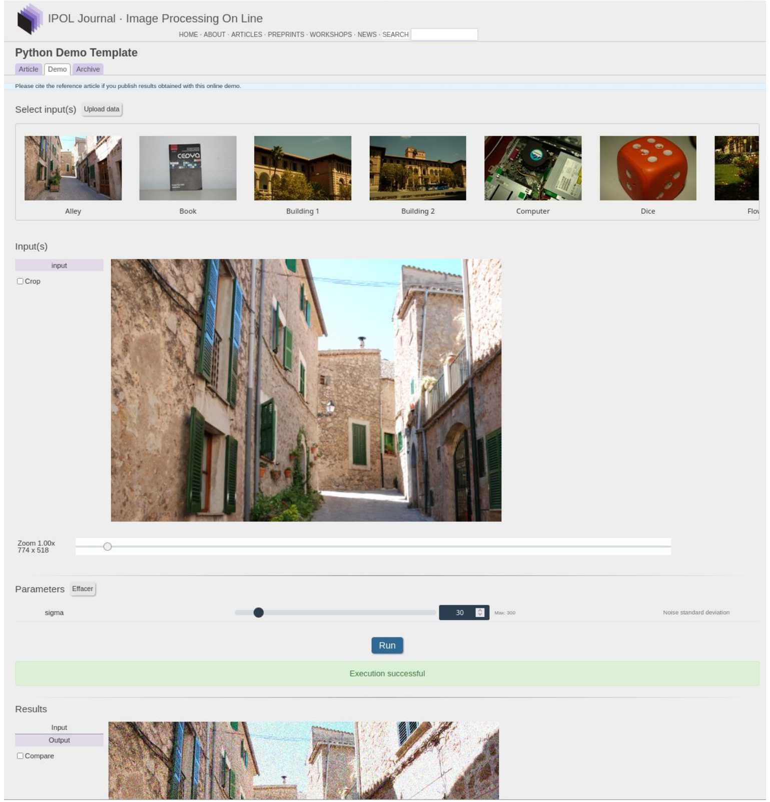
IPOL online demo

Online demos defined by:

inputs
parameters
outputs

location of the source code
description of the environment
command line to execute the program



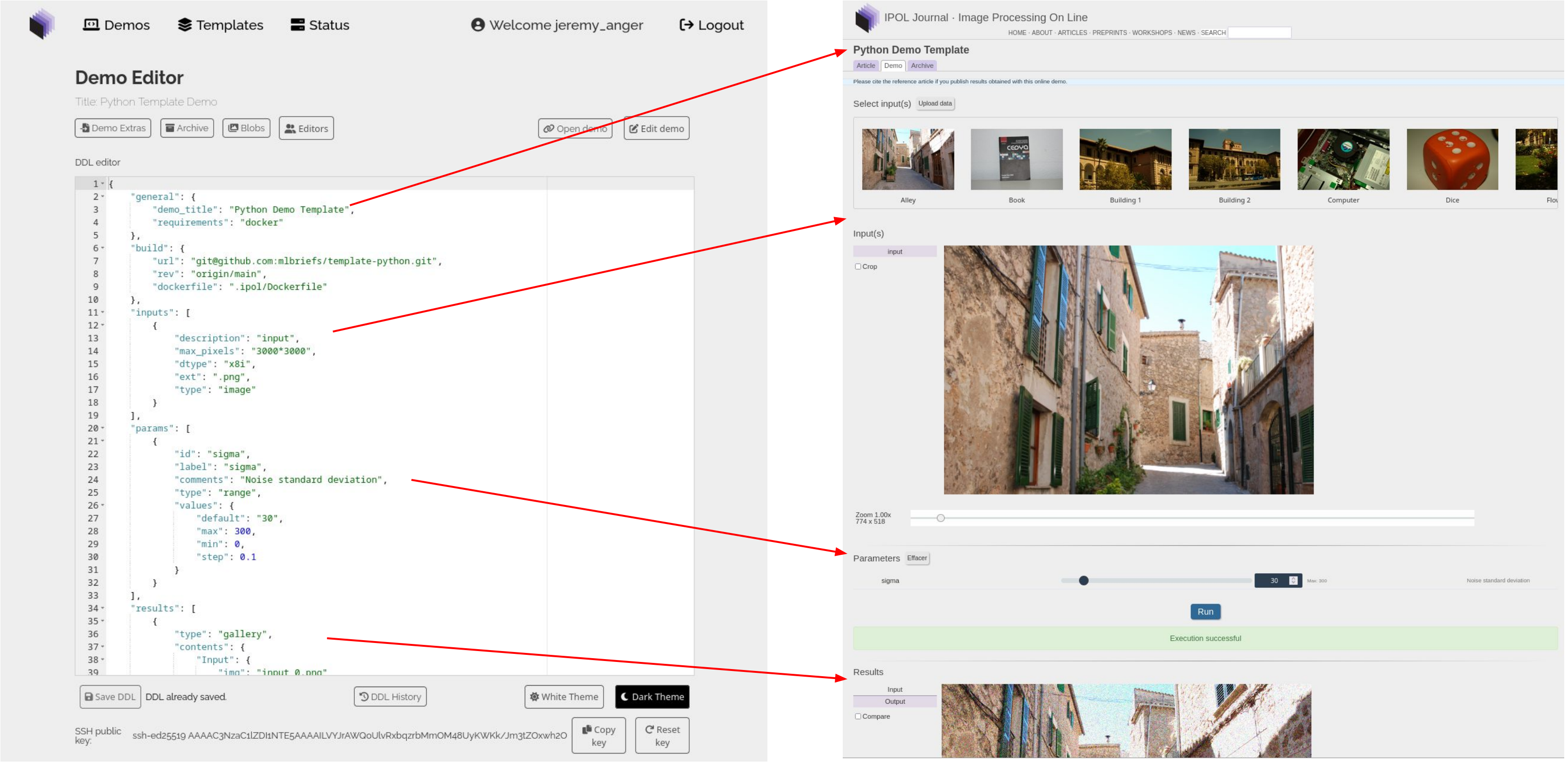
Please cite the reference article if you publish results obtained with this online demo.

Zoom 1.00x
774 x 518

sigma 30 Max: 300 Noise standard deviation

Execution successful

IPOL online demo



Demo Editor

Title: Python Template Demo

DDL editor

```
{
    "general": {
        "demo_title": "Python Demo Template",
        "requirements": "docker"
    },
    "build": {
        "url": "git@github.com:mlbriefs/template-python.git",
        "rev": "origin/main",
        "dockerfile": ".ipol/Dockerfile"
    },
    "inputs": [
        {
            "description": "input",
            "max_pixels": "3000*3000",
            "dtype": "x8i",
            "ext": ".png",
            "type": "image"
        }
    ],
    "params": [
        {
            "id": "sigma",
            "label": "sigma",
            "comments": "Noise standard deviation",
            "type": "range",
            "values": {
                "default": "30",
                "max": 300,
                "min": 0,
                "step": 0.1
            }
        }
    ],
    "results": [
        {
            "type": "gallery",
            "contents": {
                "Input": {
                    "img": "input_0.png"
```

DDL already saved.

SSH public key: ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILVYJrAWQoUlvRxbqzrbMmOM48UyKWKk/Jm3tZOxwh2O

IPOL Journal · Image Processing On Line
HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

Python Demo Template

Please cite the reference article if you publish results obtained with this online demo.

Select input(s)

Input(s)

Zoom 1.00x
774 x 518

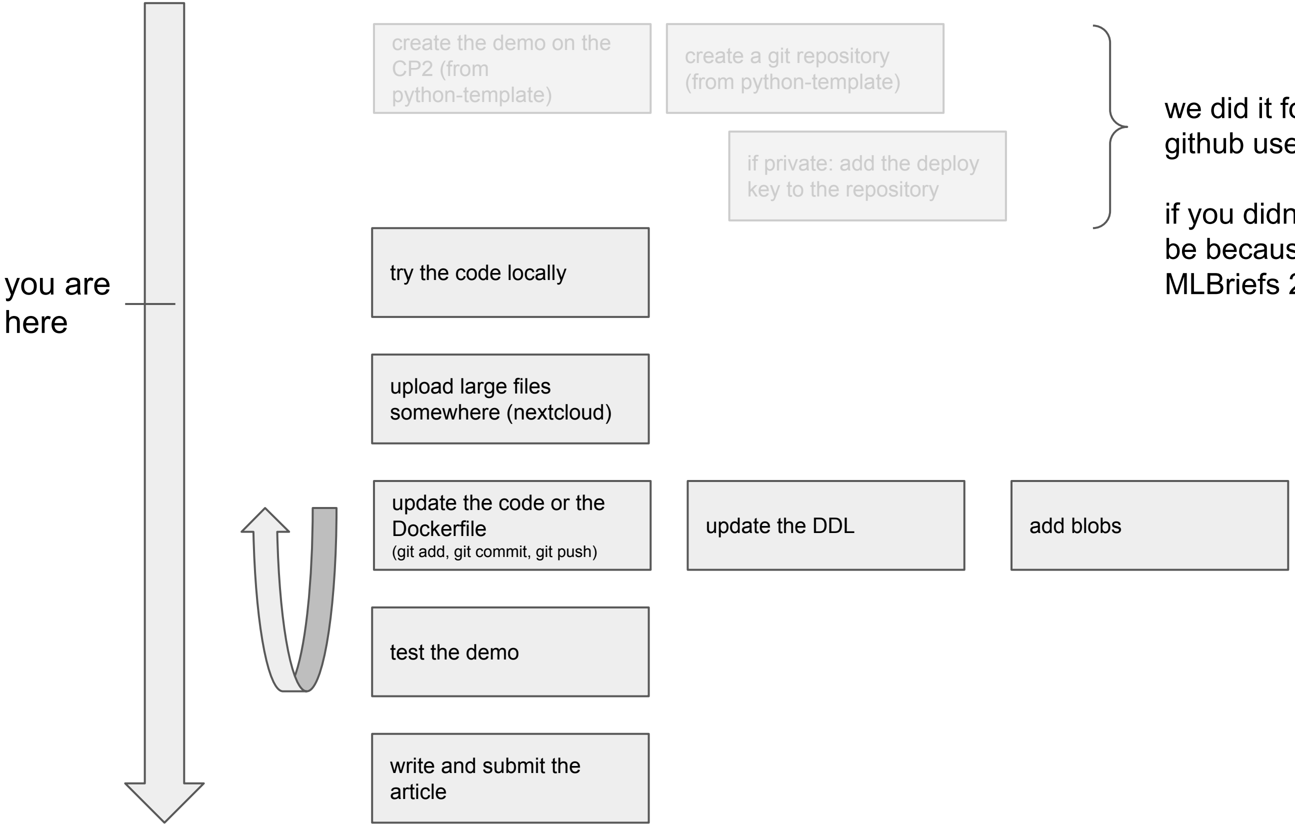
Parameters

sigma 30 Max: 300 Noise standard deviation

Execution successful

Results

The MLBriefs workflow



you are here

create the demo on the CP2 (from python-template)

create a git repository (from python-template)

if private: add the deploy key to the repository

try the code locally

upload large files somewhere (nextcloud)

update the code or the Dockerfile
(git add, git commit, git push)

update the DDL

add blobs

test the demo

write and submit the article

we did it for you (but only if you gave us a github username)

if you didn’t receive a github invitation, it might be because you can reuse the repository from MLBriefs 2 (ask us anyway)

First checklist

- check the code license if it is not your own
- check-out the template for Python: https://github.com/mlbriefs/template-python
- identify the inputs / parameters / outputs of the method
  - type and range for parameters
    - expose them to command line (*argparse*, etc)
  - format for inputs/outputs (images, plots, text, …)
- list Python requirements (with version) and system libraries required
- upload large files (weights) to our nextcloud
- the execution shouldn’t last more than 30secs (with many cores)
  - reduce the size of the inputs if necessary
- if it runs in a notebook, convert it to a Python script
- make sure the method works as expected locally!
- check that the demo & github repository was pre-created (or create it ; or ask us!)

A template for your demo

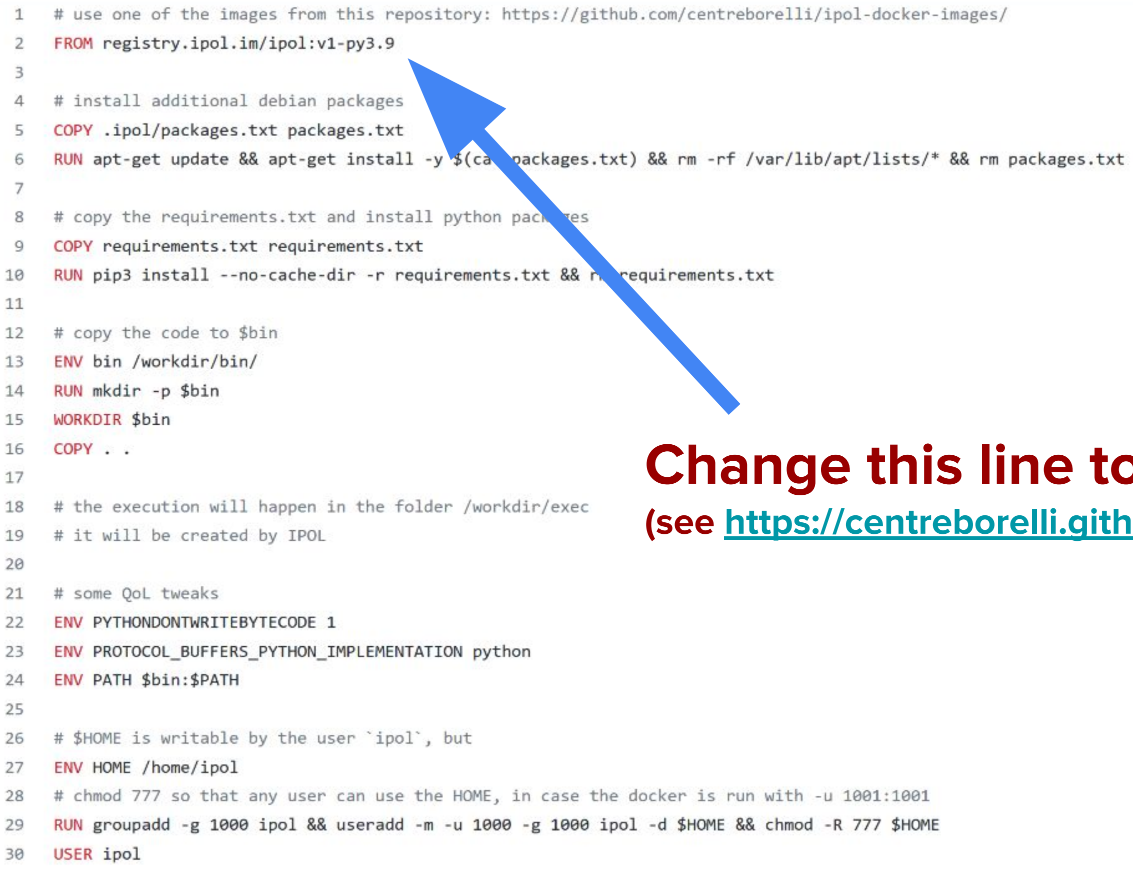
Take a look at https://github.com/mlbriefs/template-python

It's a template for Python code.

The corresponding demo is here:

public demo: https://ipolcore.ipol.im/demo/clientApp/demo.html?id=55555500001
control panel: https://ipolcore.ipol.im/cp2/showDemo?demo_id=55555500001

Dockerfile



```
# use one of the images from this repository: https://github.com/centreborelli/ipol-docker-images/
FROM registry.ipol.im/ipol:v1-py3.9

# install additional debian packages
COPY .ipol/packages.txt packages.txt
RUN apt-get update && apt-get install -y $(cat packages.txt) && rm -rf /var/lib/apt/lists/* && rm packages.txt

# copy the requirements.txt and install python packages
COPY requirements.txt requirements.txt
RUN pip3 install --no-cache-dir -r requirements.txt && rm requirements.txt

# copy the code to $bin
ENV bin /workdir/bin/
RUN mkdir -p $bin
WORKDIR $bin
COPY . .

# the execution will happen in the folder /workdir/exec
# it will be created by IPOL

# some QoL tweaks
ENV PYTHONDONTWRITEBYTECODE 1
ENV PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION python
ENV PATH $bin:$PATH

# $HOME is writable by the user `ipol`, but
ENV HOME /home/ipol
# chmod 777 so that any user can use the HOME, in case the docker is run with -u 1001:1001
RUN groupadd -g 1000 ipol && useradd -m -u 1000 -g 1000 ipol -d $HOME && chmod -R 777 $HOME
USER ipol
```

Change this line to choose the base image

(see https://centreborelli.github.io/MLBriefs/docker-images.html)

Docker image

Contains the instructions to create the environment of the demo (linux packages, pip packages, etc) and to compile the code
Recommended: Choose a docker image from https://centreborelli.github.io/MLBriefs/docker-images.html and put it in the Dockerfile

- Includes a specific Python version (3.7, 3.8 or 3.9) or Octave (6.2.0)
- Flavours with Tensorflow or Pytorch for each Python version
- A list of default packages installed

Modify `requirements.txt` to specify which Python packages should be installed with pip

- For reproducibility, specify all the packages you need even if they are already in the Docker image
- Always specify a full version for each packages (**`numpy==1.22.3`**, not `numpy=1.22.*` nor just `numpy`)
- Unless you need a different version, try to use the versions already packaged in the Docker image (this will help save space storing the images)

If needed, you can add packages to install with apt-get in `.ipol/packages.txt`

- One package per line

For more advanced demos or in specific cases (different language, ...), you can also use a different Docker image.

Clean the inputs / parameters / outputs of the code

Expose the parameters and inputs / outputs filenames:

```
python main.py --input myimage.png --sigma 25 --output result.png
```

In Python, use argparse or other tools.

On IPOL:

```
python $bin/main.py --input input_0.png --sigma $sigma --output output.png
```

Clean up the code

Make sure your code can be executed from anywhere on the filesystem:

```
[user@laptop:~/myproject]$ python main.py
…
[user@laptop:~/myproject]$ cd /tmp
[user@laptop:/tmp]$ python ~/myproject/main.py
…
```

Instead of

```
torch.load('weights.pth')
```

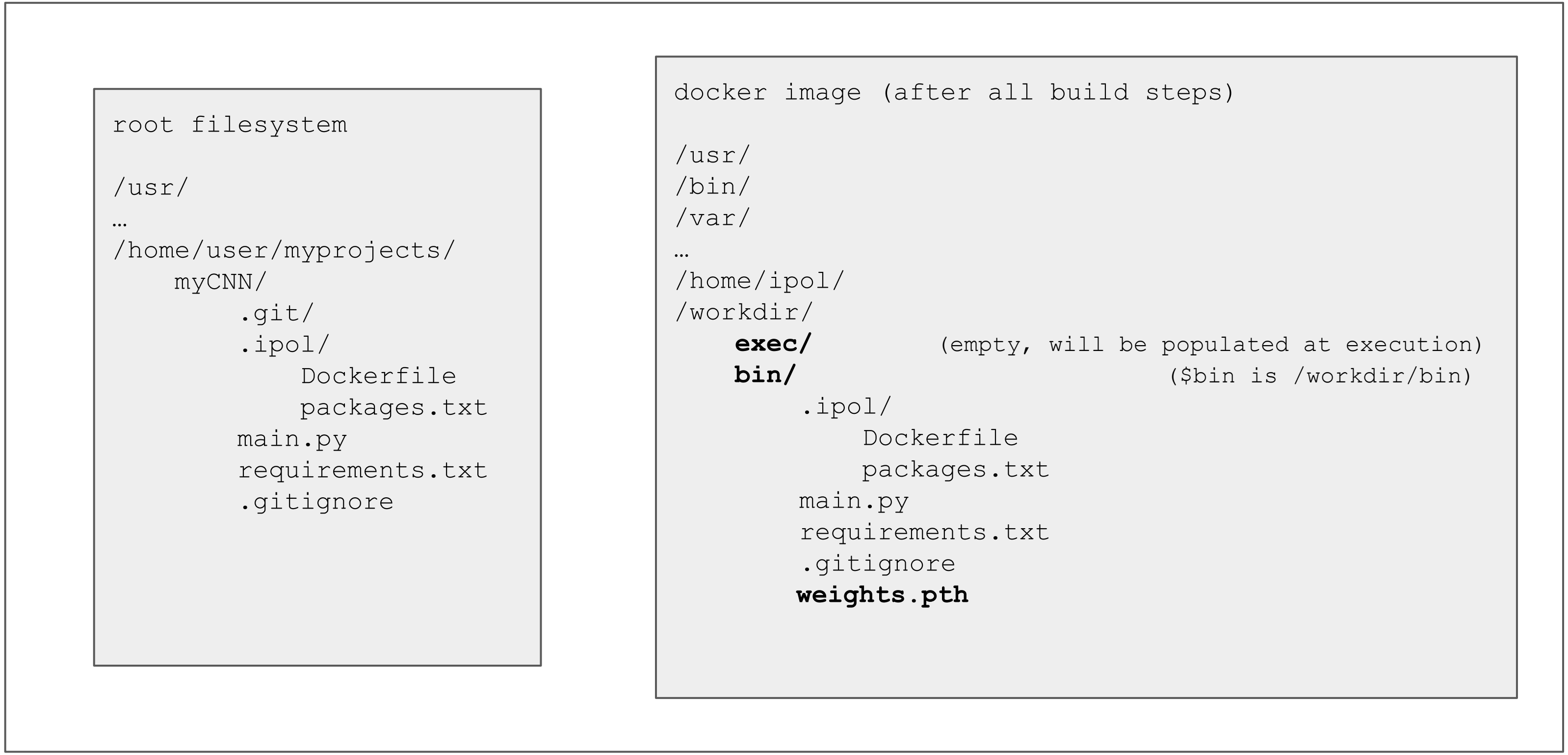
use

```
ROOT = os.path.dirname(os.path.realpath(__file__))
torch.load(os.path.join(ROOT, 'weights.pth'))
```

or expose it as parameter

In IPOL, the code (and weights) is in $bin, but the execution is elsewhere: “`python $bin/main.py` ”

docker image and file system



```
root filesystem

/usr/
…
/home/user/myprojects/
    myCNN/
        .git/
        .ipol/
            Dockerfile
            packages.txt
        main.py
        requirements.txt
        .gitignore
```

```
docker image (after all build steps)

/usr/
/bin/
/var/
…
/home/ipol/
/workdir/
    exec/        (empty, will be populated at execution)
    bin/                       ($bin is /workdir/bin)
        .ipol/
            Dockerfile
            packages.txt
        main.py
        requirements.txt
        .gitignore
        weights.pth
```

Github invitation

You were invited on a Github repository (in the organization *mlbriefs*):
https://github.com/mlbriefs/DEMOID

For now the repository only contains the template, but you can should your code there

Upload your code to your MLBriefs repository:

```
git clone git@github.com:mlbriefs/DEMOID.git
# copy your files
git add your-files      (except large files!)
git commit -m "commit message"
git push
```

make sure to update `requirements.txt`

For large files: upload them to the nextcloud

Upload large files on our nextcloud, **not** on github

Github limits to 100MB per file. After that, the push is rejected and you have to remove the commit from your branch.

Upload your large files (e.g. network weights) to our nextcloud:

> **The link cannot be shared publicly. Please email us if you need access.**
>
> Create a folder with your demo ID (starts with 777777000) first
>
> 1GB max per file

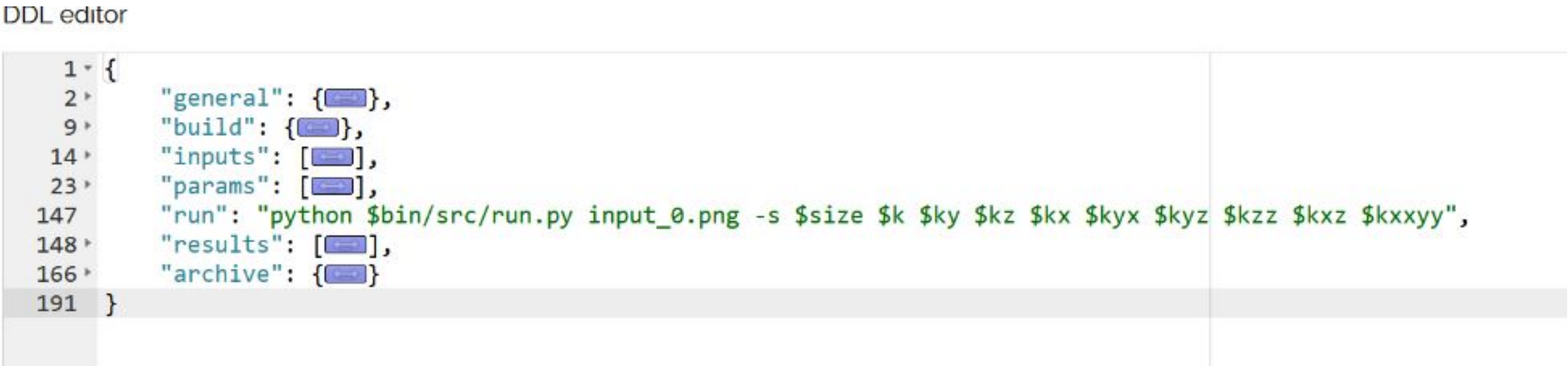
In the Dockerfile:

```
WORKDIR /workdir/bin
RUN wget "https://kiwi.cmla.ens-cachan.fr/index.php/s/yLT6TiyiwXGB54t/download?path=%2F7777000141&files=weights.pth" -O weights.pth
COPY . .
```

> –> it downloads the file to /workdir/bin/filename.pth

Editing the DDL

Editing the DDL: A big JSON file

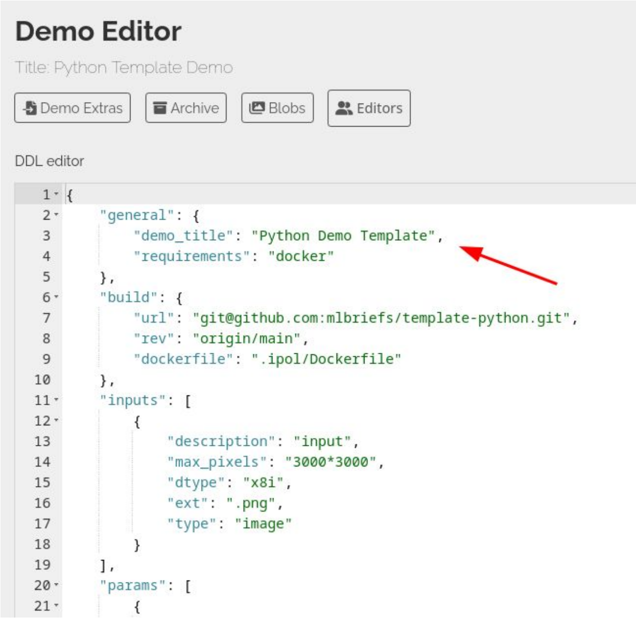


DDL editor

```
{
    "general": {↔},
    "build": {↔},
    "inputs": [↔],
    "params": [↔],
    "run": "python $bin/src/run.py input_0.png -s $size $k $ky $kz $kx $kyx $kyz $kzz $kxz $kxxyy",
    "results": [↔],
    "archive": {↔}
}
```

boucantrin.ovh.hw.ipol.im/static/quentin/doc_ipol.pdf

General, Build

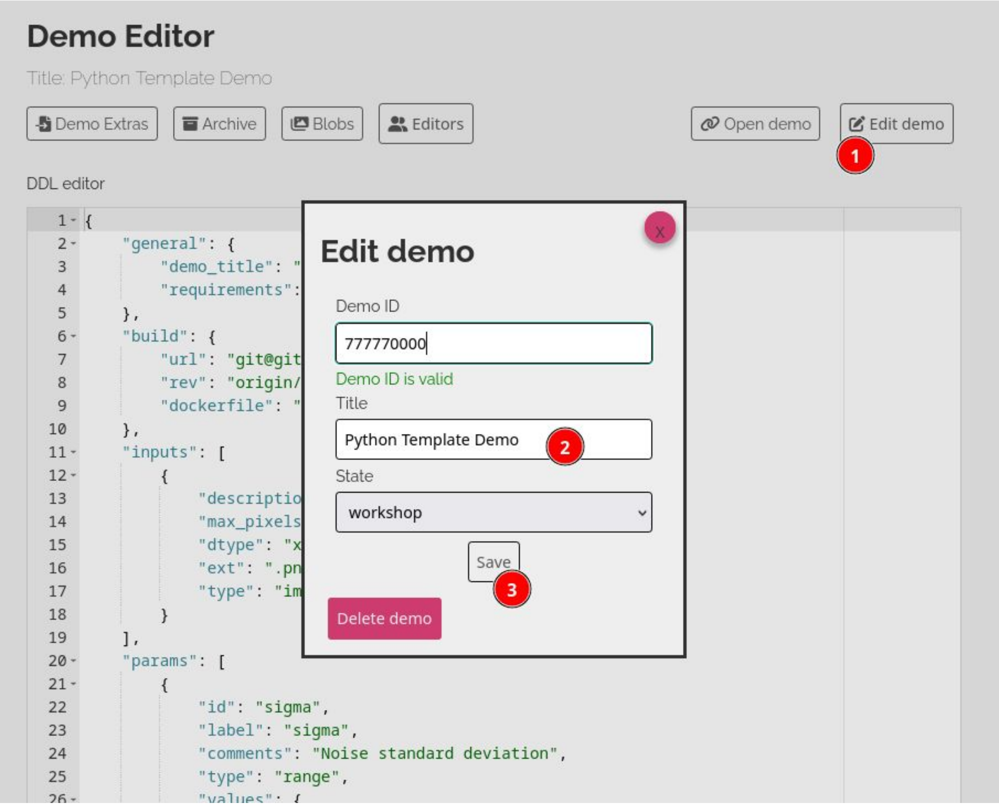


Demo Editor

Title: Python Template Demo

DDL editor

```
{
    "general": {
        "demo_title": "Python Demo Template",
        "requirements": "docker"
    },
    "build": {
        "url": "git@github.com:mlbriefs/template-python.git",
        "rev": "origin/main",
        "dockerfile": ".ipol/Dockerfile"
    },
    "inputs": [
        {
            "description": "input",
            "max_pixels": "3000*3000",
            "dtype": "x8i",
            "ext": ".png",
            "type": "image"
        }
    ],
    "params": [
        {
```



Demo Editor

Title: Python Template Demo

DDL editor

```
{
    "general": {
        "demo_title": "
        "requirements":
    },
    "build": {
        "url": "git@git
        "rev": "origin/
        "dockerfile": "
    },
    "inputs": [
        {
            "descriptio
            "max_pixels
            "dtype": "x
            "ext": ".pn
            "type": "im
        }
    ],
    "params": [
        {
            "id": "sigma",
            "label": "sigma",
            "comments": "Noise standard deviation",
            "type": "range",
            "values": {
```

Edit demo

Demo ID: 777770000

Demo ID is valid

Title: Python Template Demo

State: workshop

Inputs

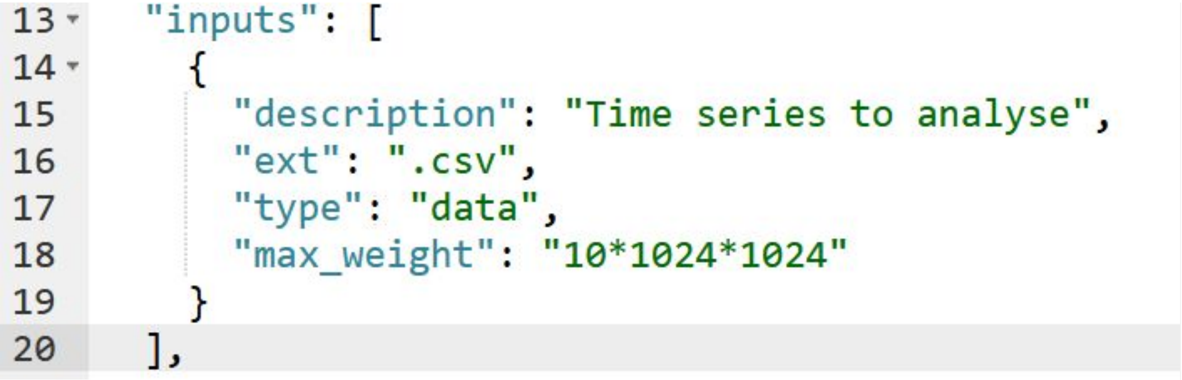
- relative to /workdir/exec/ (current working directory of the process)

- named sequentially input_0.$ext, input_1.$ext, etc
- Retrieve the filename with input_0.$ext, or $input_0

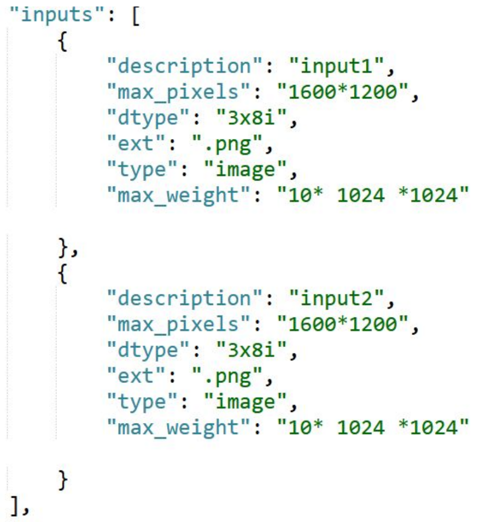
- three supported types:
    - “image”: images (8bits)
        - can be resized by the system if too large (“max_pixels”)
    - “video”: video file format
    - “data”: everything else
        - “ext” defines how the file will be renamed by the system, eg:

          the user upload a file “mydata.txt”
          in the DDL: “ext”: “csv”
          at the start of the execution, the file will be named “input_0.csv”
          (but the content is untouched)
        - No format check for the data type: verify yourself that the user sent the correct formatting

Inputs



```
  "inputs": [
    {
      "description": "Time series to analyse",
      "ext": ".csv",
      "type": "data",
      "max_weight": "10*1024*1024"
    }
  ],
```



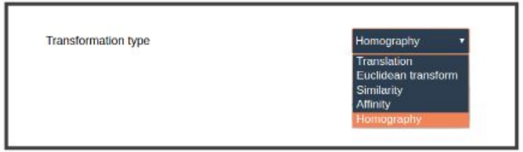
```
"inputs": [
    {
        "description": "input1",
        "max_pixels": "1600*1200",
        "dtype": "3x8i",
        "ext": ".png",
        "type": "image",
        "max_weight": "10* 1024 *1024"

    },
    {
        "description": "input2",
        "max_pixels": "1600*1200",
        "dtype": "3x8i",
        "ext": ".png",
        "type": "image",
        "max_weight": "10* 1024 *1024"

    }
],
```

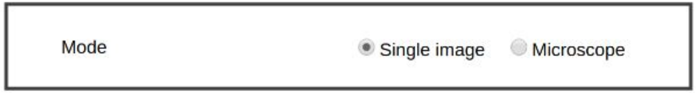
For image inputs, check this template for possibilities:
https://ipolcore.ipol.im/cp2/showDemo?demo_id=55555500003

Parameters



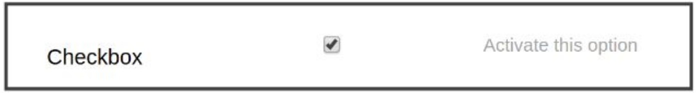
Transformation type: Homography

Figure 2: Selection collapsed example. In this case, the selection offers five options to choose.



Mode ● Single image ○ Microscope

Figure 3: Radio buttons example. The label description is Mode and the parameter offers two radio buttons. The vertical option is disabled.



Checkbox ☑ Activate this option

Figure 5: Checkbox example. This can be used in the demos that need to activate or not an option.

Parameters

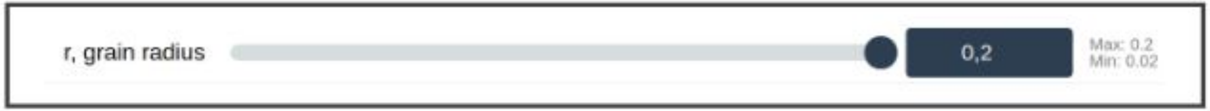
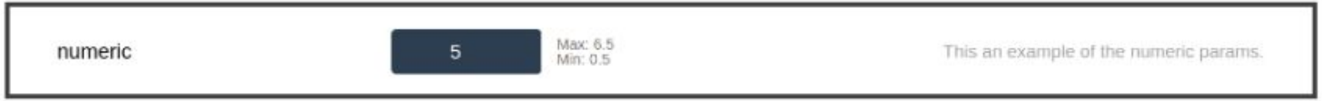

r, grain radius 0,2 Max: 0.2 Min: 0.02

Figure 1: Range type example. It shows a slider with values from 0.02 to 0.2.


numeric 5 Max: 6.5 Min: 0.5

This an example of the numeric params.

Figure 6: Numeric example. The label explains that the sliders below represent matrix values according to the image depicted in the label.

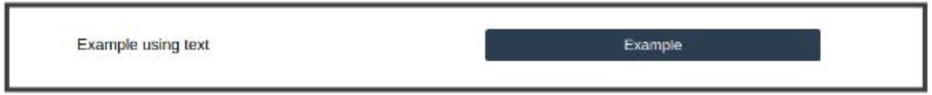
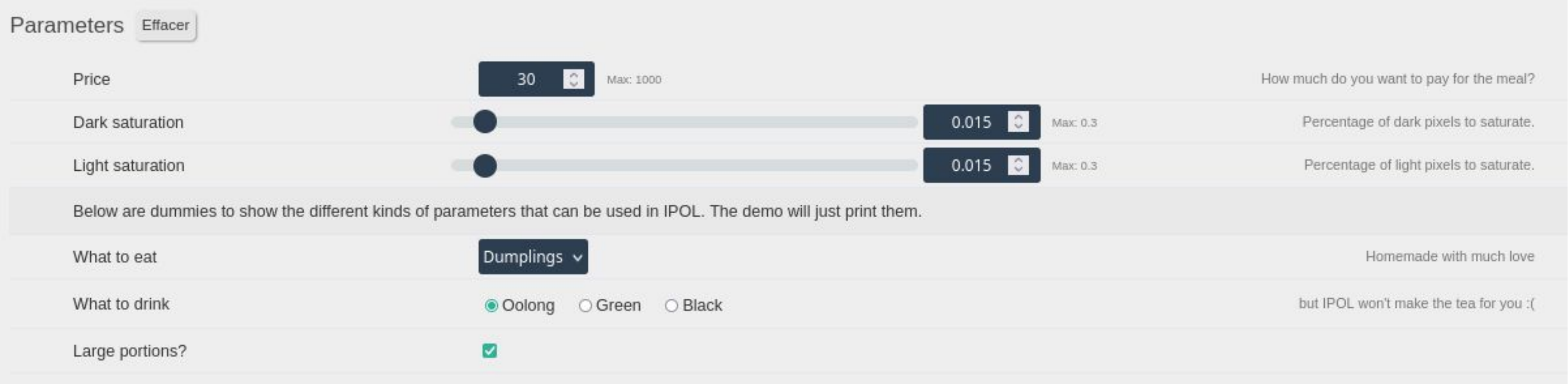

Example using text Example

Figure 7: Text example. The user can write some text as parameter for the demo.

Parameters

https://ipolcore.ipol.im/cp2/showDemo?demo_id=55555500002

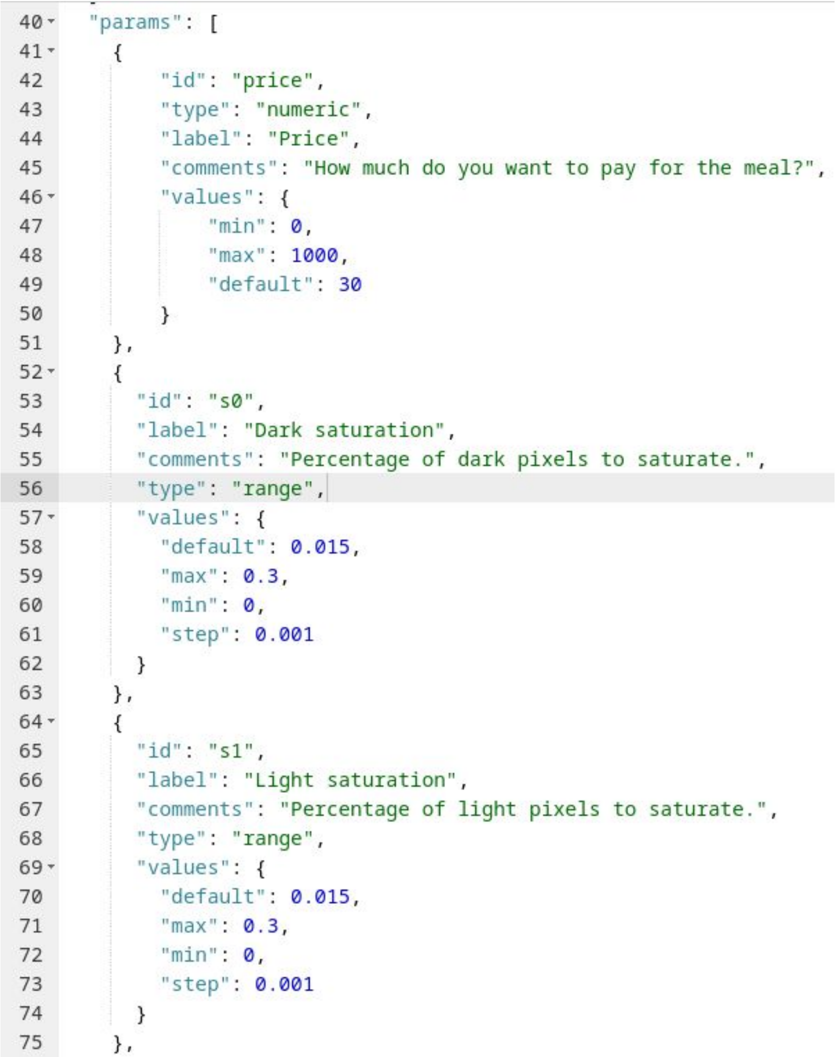


Parameters

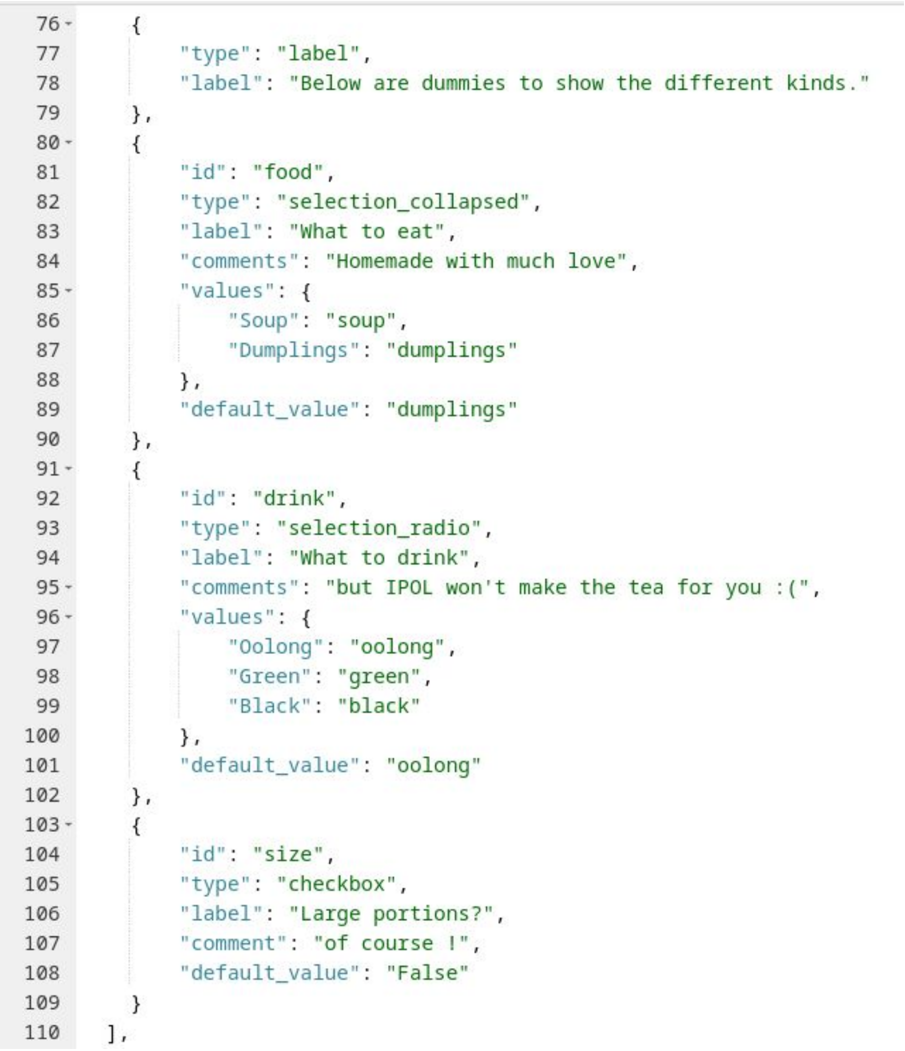
Price	30	Max: 1000	How much do you want to pay for the meal?
Dark saturation	0.015	Max: 0.3	Percentage of dark pixels to saturate.
Light saturation	0.015	Max: 0.3	Percentage of light pixels to saturate.

Below are dummies to show the different kinds of parameters that can be used in IPOL. The demo will just print them.

What to eat	Dumplings	Homemade with much love
What to drink	● Oolong ○ Green ○ Black	but IPOL won't make the tea for you :(
Large portions?	☑	

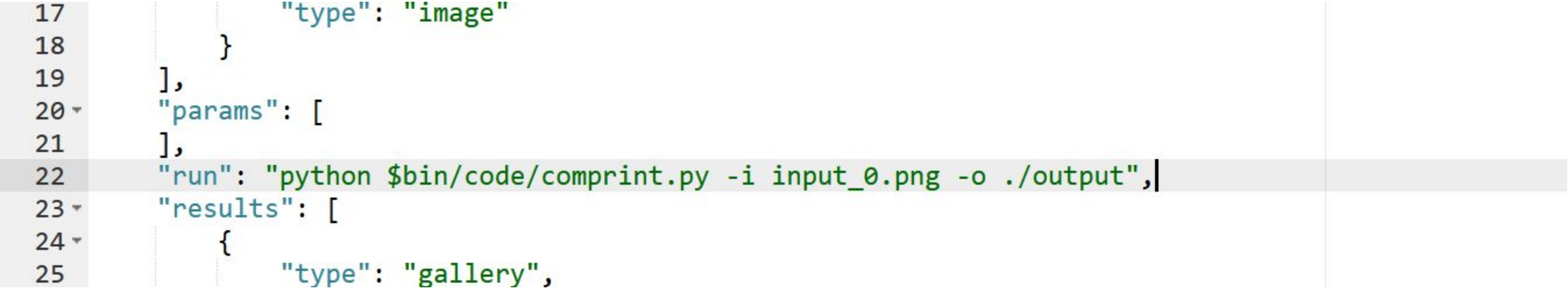


```
  "params": [
    {
        "id": "price",
        "type": "numeric",
        "label": "Price",
        "comments": "How much do you want to pay for the meal?",
        "values": {
            "min": 0,
            "max": 1000,
            "default": 30
        }
    },
    {
      "id": "s0",
      "label": "Dark saturation",
      "comments": "Percentage of dark pixels to saturate.",
      "type": "range",
      "values": {
        "default": 0.015,
        "max": 0.3,
        "min": 0,
        "step": 0.001
      }
    },
    {
      "id": "s1",
      "label": "Light saturation",
      "comments": "Percentage of light pixels to saturate.",
      "type": "range",
      "values": {
        "default": 0.015,
        "max": 0.3,
        "min": 0,
        "step": 0.001
      }
    },
```



```
    {
        "type": "label",
        "label": "Below are dummies to show the different kinds."
    },
    {
        "id": "food",
        "type": "selection_collapsed",
        "label": "What to eat",
        "comments": "Homemade with much love",
        "values": {
            "Soup": "soup",
            "Dumplings": "dumplings"
        },
        "default_value": "dumplings"
    },
    {
        "id": "drink",
        "type": "selection_radio",
        "label": "What to drink",
        "comments": "but IPOL won't make the tea for you :(",
        "values": {
            "Oolong": "oolong",
            "Green": "green",
            "Black": "black"
        },
        "default_value": "oolong"
    },
    {
        "id": "size",
        "type": "checkbox",
        "label": "Large portions?",
        "comment": "of course !",
        "default_value": "False"
    }
  ],
```

Run

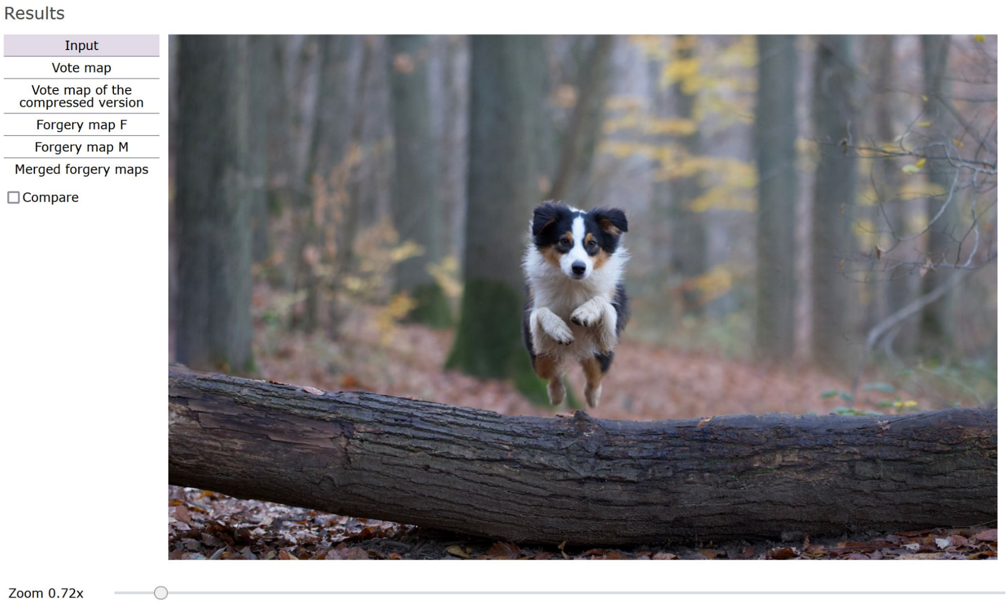


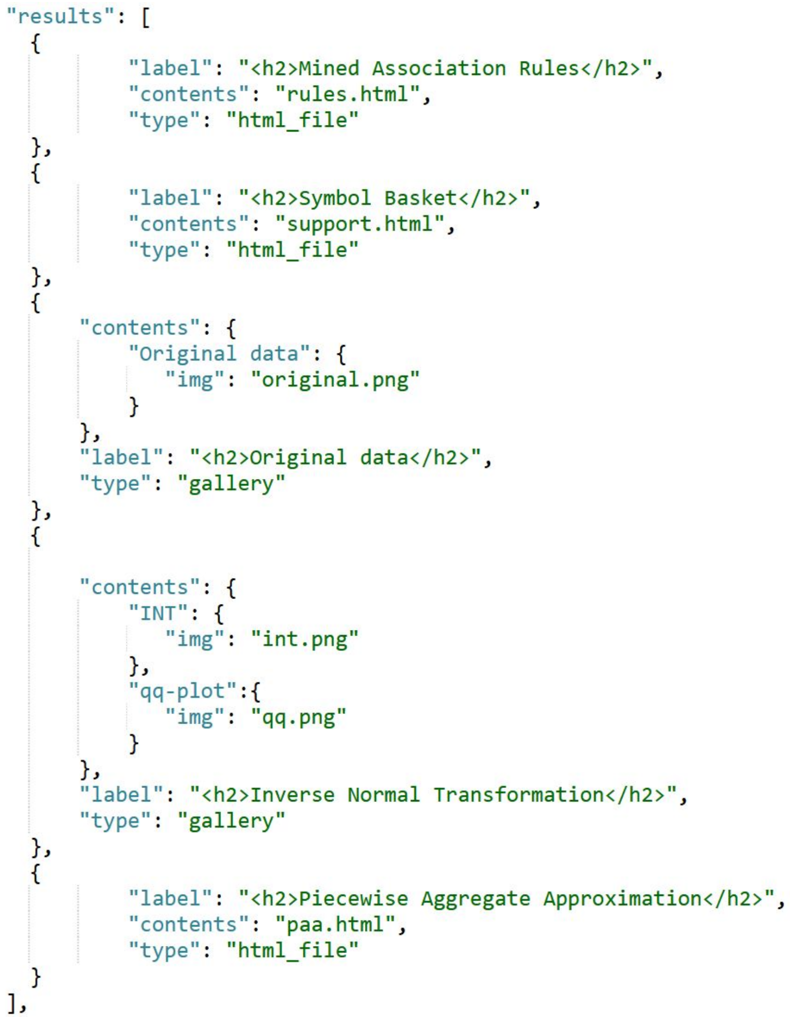
```
            "type": "image"
        }
    ],
    "params": [
    ],
    "run": "python $bin/code/comprint.py -i input_0.png -o ./output",
    "results": [
        {
            "type": "gallery",
```

Results

- should be saved next to the inputs
- save static plots as images and show them with “type”: “gallery”
- save texts to plain files and show them with “type”: “text_file”
- Save other visualizations to HTML and show them with “html_file”
    - Save interactive outputs (plotly, bokeh,...) with mode ‘cdn’!
        - Plotly: Figure.write_html(“output.html”, include_plotlyjs=’cdn’)
        - Bokeh:
            1. from bokeh.plotting import output_file, save
            2. output_file(“output.html”, mode=’cdn’)
            3. p = figure(),...
            4. save(p)
    - Pandas: Dataframe.to_html
    - More complex HTML+JS files can in theory be embedded, communicate with us!

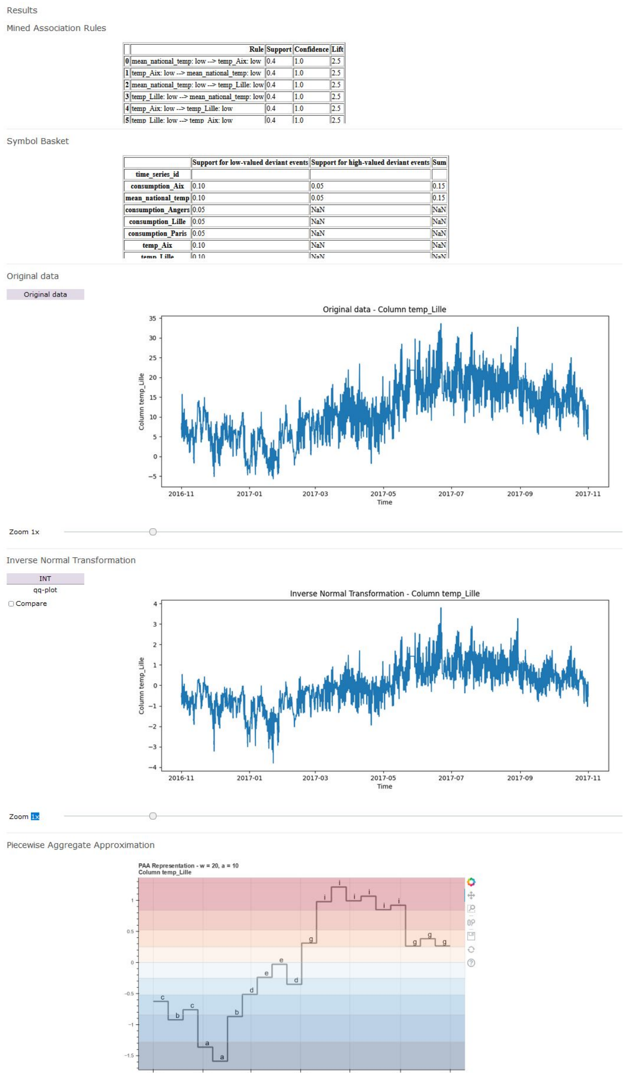
```
"results": [
    {
        "type": "gallery",
        "contents": {
            "Input": {
                "img": "input_0.png"
            },
            "Vote map": {
                "img": "colored_votes.png"
            },
            "Vote map of the compressed version": {
                "img": "colored_votes_jpeg.png"
            },
            "Forgery map F": {
                "img": "mask_f.png"
            },
            "Forgery map M": {
                "img": "mask_m.png"
            },
            "Merged forgery maps": {
                "img": "result_zero.png"
            }
        }
    },
    {
```





```
"results": [
  {
          "label": "<h2>Mined Association Rules</h2>",
          "contents": "rules.html",
          "type": "html_file"
  },
  {
          "label": "<h2>Symbol Basket</h2>",
          "contents": "support.html",
          "type": "html_file"
  },
  {
      "contents": {
          "Original data": {
             "img": "original.png"
          }
      },
      "label": "<h2>Original data</h2>",
      "type": "gallery"
  },
  {

      "contents": {
          "INT": {
             "img": "int.png"
          },
          "qq-plot":{
             "img": "qq.png"
          }
      },
      "label": "<h2>Inverse Normal Transformation</h2>",
      "type": "gallery"
  },
  {
          "label": "<h2>Piecewise Aggregate Approximation</h2>",
          "contents": "paa.html",
          "type": "html_file"
  }
],
```



Results

Mined Association Rules

	Rule	Support	Confidence	Lift
0	mean_national_temp: low --> temp_Aix: low	0.4	1.0	2.5
1	temp_Aix: low --> mean_national_temp: low	0.4	1.0	2.5
2	mean_national_temp: low --> temp_Lille: low	0.4	1.0	2.5
3	temp_Lille: low --> mean_national_temp: low	0.4	1.0	2.5
4	temp_Aix: low --> temp_Lille: low	0.4	1.0	2.5
5	temp_Lille: low --> temp_Aix: low	0.4	1.0	2.5

Symbol Basket

	Support for low-valued deviant events	Support for high-valued deviant events	Sum
time_series_id			
consumption_Aix	0.10	0.05	0.15
mean_national_temp	0.10	0.05	0.15
consumption_Angers	0.05	NaN	NaN
consumption_Lille	0.05	NaN	NaN
consumption_Paris	0.05	NaN	NaN
temp_Aix	0.10	NaN	NaN
temp_Lille	0.10	NaN	NaN

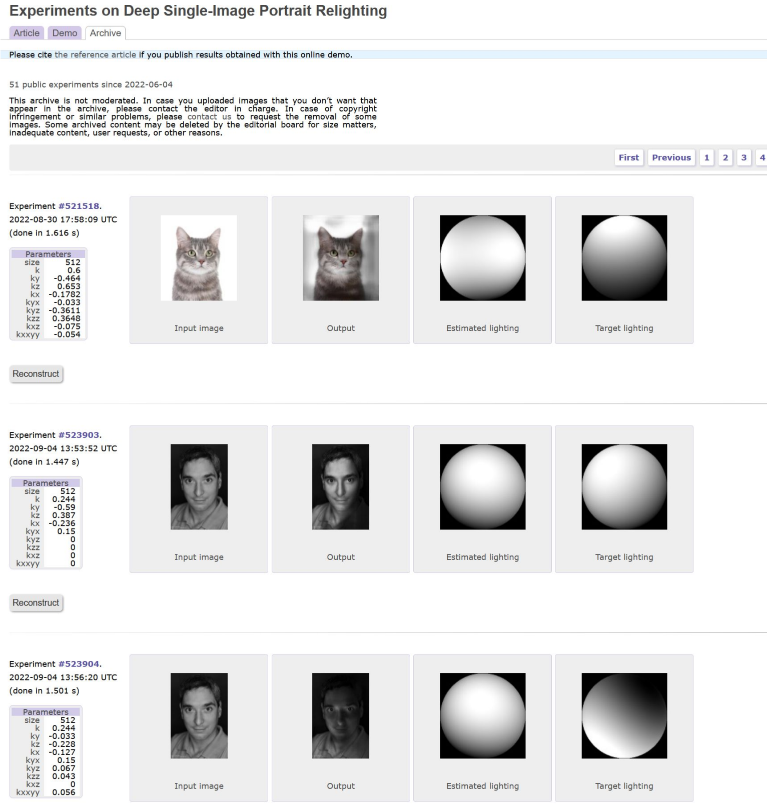
Original data

Inverse Normal Transformation

Piecewise Aggregate Approximation

Archiving results

```
"archive": {
    "enable_reconstruct": true,
    "archive_always": false,
    "files": {
        "input_0.png": "Input image",
        "out_img.png": "Output",
        "out_estimated.png": "Estimated lighting",
        "out_target.png": "Target lighting"
        },
  "params": [
        "size",
        "k",
        "ky",
        "kz",
        "kx",
        "kyx",
        "kyz",
        "kzz",
        "kxz",
        "kxxyy"
    ],
    "info": {
        "run_time": "run time"
    }
},
```



Experiments on Deep Single-Image Portrait Relighting

Please cite the reference article if you publish results obtained with this online demo.

51 public experiments since 2022-06-04

This archive is not moderated. In case you uploaded images that you don't want that appear in the archive, please contact the editor in charge. In case of copyright infringement or similar problems, please contact us to request the removal of some images. Some archived content may be deleted by the editorial board for size matters, inadequate content, user requests, or other reasons.

Experiment #521518.
2022-08-30 17:58:09 UTC
(done in 1.616 s)

Parameters	
size	512
k	0.6
ky	-0.464
kz	0.653
kx	-0.1782
kyx	-0.033
kyz	-0.3611
kzz	0.3648
kxz	-0.075
kxxyy	-0.054

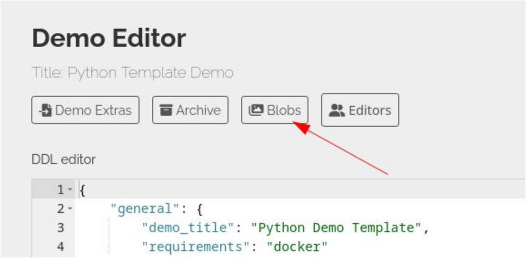
Experiment #523903.
2022-09-04 13:53:52 UTC
(done in 1.447 s)

Parameters	
size	512
k	0.244
ky	-0.59
kz	0.387
kx	-0.236
kyx	0.15
kyz	0
kzz	0
kxz	0
kxxyy	0

Experiment #523904.
2022-09-04 13:56:20 UTC
(done in 1.501 s)

Parameters	
size	512
k	0.244
ky	-0.033
kz	-0.228
kx	-0.127
kyx	0.15
kyz	0.067
kzz	0.043
kxz	0
kxxyy	0.056

Blobs



Demo Editor

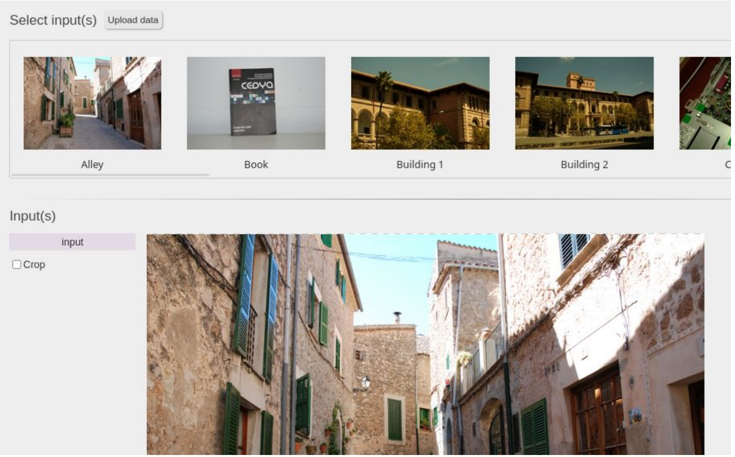
Title: Python Template Demo

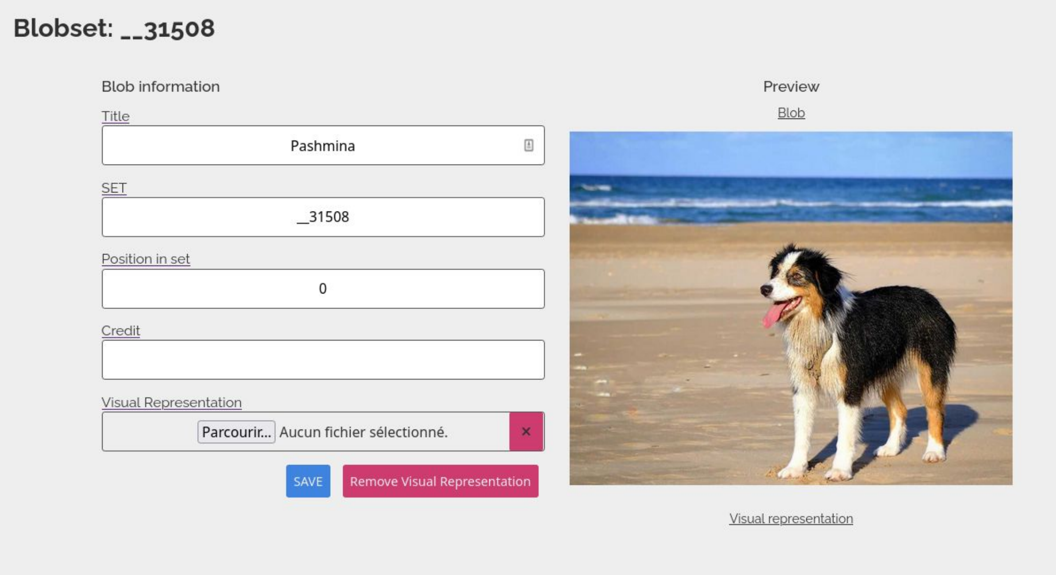
DDL editor

```
{
    "general": {
        "demo_title": "Python Demo Template",
        "requirements": "docker"
```

use a blob template if possible

if the file is not an image, prepare a visual representation to illustrate the data





Blobset: __31508

Blob information

Title: Pashmina

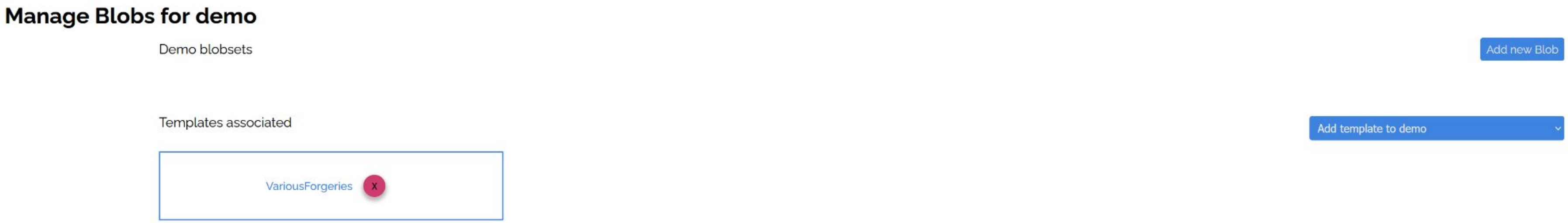
SET: __31508

Position in set: 0

Credit:

Visual Representation: Parcourir... Aucun fichier sélectionné.

Blobs and templates



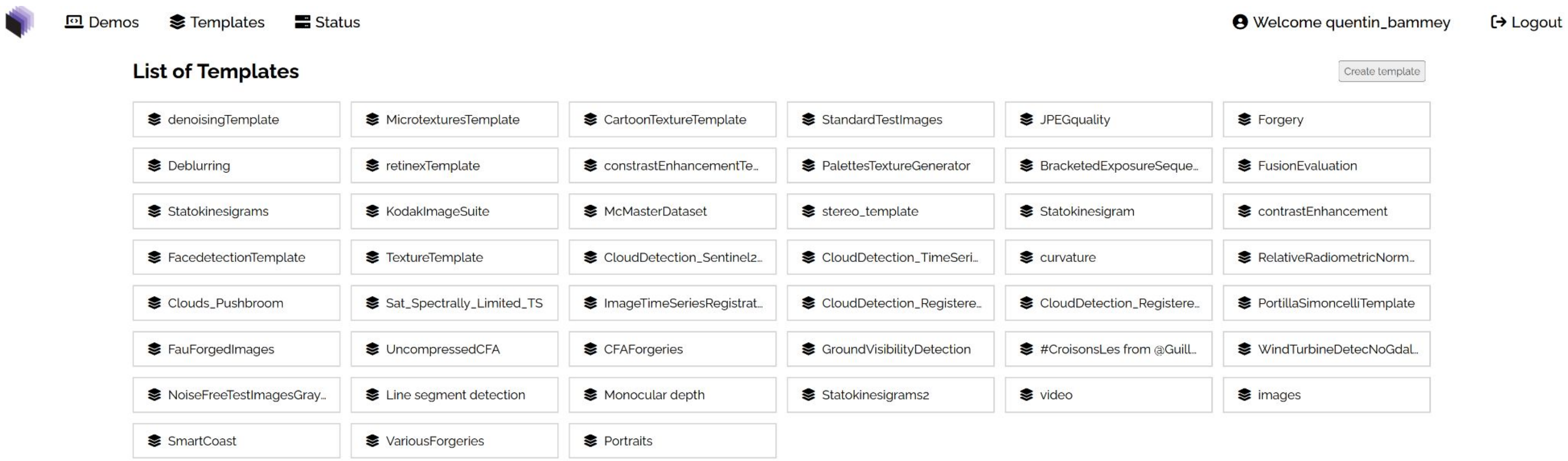
Manage Blobs for demo

Demo blobsets

Templates associated

VariousForgeries

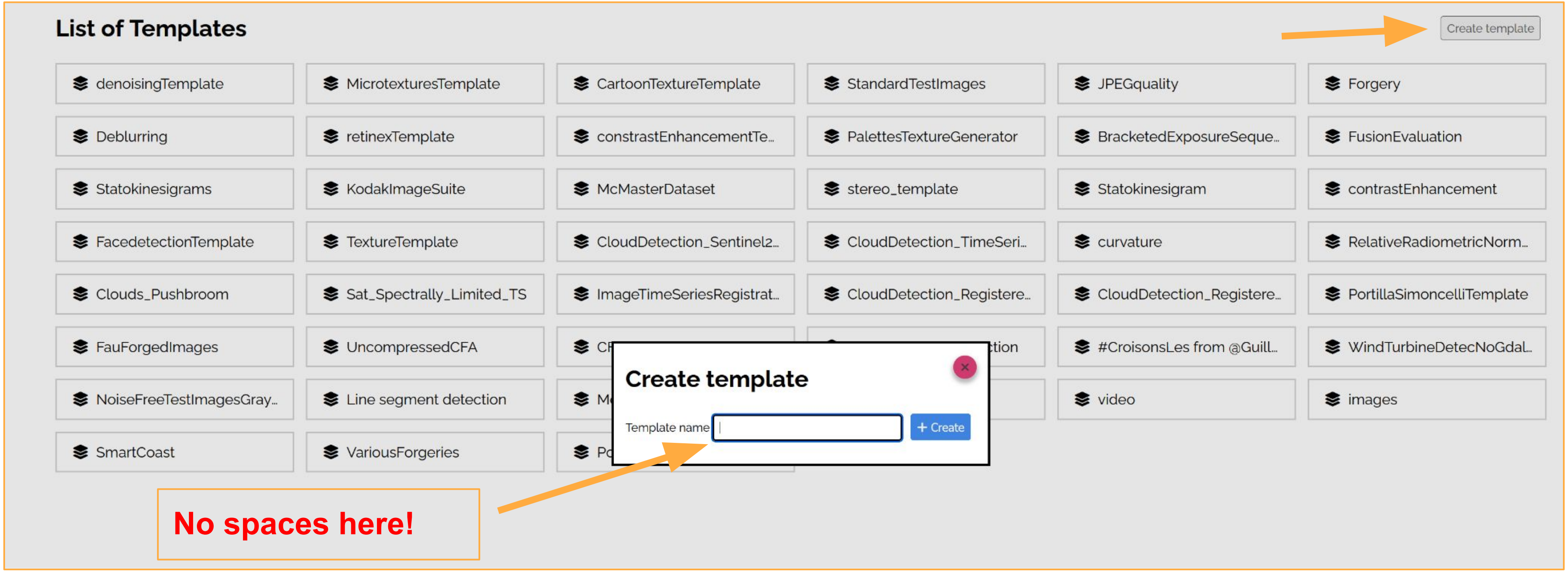
Templates



List of Templates

- denoisingTemplate
- MicrotexturesTemplate
- CartoonTextureTemplate
- StandardTestImages
- JPEGquality
- Forgery
- Deblurring
- retinexTemplate
- constrastEnhancementTe...
- PalettesTextureGenerator
- BracketedExposureSeque...
- FusionEvaluation
- Statokinesigrams
- KodakImageSuite
- McMasterDataset
- stereo_template
- Statokinesigram
- contrastEnhancement
- FacedetectionTemplate
- TextureTemplate
- CloudDetection_Sentinel2...
- CloudDetection_TimeSeri...
- curvature
- RelativeRadiometricNorm...
- Clouds_Pushbroom
- Sat_Spectrally_Limited_TS
- ImageTimeSeriesRegistrat...
- CloudDetection_Registere...
- CloudDetection_Registere...
- PortillaSimoncelliTemplate
- FauForgedImages
- UncompressedCFA
- CFAForgeries
- GroundVisibilityDetection
- #CroisonsLes from @Guill...
- WindTurbineDetecNoGdal...
- NoiseFreeTestImagesGray...
- Line segment detection
- Monocular depth
- Statokinesigrams2
- video
- images
- SmartCoast
- VariousForgeries
- Portraits

Templates



List of Templates

Create template

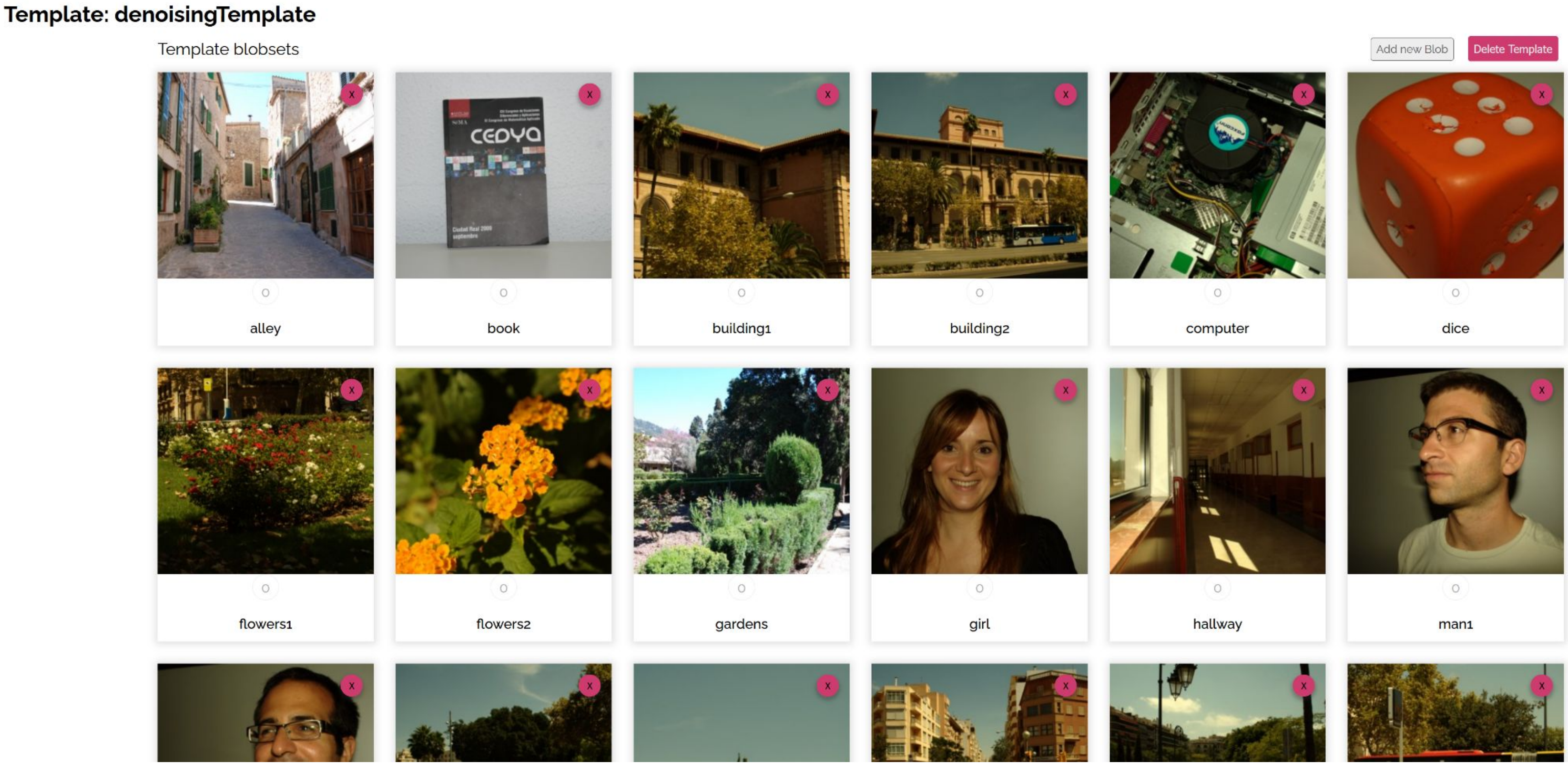
- denoisingTemplate
- MicrotexturesTemplate
- CartoonTextureTemplate
- StandardTestImages
- JPEGquality
- Forgery
- Deblurring
- retinexTemplate
- constrastEnhancementTe...
- PalettesTextureGenerator
- BracketedExposureSeque...
- FusionEvaluation
- Statokinesigrams
- KodakImageSuite
- McMasterDataset
- stereo_template
- Statokinesigram
- contrastEnhancement
- FacedetectionTemplate
- TextureTemplate
- CloudDetection_Sentinel2...
- CloudDetection_TimeSeri...
- curvature
- RelativeRadiometricNorm...
- Clouds_Pushbroom
- Sat_Spectrally_Limited_TS
- ImageTimeSeriesRegistrat...
- CloudDetection_Registere...
- CloudDetection_Registere...
- PortillaSimoncelliTemplate
- FauForgedImages
- UncompressedCFA
- CF
- ction
- #CroisonsLes from @Guill...
- WindTurbineDetecNoGdal...
- NoiseFreeTestImagesGray...
- Line segment detection
- M
- video
- images
- SmartCoast
- VariousForgeries
- P

Create template

Template name

No spaces here!

Templates



Template: denoisingTemplate

Template blobsets