




NHL
STENDEN
computer vision
& data science



Van De Loosdrecht
Machine Vision

Computer Vision & Deep Learning

VisionLab Python interoperability

12 September 2018

Copyright © 2001 – 2018 by
NHL Stenden Hogeschool and Van de Loosdrecht Machine Vision BV
All rights reserved
Jaap van de Loosdrecht, Klaas Dijkstra
j.van.de.loosdrecht@nhl.nl, jaap@vdlmv.nl

VisionLab Python interoperability

Overview

Using Python from VisionLab

- Open/save *.py and *.npy files
- Calling Python commands from VisionLab scripts
- Demo Inference, using a Deep Learning network
- Overview VisionLab script commands for Python

Using VisionLab from Python

- Install VisionLab as Python module
- Calling VisionLab operators from Python scripts

Note: use a Python environment with the correct version number. The current version number of Python that VisionLab uses can be found in the file changes.txt in the VisionLab distribution

9/18/2018

VisionLab Python interoperability

2

Script commands overview

The VisionLab script language has the following commands:

- **PyStart**: start the Python interpreter
- **PyStop**: stop the Python interpreter
- **PyEval**: evaluate a Python function
- **PyExec**: execute a Python statement
- **PyExecFile**: load a Python file and execute
- **PyImageToNP**: convert a VisionLab image to a Numpy array
- **PyNpToImage**: convert a Numpy array to a VisionLab image
- **PyCWD**: change current working directory of Python interpreter
- **PyPWD**: return current working directory of Python interpreter
- **(I)ReadNumPy**: read numpy array and convert to VisionLab image
- **(I)WriteNumPy**: write VisionLab image to numpy array

9/18/2018

VisionLab Python interoperability

3

Python files

With GUI open file VisionLab supports:

- ***.py**: open, edit and save Python source code files
- ***.npy**: open NumPy array and convert to VisionLab image
save VisionLab image as NumPy array

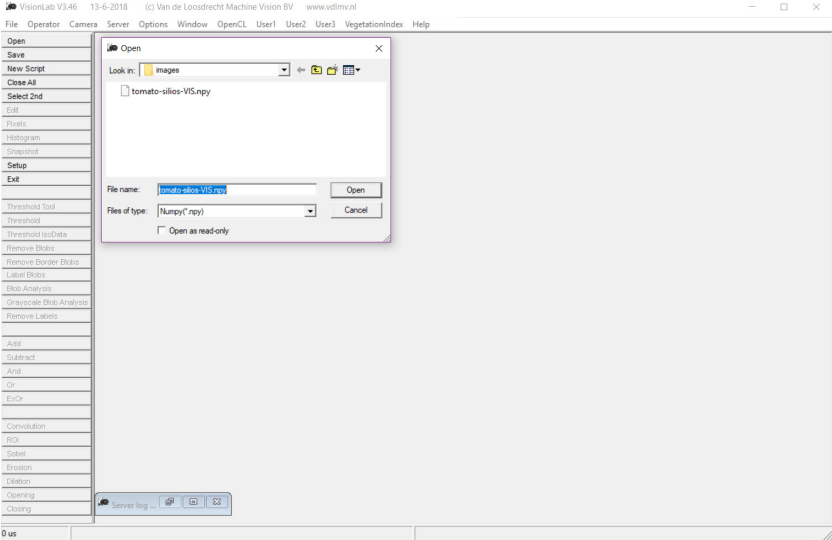
Example open file tomato-silios-VIS.npy

9/18/2018

VisionLab Python interoperability

4

Open NumPy file

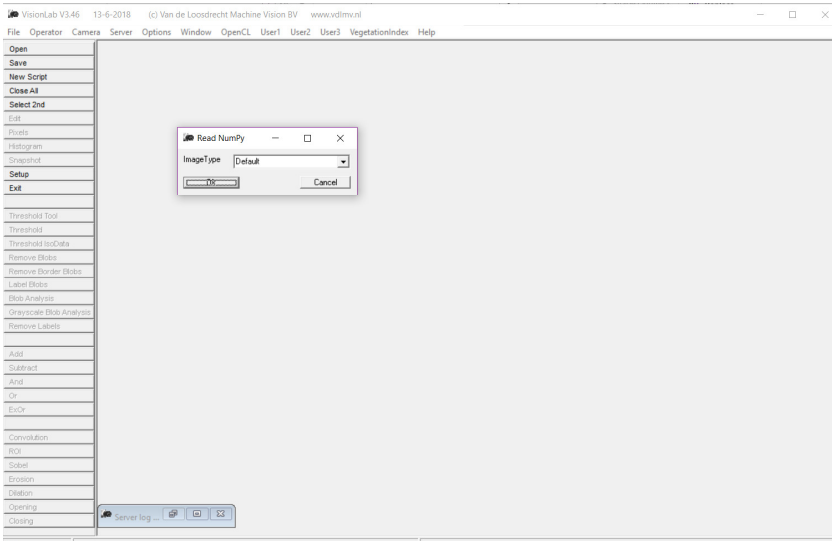


9/18/2018

VisionLab Python interoperability

5

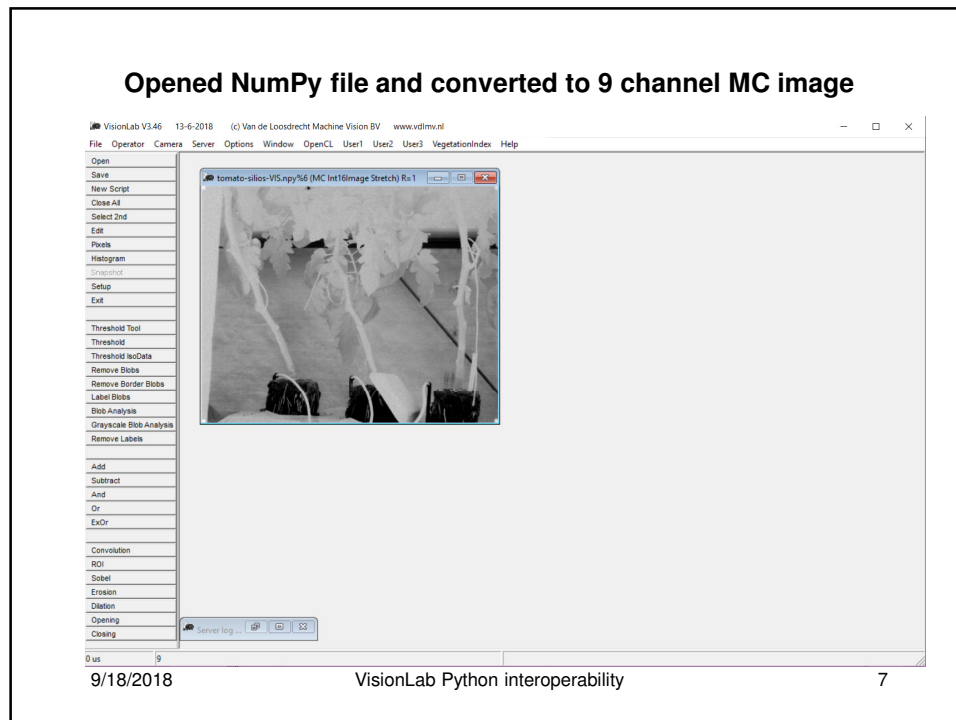
Choose image type
default is (MC) image of same type as NumPy array



9/18/2018

VisionLab Python interoperability

6



Demo inference Deep Learning

Files used:

- **inference.py, Python code:**
 - Load a pre-trained DL network
 - Define predict function with has a numpy array as parameter
- **Mug.jls, VisionLab script:**
 - Initialise path server with path client
 - Start Python interpreter with Anaconda environment
 - Import predict function from Python module inference
 - Read with VisionLab test image*
 - Zoom image to required format for pre-trained DL network
 - Convert VisionLab image to Numpy array
 - Call Python predict function with Numpy array
 - Display result

* image crop of <https://www.flickr.com/photos/jfanaian/4994221690/>

9/18/2018

VisionLab Python interoperability

8

Demo inference Deep Learning

Note this demo can only be executed if an Anaconda environment with name py36 is installed with:

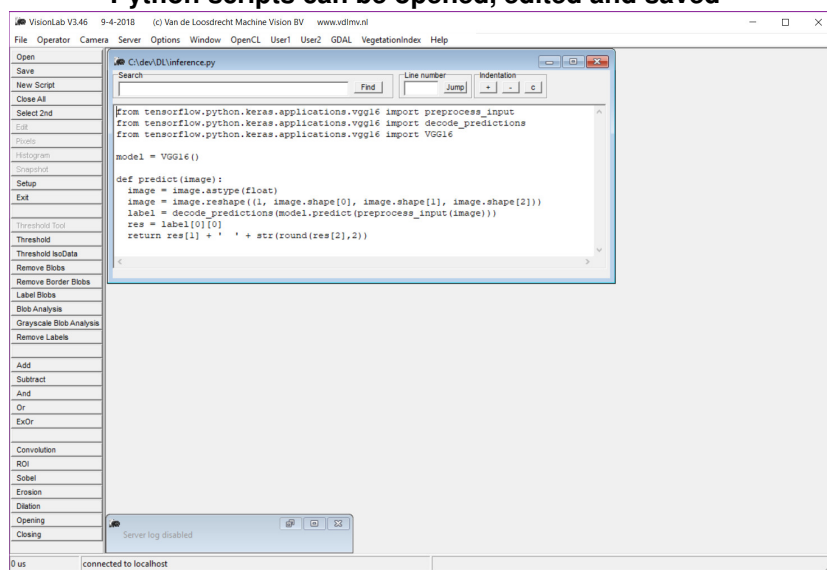
- Python 3.6
- Tensorflow 1.7

9/18/2018

VisionLab Python interoperability

9

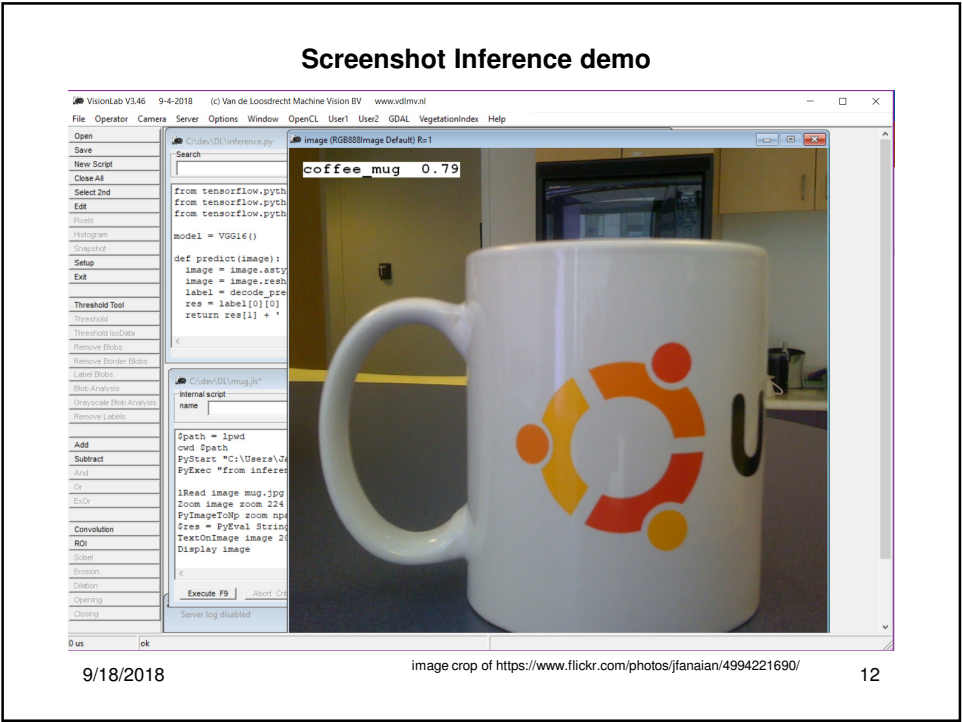
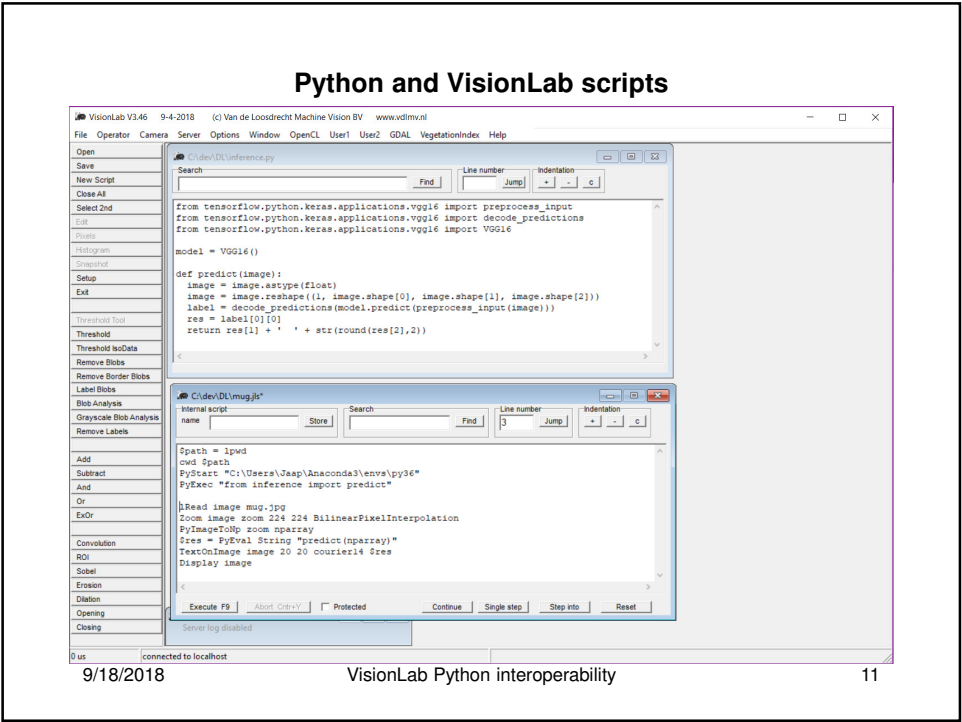
Python scripts can be opened, edited and saved



9/18/2018

VisionLab Python interoperability

10



Starting/Stoping the Python command interpreter

Note:

Currently it is only possible to start the Python command interpreter once with the VisionLab script command “PyStart”.

The “PyStop” is a NOP and starting the Python command interpreter for a second time will result in an error message in the server console.

This “bug” is due to that currently it is not possible to release the resources allocated by the NumPy module.

9/18/2018

VisionLab Python interoperability

13

Python command interpreter under Linux

The PyStart command should specify the home directory of the Python environment used

In order to find the correct version python3.6m.so LD_LIBRARY_PATH should be exported. This can be done by adding a command in .bashrc like:

```
export LD_LIBRARY_PATH="/mnt/d/anaconda3-linux/pkgs/python-3.6.6-hc3d631a_0/lib"
```

9/18/2018

VisionLab Python interoperability

14

Live demo inference with reloading Python module

Restart VisionLab

Files used:

- inference.py, same as previous
- inference.jls, VisionLab script:
 - Initialise path server with path client
 - Do only one time:
 - Start Python interpreter with Anaconda environment
 - Import Python module inference
 - Initialises camera
 - Reload module inference
 - Loop
 - Capture image
 - Zoom image to required format for pre-trained DL network
 - Convert VisionLab image to Numpy array
 - Call Python predict function with Numpy array
 - Display result

9/18/2018

VisionLab Python interoperability

15

Live demo inference with reloading Python module

Note this demo can only be executed if an Anaconda environment with name py36 is installed with:

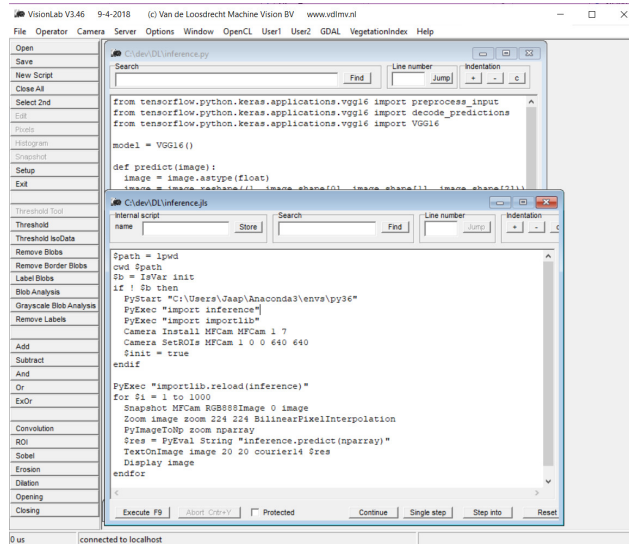
- Python 3.6
- Tensorflow 1.7
- Camera initialisation commands must be adapted to camera used
- Python file can be edited and saved, is reloaded when VisionLab script is executed

9/18/2018

VisionLab Python interoperability

16

Screenshot



9/18/2018

VisionLab Python interoperability

17

Script commands overview

The VisionLab script language has the following commands:

- **PyStart**: start the Python interpreter
- **PyStop**: stop the Python interpreter
- **PyEval**: evaluate a Python function
- **PyExec**: execute a Python statement
- **PyExecFile**: load a Python file and execute
- **PyImageToNP**: convert a VisionLab image to a Numpy array
- **PyNpToImage**: convert a Numpy array to a VisionLab image
- **PyCWD**: change current working directory of Python interpreter
- **PyPWD**: return current working directory of Python interpreter
- **(I)ReadNumPy**: read numpy array and convert to VisionLab image
- **(I)WriteNumPy**: write VisionLab image to numpy array

9/18/2018

VisionLab Python interoperability

18

VisionLab script commands for Python

- **PyStart <PythonHomeDir>**

Start the Python interpreter with optional parameter the "Python home directory". In case Anaconda environment are used this should be home directory of the root of the desired Conda environment, like
`\Users\<username>\Anaconda3\envs\<env_name>`

- **PyStop**

Stop the Python interpreter and releases the memory used

9/18/2018

VisionLab Python interoperability

19

VisionLab script commands for Python

- **PyEval <function result type> <Python command string>**

Execute the string with the Python command, the expected result should be of the specified type.

In order to avoid unwanted pre-processing by the VisionLab interpreter, it is advised to specify the Python command surrounded with double quotes. If a Python string constant is to be used, embed the Python string with single quotes.

Possible values for <function result type> are:
None String Integer Float Bool

Example:

```
$res = PyEval String "str(np_arrayX.shape)"
```

9/18/2018

VisionLab Python interoperability

20

VisionLab script commands for Python

- **PyExec <Python command string>**

Execute the string with the Python command, no function result is expected

In order to avoid unwanted pre-processing by the VisionLab interpreter, it is advised to specify the Python command surrounded with double quotes. If a Python string constant is to be used, embed the Python string with single quotes.

Example:

PyExec "py_operator_that_needs_a_string('string_parameter')"

9/18/2018

VisionLab Python interoperability

21

VisionLab script commands for Python

- **PyExecFile <function result type> <file name>**

Execute specified Python file in Python interpreter, the expected result should be of the specified type

Possible values for <function result type> are:

None String Integer Float Bool

9/18/2018

VisionLab Python interoperability

22

VisionLab script commands for Python

- **PyImageToNp <image> <narray>**

A numpy array with the name <narray> is added to the global scope of the Python interpreter. This numpy array is initialised with the shape and contents of the specified VisionLab image.

- **PyNptolImage <narray> <image> <imageType>**

A VisionLab image with the name <image> and type <imageType> is created in the VisionLab interpreter and initialised from the numpy array narray

Possible values for <imageType> are:

ByteImage DoubleImage FloatImage Int8Image Int16Image Int32Image
 RGB888Image RGB161616Image HSV888Image HSV161616Image
 YUV888Image YUV161616Image ComplexDoubleImage
 ComplexFloatImage

9/18/2018

VisionLab Python interoperability

23

VisionLab script commands for Python

- **PyCWD <path>:** change current working directory of Python interpreter to specified path
- **PyPWD:** return current working directory of Python interpreter
- **IReadNumPy <image> <fileName> <imageType>:** a numpy array with fileName is read using file path of client and converted to an image of specified type. Note that element type of numpy array and pixel type of image must be the same
- **ReadNumPy <image> <fileName> <imageType>:** a numpy array with fileName is read using file path of server and converted to an image of specified type. Note that element type of numpy array and pixel type of image must be the same

9/18/2018

VisionLab Python interoperability

24

VisionLab script commands for Python


- **IWriteNumPy <image> <fileName>**: image is converted to a numpy array and written to disk with fileName using file path of client
- **WriteNumPy <image> <fileName>**: image is converted to a numpy array and written to disk with fileName using file path of server.

9/18/2018

VisionLab Python interoperability

25

Install VisionLab Python Package

- 1) Install  **vlpypkg** in your **anaconda** environment:
`python setup.py install`
- 2) Manually copy **license_visionlab.txt** to your **python scripts folder**
- 3) Import VisionLab using: `import visionlab`
- 4) Init the package with `visionlab.InitVisionLib()`.
- 5) All visionlab operators are available under visionlab, e.g.:
 - 1) `img = visionlab.ByteImage(10, 50)`
 - 2) `visionlab.Noise(img, 0.5, 0, 255)`
 - 3) `Visionlab.Threshold(img, 0, 128)`

For function parameters and other details look at:

<http://www.vdLMV.nl/VisionLabHelp.html>

VisionLab Python Package and NumPy

- **def to_np(img : visionlab.Image) -> np.ndarray:**
 - **Description:** Convert a VisionLab image to a NumPy array.
 - **img:** Any VisionLab Image of type visionlab.Image.
 - **Output:** NumPy Array of type np.ndarray.
- **def to_vl(arr : np.ndarray, type: Type = None, layout : Layout = Layout.Mosaic) -> visionlab.Image**
 - **Description:** Convert a NumPy array to a VisionLab Image.
 - **arr:** NumPy Array of type np.ndarray.
 - **type:** Type of the output. When None the type is inferred.
 - **layout:** The layout of the NumPy array.
 - **Output:** The VisionLab image of type visionlab.Image.

Type and Layout are explained on the next slide.

9/18/2018

VisionLab Python interoperability

27

VisionLab Python Package and Type and Layout

- **class Type(enum.Enum)**
 - Indicates any of the VisionLab image Types:
 - ByteImage = 0 ... ComplexDoubleImage = 13
- **class Layout(enum.Enum)**
 - Determines the order of dimensions in the NumPy array:
 - Mosaic = 0 indicates a shape of [height, width, channels]
 - Planar = 1 indicates a shape of [channels, height, width]

9/18/2018

VisionLab Python interoperability

28