

# OpenCV in Python. Instalare

Catalin Stoean

[catalin.stoean@inf.ucv.ro](mailto:catalin.stoean@inf.ucv.ro)

<http://inf.ucv.ro/~cstoean>

# Instalarea

- Presupun ca nu avem Python instalat
- Navigam la <https://www.anaconda.com/> si apoi descarcam versiunea Python 3.x, unde x este cea mai recenta versiune
  - Nu adaugati Anaconda la calea variabilelor de mediu (in laborator)
- Dupa instalare, de la Start pornim Anaconda Prompt
  - Se deschide o fereastră asemanătoare DOS Prompt
  - De aici putem rula python
- Petru a instala orice pachet pe care vrem sa il folosim in python, putem da comanda
  - pip install *pachet*

# Instalarea TensorFlow & Keras

Anaconda Prompt

```
(base) C:\Users\Lab User>pip install tensorflow
```

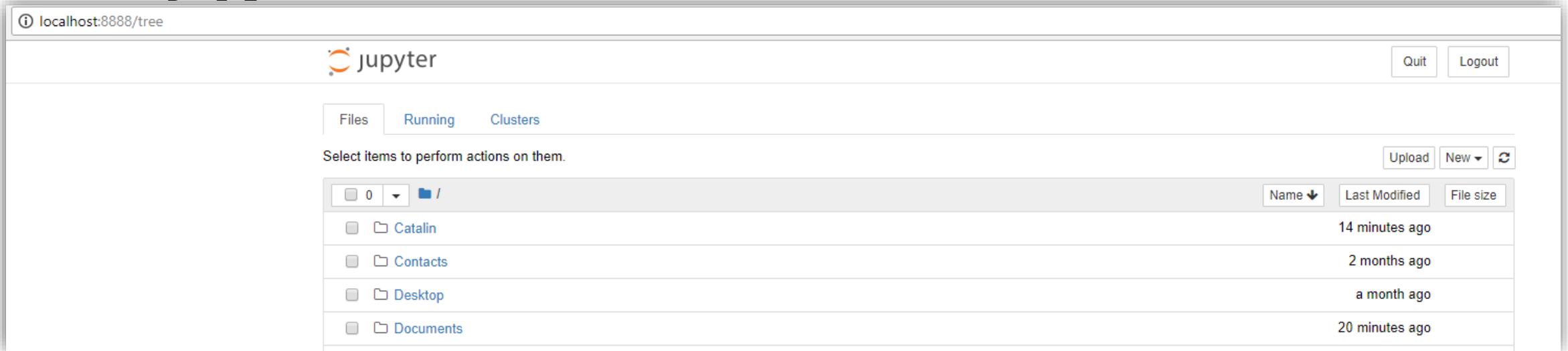
- TensorFlow este o librerie pentru masini instruibile construita de o echipa Google.
- Urmatorul pachet de instalat: Keras
  - `pip install keras`
- Keras este o librerie de retele neuronale care functioneaza peste TensorFlow
- Putem face si o actualizare pentru pip, daca ne este sugerata:

```
Successfully installed keras-2.2.2 keras-applications-1.0.4 keras-preprocessing-1.0.2  
You are using pip version 10.0.1, however version 18.0 is available.  
You should consider upgrading via the 'python -m pip install --upgrade pip' command.  
(base) C:\Users\Lab User>python -m pip install --upgrade pip
```



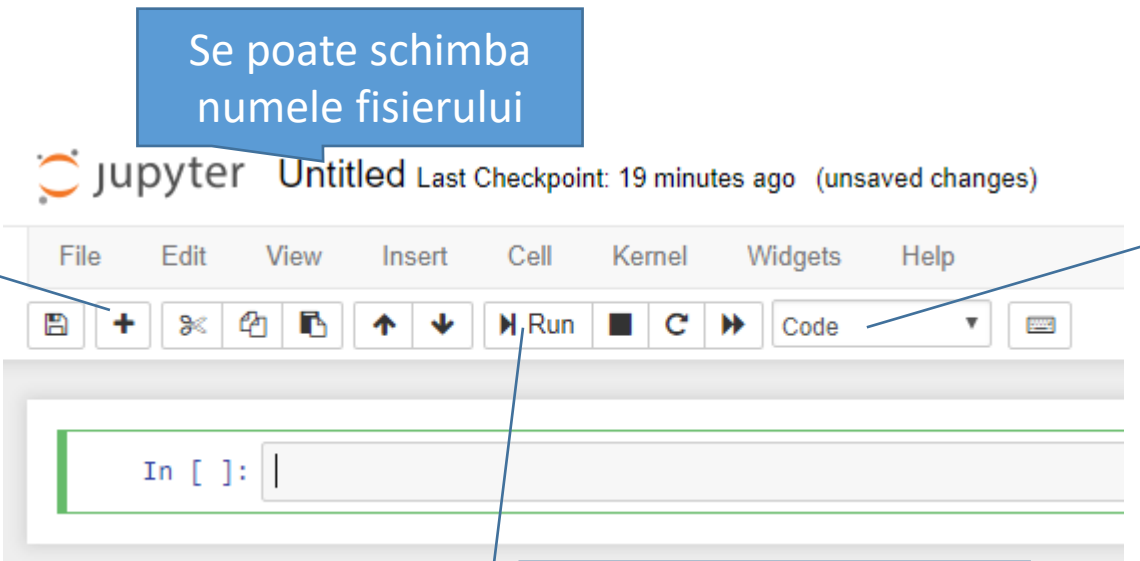
# Instalarea – cont.

- `pip install opencv-python`
- Vom folosi Jupyter notebook (deja instalat cu Anaconda)
  - Acesta este o aplicatie web care permite scrierea si rulara de cod Python (si altele)
- Deschidem interfata cu comanda:
  - `jupyter notebook`



# Jupyter

- De aici se poate naviga pana la folderul in care dorim sa ne salvam fisierele si folosim New pentru
  - a crea un fisier (python) nou
  - Sau a crea un folder nou

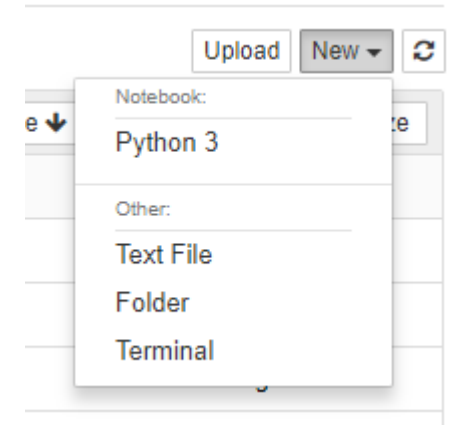


The screenshot shows the Jupyter web interface. At the top, it says 'jupyter Untitled Last Checkpoint: 19 minutes ago (unsaved changes)'. Below that is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar contains icons for saving, adding a new cell, undo, redo, and running. The 'Run' button is highlighted with a blue box and the text 'Rularea celulei curente'. The 'Code' dropdown menu is also highlighted with a blue box and the text 'Tipul celulei curente de mai jos. code = cod python, markdown = text, heading = titlu (nu este rulat)'. A blue box on the left says 'Adaugare a unei celule noi' with an arrow pointing to the '+' icon. Another blue box at the top says 'Se poate schimba numele fisierului' with an arrow pointing to the 'File' menu.

Se poate schimba numele fisierului

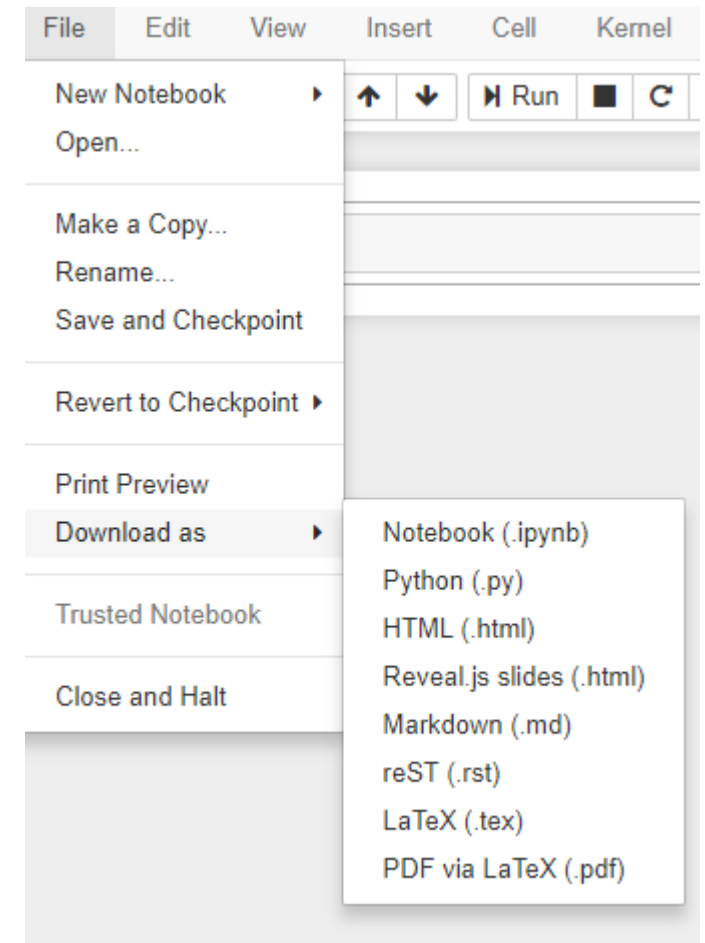
Adaugare a unei celule noi

Rularea celulei curente



# Jupyter notebook

- Se pot adauga celule atat sub, cat si deasupra unei celule curente
- Un notebook se poate exporta in diferite moduri (vezi poza)
- O rulare poate fi oprita
- Variabilele sunt valabile de la o celula la alta
- Se pot rula toate celulele din notebook o data



# Deschiderea unei imagini

```
In [ ]: import numpy as np
import cv2

# incarcam o imagine
img = cv2.imread('yes.jpg')
```

In timpul rularii, aici apare \*  
Dupa rulare, apare un numar care spune de cate ori a fost rulat

```
In [*]: cv2.imshow('Vechea noastra poza',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [ ]:



Poza s-a deschis intr-o fereastră separata  
si asteapta o tasta ca sa se inchida.

# Deschidem, inchidem sau salvam&inchidem

- Salvarea unei poze in fisier se face (ca si in C++) cu imwrite

```
In [ ]: import numpy as np
import cv2

img = cv2.imread('yes.jpg')
cv2.imshow('Vechea noastra poza', img)
tasta = cv2.waitKey(0)
if tasta == 27:          # daca se apasa ESC, inchidem fereastra
    cv2.destroyAllWindows()
elif tasta == ord('s'): # daca se apasa 's' salvam cu ce extensie vrem si inchidem fereastra
    cv2.imwrite('yes.png', img)
    cv2.destroyAllWindows()
```

- Intr-o alta celula din acelasi notebook nu mai era nevoie sa facem import-urile sau sa citim img
  - Ar fi fost cunoscute din celula de mai sus



# Putem afisa poza si in notebook cu Matplotlib

- OpenCV citeste pozele in format BGR, in timp ce Matplotlib are format RGB
  - Conversia se face cu `cvtColor`.
- Daca vrem sa nu apara axele, putem adauga
- `plt.axis('off')`

```
from matplotlib import pyplot as plt  
plt.imshow(img, interpolation = 'bicubic')  
plt.show()
```



```
from matplotlib import pyplot as plt  
imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
plt.imshow(imgRGB, interpolation = 'bicubic')  
plt.show()
```



Exc: Afisati mai multe poze in aceeasi imagine folosind Matplotlib