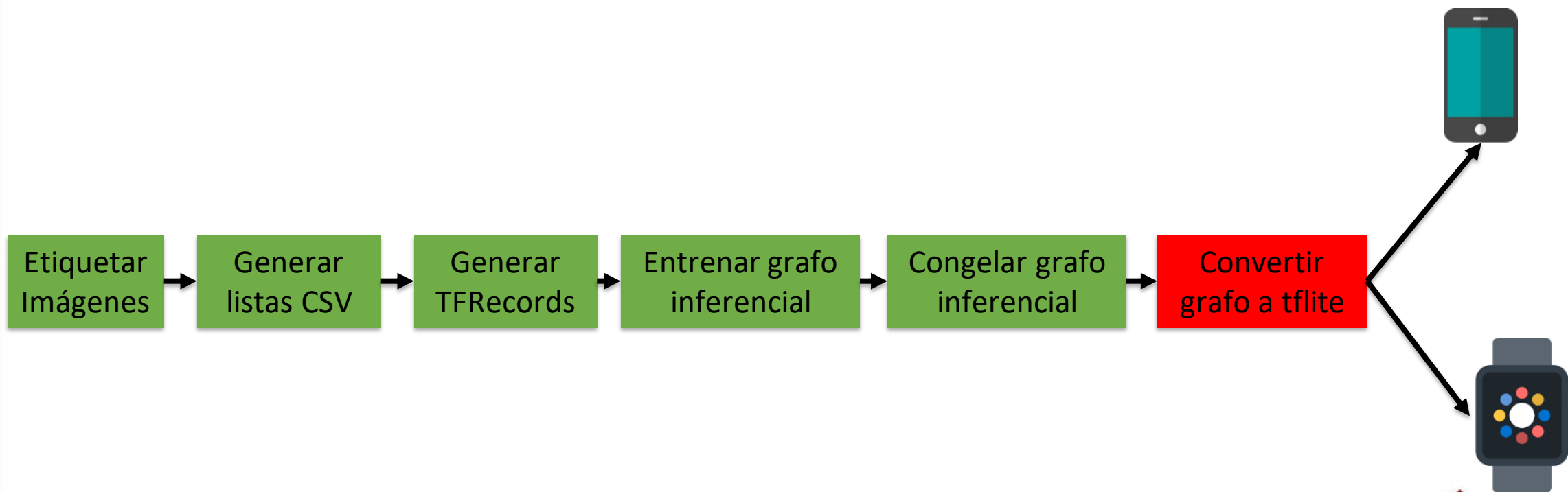
**Professor:** Luis Fernando Castellanos Guarín

**Email:** [Luis.castellanosg@usantoto.edu.co](mailto:Luis.castellanosg@usantoto.edu.co)

**Phone:** 3214582098



## Proceso Global



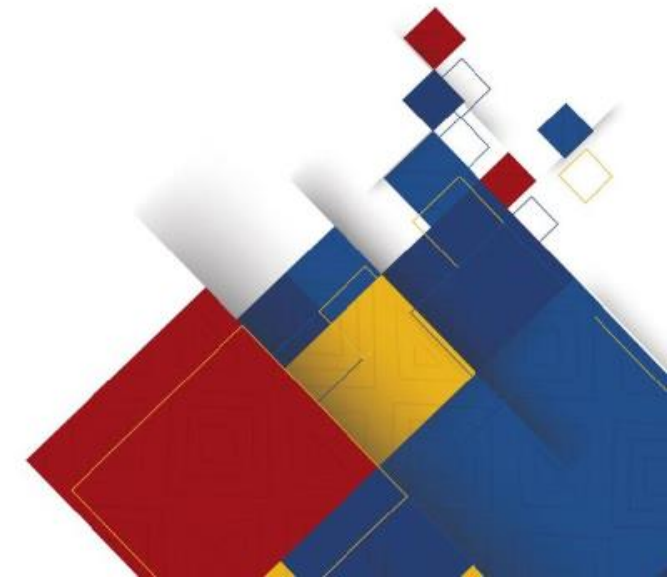
¡Siempre  
hacia lo alto!



# CONTENIDO

1. Instalar AnaConda en windows.
2. Creando un ambiente para tensorflow 1.15
3. Creando tf-lite (grafo inferencial liviano).
4. Ejecutando grafo inferencial
  1. Instalando OpenCV (Windows/Linux)
  2. en Windows.
  3. en Linux
  4. en Android

¡Siempre  
hacia lo alto!







# Instalando Anaconda en Windows

- Instalar software Anaconda, pueden usar el “paso a paso”  
<https://programarfacil.com/blog/vision-artificial/instalar-opencv-python-anaconda/>



- Verificar versiones instaladas de Python y de pip
  - `python -V`
  - `pip -V`

Si quieres comprobar si existe una versión nueva del gestor de paquetes Pip -> `python -m pip install -U pip`



# Creando ambiente Conda en Windows

Abrir “Anaconda navigator” / environments:

Creamos un nuevo ambiente al que llamaremos **tensorflow1\_15**

Este proceso puede durar varios minutos, dependiendo de la velocidad de internet y si la maquina tiene CPU o GPU.

Search Environments

base (root)

**tensorflow1\_15**

tensorflow\_env

Create Clone Import Remove

Installed Channels Update index... tensor|X

Name	T	Description	Version
✓ tensorboard		Tensorboard lets you watch tensors flow	1.15.0
✓ tensorflow		Tensorflow is a machine learning library.	1.15.0
✓ tensorflow-estimator			1.15.1

3 packages available matching "tenso" 1 package selected Apply Clear

¡Siempre  
hacia lo alto!



## Creando ambiente Conda en Windows

Por CMD ejecutar el comando

```
C:\> conda activate tensorflow1_15
```

La ruta de CMD, cambiara quedando así:

```
(tensorflow1_15) C:\>
```

Ahora instalaremos la versión de tensorflow 1.15

```
(tensorflow1_15) C:\>pip install tensorflow==1.15
```

Tenga un poco de paciencia, se demora unos minutos dependiendo de su conexión a internet y de su CPU.

¡Siempre  
hacia lo alto!



## Creando grafo inferencial liviano (tflite)

Un grafo inferencial requiere computadores con un hardware de alto rendimiento (mínimo un Intel I5 o un ADM A12) y aun así su proceso es lento.....pero en un dispositivo como una Raspberry o un teléfono que tiene menor capacidad de hardware simplemente No se puede ejecutar por eso es necesario convertir el grafo en una versión liviana (tflite)





## Creando grafo inferencial liviano (tflite)

Descargue los dos archivos que están en la carpeta de Google drive:

..\deteccion\_objectos\models\research\tflite

- tflite\_graph.pb
- tflite\_graph.pbtxt

Desde la carpeta donde guardo los archivos descargados, ejecute desde CMD y estando activado el environment de tensorflow1\_15:

```
(tensorflow1_15) E:\> tflite_convert --input_shape=1,300,300,3 --input_arrays=normalized_input_image_tensor --output_arrays=TFLite_Detection_PostProcess,TFLite_Detection_PostProcess:1,TFLite_Detection_PostProcess:2,TFLite_Detection_PostProcess:3 --allow_custom_ops --graph_def_file=tflite_graph.pb --output_file=detect.tflite
```

El proceso le debe crear un archivo llamado:

- Detect.tflite

¡Siempre  
hacia lo alto!

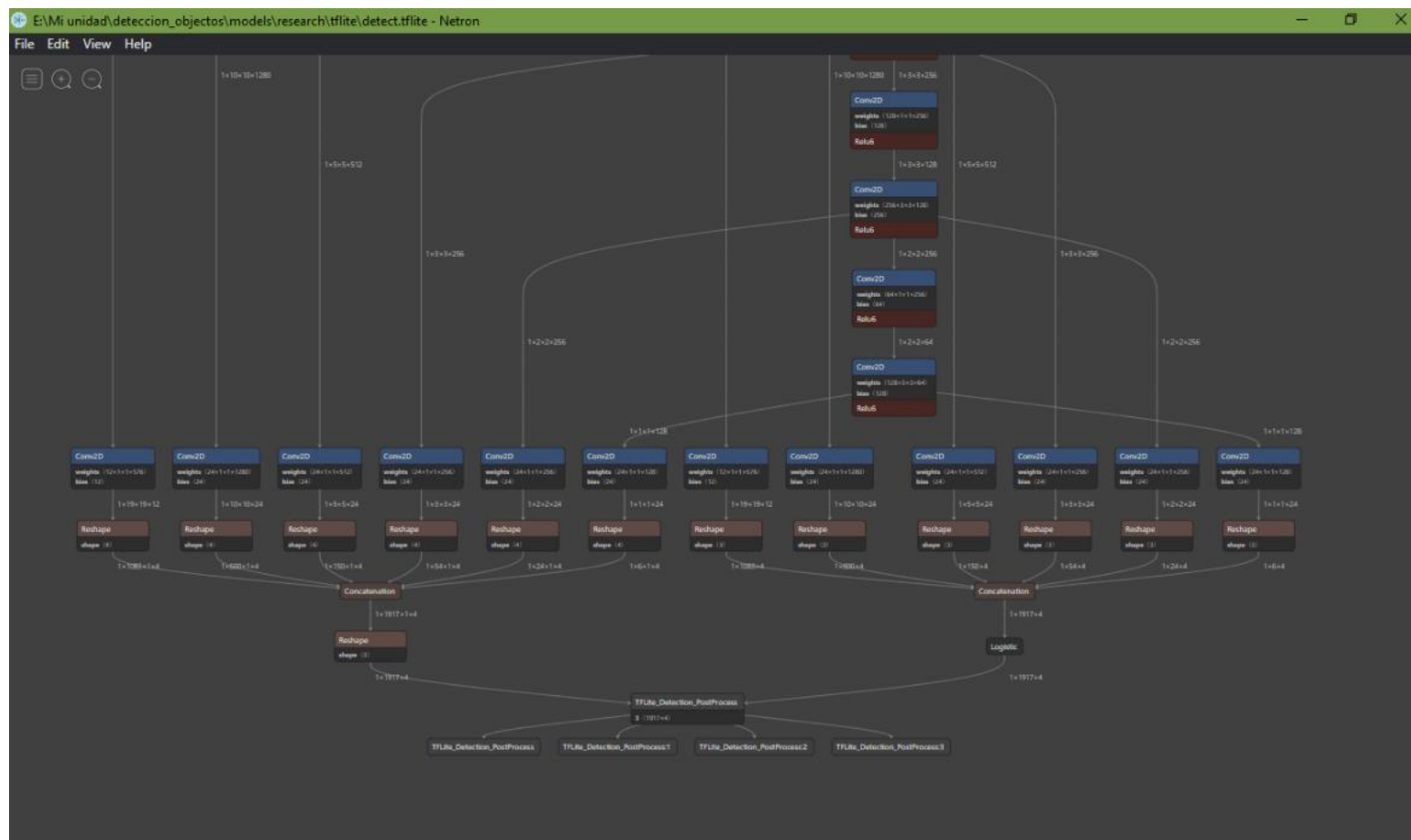


# Conociendo el grafo inferencial

El software NETRON, que esta disponible desde GITHUB

<https://github.com/lutzroeder/netron/>

Nos permite visualizar los grafos resultantes del entrenamiento de una red neuronal, en este caso de **detect.tflite**





# Instalando OpenCv

Necesitaremos tener instalado en nuestro S.O (Windows/Linux) la librería OpenCV, quien nos permitirá tener acceso a imágenes obtenidas desde un solo archivo o desde videos, cámaras de vigilancia y a aplicarles el grafo de inferencia que construimos:

## Windows:

1. Ingresar a anaconda navigator
2. seleccionar el ambiente de **tensorflow1\_15**
3. Clic en Not installed
4. Buscar la librería opencv
5. Seleccionar los tres componente
6. Dar clic en aplicar.

The screenshot shows the Anaconda Navigator application window. On the left sidebar, the 'Environments' tab is selected, and the 'tensorflow1\_15' environment is highlighted. In the main panel, the 'Channels' tab is active, and the 'Not installed' dropdown is selected. The search bar contains 'opencv', and three packages are listed: 'libopencv', 'opencv', and 'py-opencv'. The 'Apply' button at the bottom right is circled in red.

Name	Description	Version
libopencv	Computer vision and machine learning software library.	3.4.2
opencv	Computer vision and machine learning software library.	3.4.2
py-opencv	Computer vision and machine learning software library.	3.4.2





## Instalando OpenCv (Linux)

Para Instalar OpenCV en Linux requiere un poco más de esfuerzo que en Windows (es un poco más complejo):

En la carpeta compartida esta unos archivos tipo bash que nos automatizan los procesos .

**\USTA-202001\_7°\_DEEP\_LEARNING\Install\_opencv\_optimizacion\_Linux**

1. necesitaremos actualizar el SO y borrar algunas software (**P1\_linux\_Optimization.sh**)

```
sudo apt-get update --assume-yes  
sudo apt-get upgrade --assume-yes  
sudo apt-get clean --assume-yes  
sudo apt-get autoclean --assume-yes  
sudo apt-get install libatlas-base-dev --assume-yes  
sudo shutdown -r now
```



## Instalando OpenCv (Linux)

### 2. Instalaremos las dependencias que se requieren para python3: **P2\_instalar\_dependencias\_openCV.sh**

```
echo "P1: Instalando herramientas de desarrollador, incluido CMake"
sudo apt-get install build-essential cmake pkg-config --assume-yes
echo "P2: Instalando paquetes de E/S para manejar imagenes JPEG, PNG, TIFF, etc"
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev --assume-yes
echo "P3: Instalando paquetes de E/S de video"
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev --assume-yes
sudo apt-get install libxvidcore-dev libx264-dev --assume-yes
echo "P4: Instalando la biblioteca de desarrollo GTK, necesaria para OpenCV"
sudo apt-get install libfontconfig1-dev libcairo2-dev --assume-yes
sudo apt-get install libgdk-pixbuf2.0-dev libpango1.0-dev --assume-yes
sudo apt-get install libgtk2.0-dev libgtk-3-dev --assume-yes
echo "P5: Instalando dependencias adicionales para trabajar matrices en OpenCV"
sudo apt-get install libatlas-base-dev gfortran --assume-yes
echo "P6: Instalando librerias para manejar datos HDF5 y GUI Qt"
sudo apt-get install libhdf5-dev libhdf5-serial-dev libhdf5-103 --assume-yes
sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5 --assume-yes
echo "P7: Instalando archivos de encabezado de Python 3 "
sudo apt-get install python3-dev --assume-yes
sudo shutdown -r now
```



## Instalando OpenCv (Linux)

3. Instalaremos el modulo PIP y un software para crear ambientes de trabajo. **P3\_instalar\_entorno\_virtual\_openCV.sh**

```
echo "-----Creando entorno virtual de Python3 e instalar libreria NumPy-----"
ls -lah
echo "-----P1: Instalando herramienta PIP en python3-----"
wget https://bootstrap.pypa.io/get-pip.py
sudo python3 get-pip.py
sudo rm -rf ~/.cache/pip
echo "-----P1: Instalando entorno virtual llamado virtualenvwrapper-----"
sudo pip install virtualenv virtualenvwrapper
```

Agregamos variables del sistema sobre el nuevo entorno virtual

```
nano ~/.bashrc
```

y agregue las siguientes líneas al final del archivo

```
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh
```

Aplique los cambios a la sesión actual

```
source ~/.bashrc
```





# Instalando OpenCv (Linux)

4. Creando ambiente de virtual

```
mkvirtualenv IA_opencv -p python3
```

5. Instalamos OpenCV en el ambiente

```
(IA_opencv) root:~$ pip install opencv-contrib-python==4.1.0.25
```

No solo descarga OpenCV también descarga Numpy

Una vez termine reinicie su SO para evidenciar los cambios

Recuerde que todo de acá en adelante se hará desde el ambiente

¡Siempre  
hacia lo alto!



**Listo ya tenemos todo para  
ejecutar nuestro grafo inferencial**

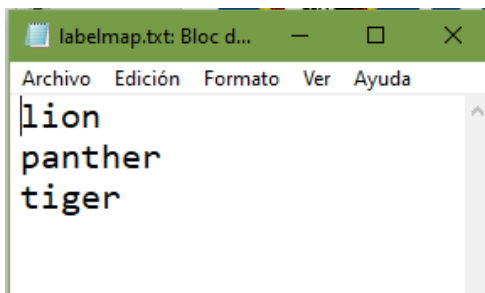
**¡Siempre  
hacia lo alto!**



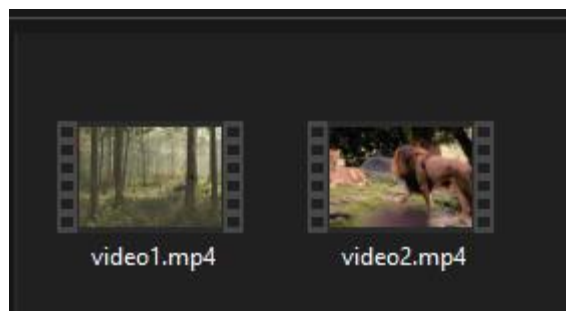
## Ejecutando grafo inferencial (Windows/linux)

Crearemos una carpeta en c:/ a la que llamaremos **IA\_tflite**, dentro de ella vamos a crear dos carpetas:

- **Model**, dentro de ella colocaremos nuestro grafo inferencial liviano (**detect.tflite**) al que le cambiaremos el nombre a **model.tflite**...adicional crearemos un archivo que llamaremos **labelmap.txt**, donde tendremos las etiquetas



- **Multimedia**, dentro de ella tendremos los videos e imágenes con el que deseamos poner a prueba nuestro grafo inferencial, para este ejercicio crearemos dos (**video1.mp4** y **video2.mp4**).







## Ejecutando grafo inferencial (Windows/linux)

En la carpeta compartida de drive

**USTA-202001\_7°\_DEEP\_LEARNING\Computer\_vision\IA\_tflite**

Estan los archivos para ejecutar

### **Python (archivos base):**

- TFLite\_detection\_image.py
- TFLite\_detection\_video.py

### **Windows:**

- windows\_video.bat

### **Linux:**

- linux\_video.sh
- linux\_webcam.sh

¡Siempre  
hacia lo alto!



UNIVERSIDAD SANTO TOMÁS  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA  
SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

¡Siempre  
hacia lo alto!

[USTATUNJA.EDU.CO](http://USTATUNJA.EDU.CO)



@santotomastunja