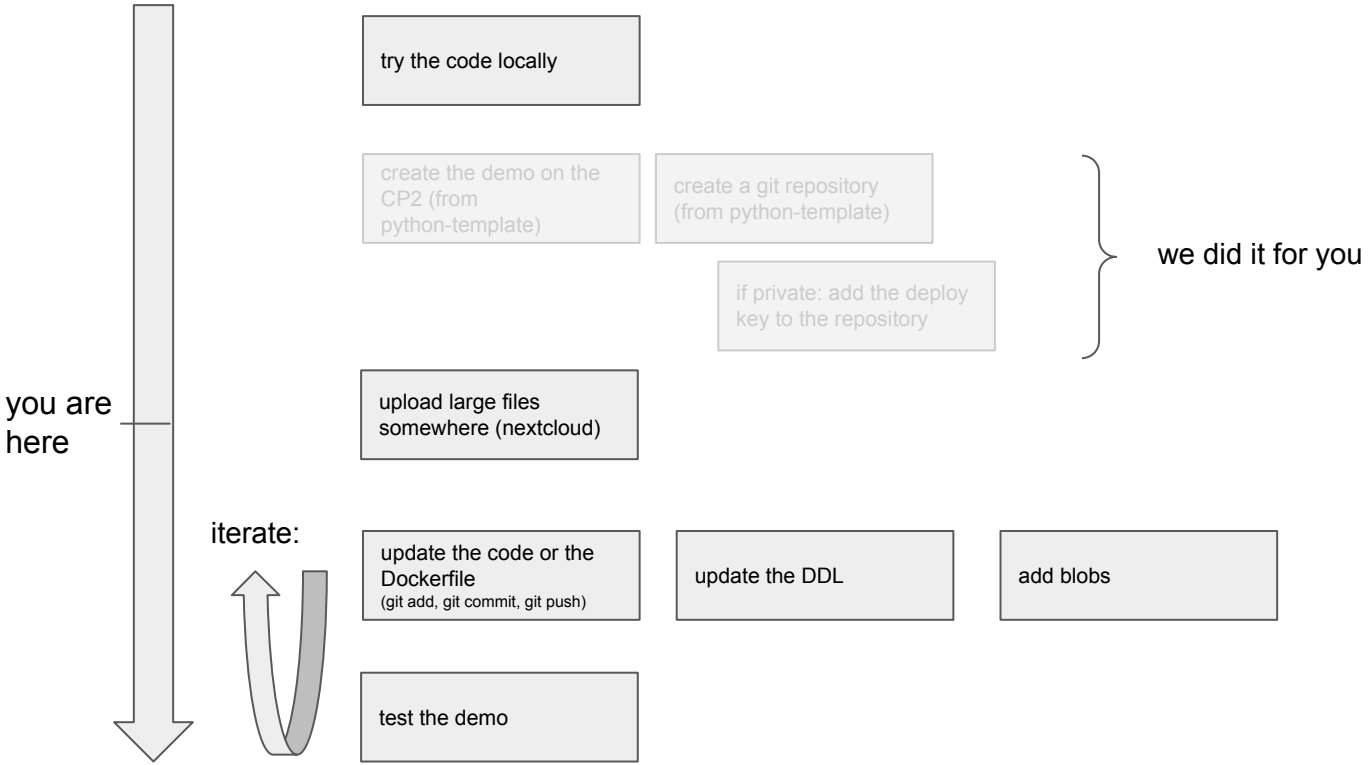
Day 2

From a local code to a MLBriefs demo

The MLBriefs workflow



Github

You should already have a repository on the mlbriefs github organization:

https://github.com/mlbriefs/DEMOID

It is pre-filled with the Python template

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

Python template

Recommended: use the Python template:

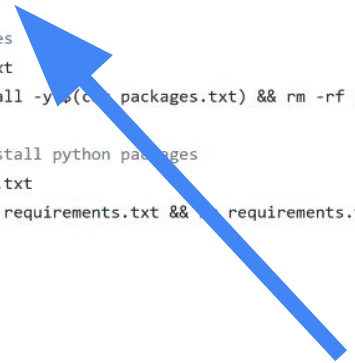
https://github.com/mlbriefs/template-python

It contains:

- `.ipol/Dockerfile`: from which the Docker image is built
- `requirements.txt`: Python libraries to install with pip
- `.ipol/packages.txt`: to install specific packages

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

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/MLBriefs2022/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.

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:

> https://kiwi.cmla.ens-cachan.fr/index.php/s/yLT6TiyiwXGB54t
>
> 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=%2F77777000141&files=we
ights.pth" -O weights.pth
COPY . .
```

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

docker image and file system

```
root filesystem

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

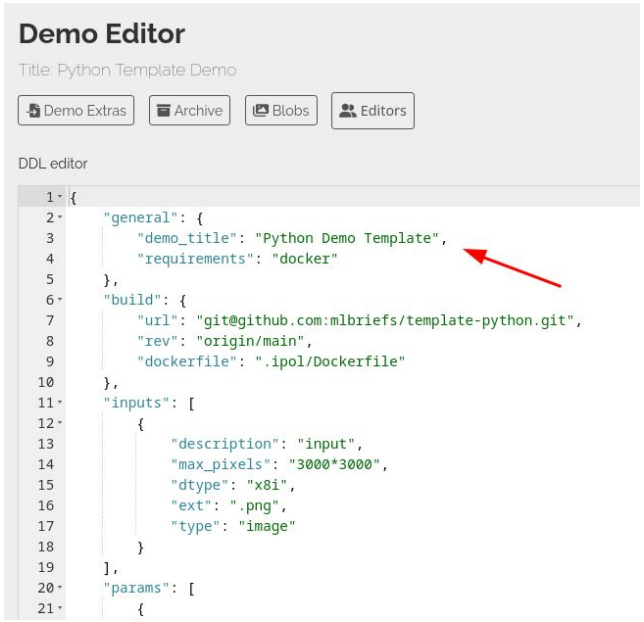
```
docker image (after all build steps)

/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
```

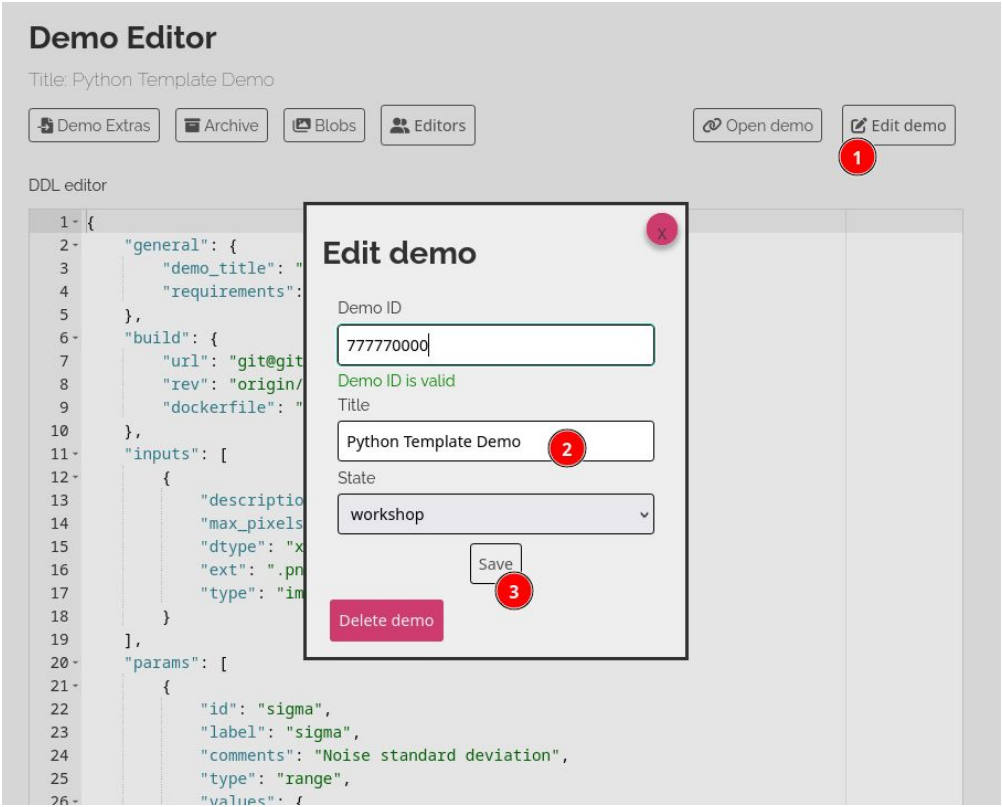
Editing the DDL

DDL editor

```
1   {
2       "general": {…},
9       "build": {…},
14      "inputs": […],
23      "params": […],
147     "run": "python $bin/src/run.py input_0.png -s $size $k $ky $kz $kx $kyx $kyz $kzz $kxz $kxxyy",
148     "results": […],
166     "archive": {…}
191 }
```

General, Build





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

http://dev.ipol.im/~tina/ipol/ddl.pdf#section.4

Inputs

```
    {
      "description": "input",
      "dtype": "x8i",
      "ext": ".png",
      "max_pixels": "10000*10000",
      "type": "image"
    }
```

```
"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"

    }
],
```

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

Parameters

http://dev.ipol.im/~tina/ipol/ddl.pdf#section.5

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

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

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

Parameters

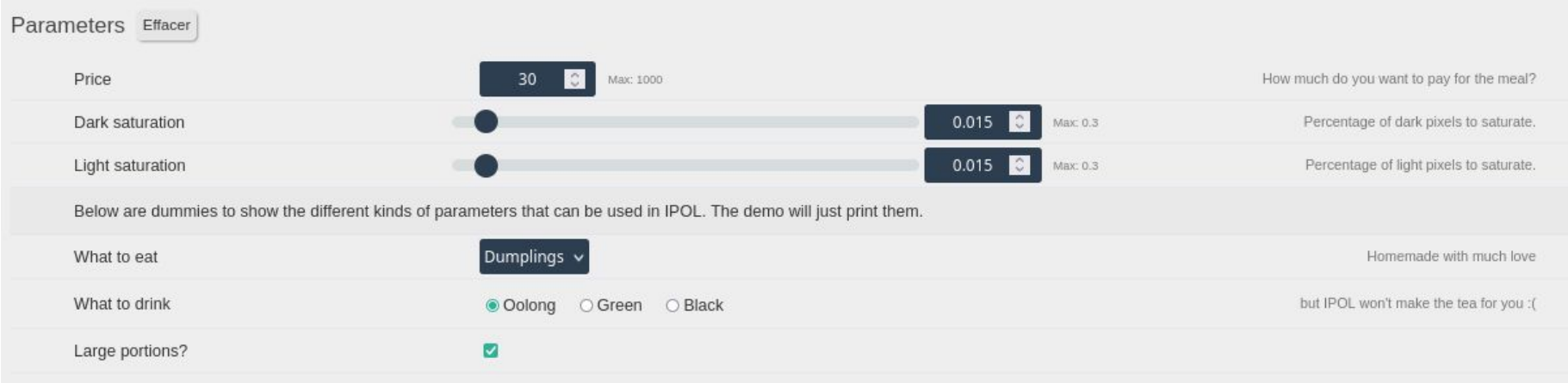
Figure 1: Range type example. It shows a slider with values from 0.02 to 0.2.

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

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

Parameters

https://ipolcore.ipol.im/cp2/showDemo?demo_id=555553108203 7&title=



```
"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
- http://dev.ipol.im/~tina/ipol/ddl.pdf#section.8
- 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

```
"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

Input
Vote map
Vote map of the compressed version
Forgery map F
Forgery map M
Merged forgery maps
Compare

Zoom 0.72x

```
"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

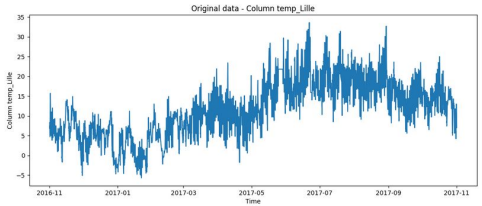
	Rule	Support	Confidence	Lift
0	mean_national_temp: low --> temp_Aix: low	0.4	1.0	2.5
1	temp_Aix: low --> mean_national_temp: low	0.4	1.0	2.5
2	mean_national_temp: low --> temp_Lille: low	0.4	1.0	2.5
3	temp_Lille: low --> mean_national_temp: low	0.4	1.0	2.5
4	temp_Aix: low --> temp_Lille: low	0.4	1.0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

Original data



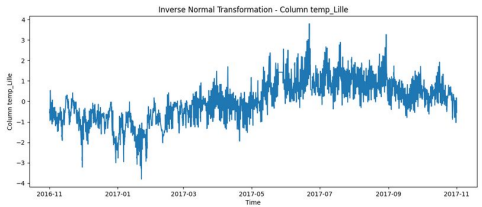
Zoom 1x

Inverse Normal Transformation

INT

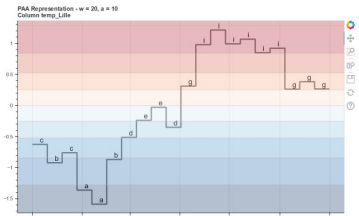
qq-plot

Compare



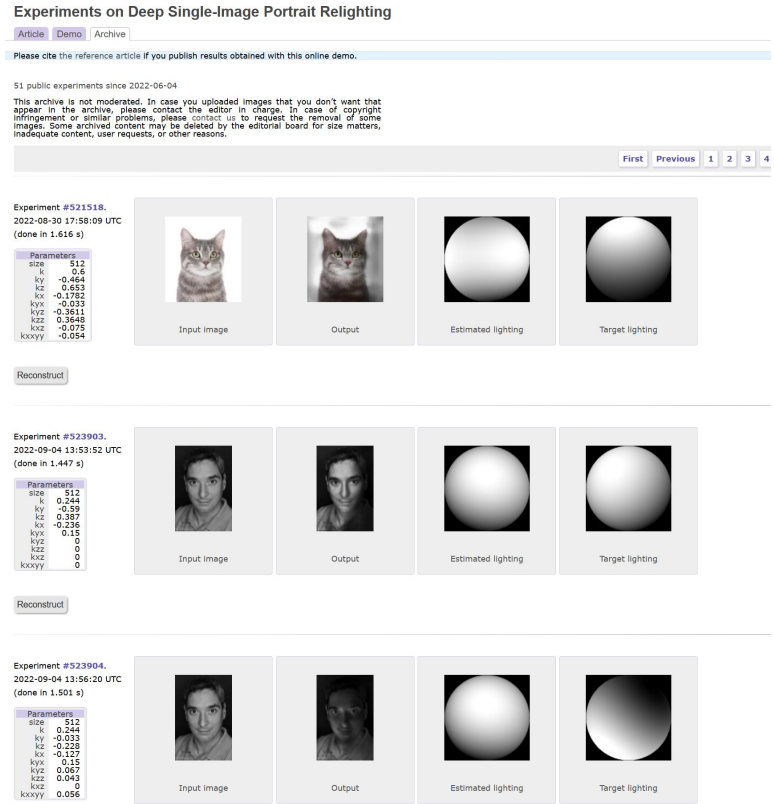
Zoom 1x

Piecewise Aggregate Approximation



Archiving results

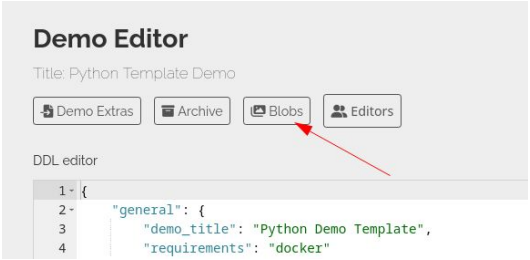
```
"archive": {
    "enable_reconstruct": true,
    "archive_always": false,
    "files": {
        "input_0.png": "Input image",
        "out_img.png": "Output",
        "out_estimated.png": "Estimated lighting",
        "out_target.png": "Target lighting"
        },
 "params": [
        "size",
 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

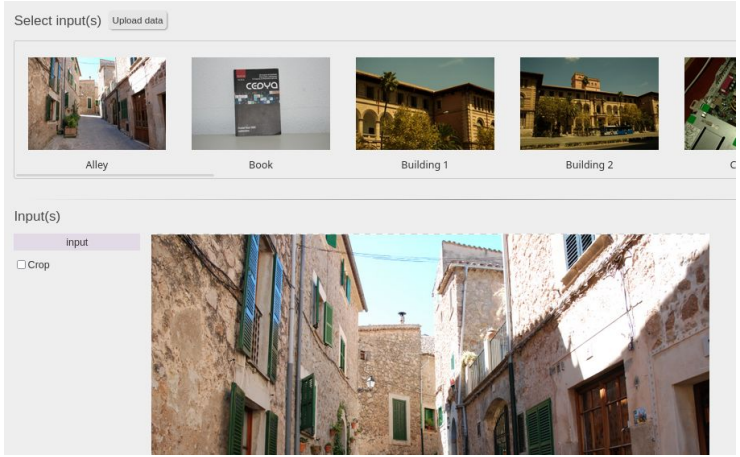


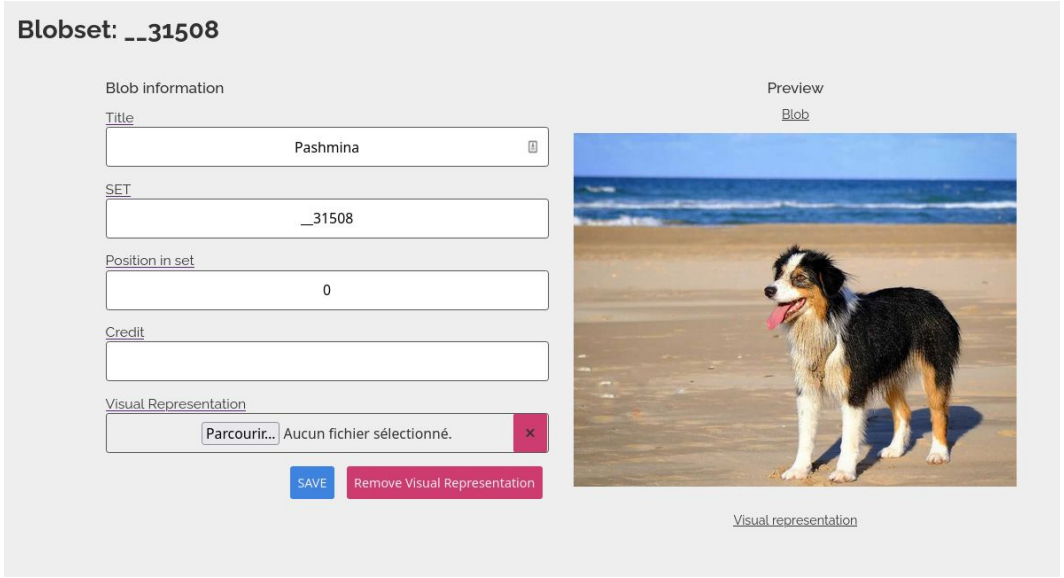
Blobs

use a blob template if possible

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





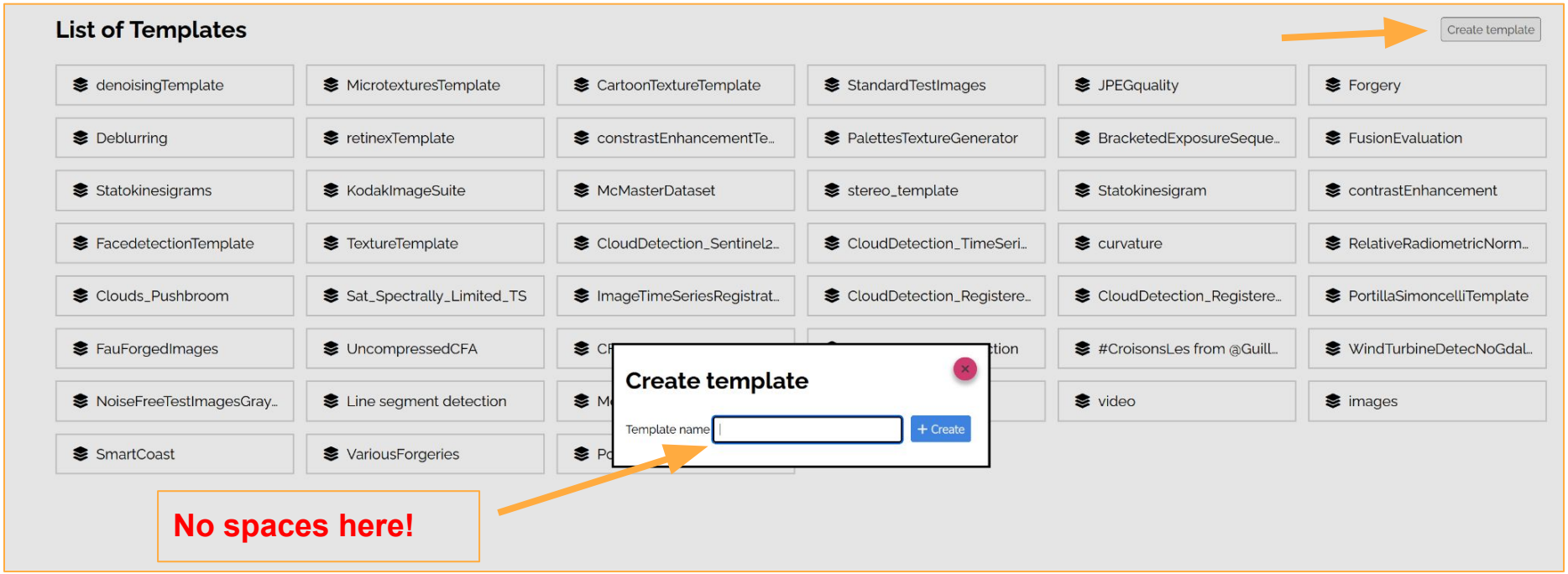


Blobs and templates

Templates


Demos
Templates
Status
Welcome quentin_bammey
Logout
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
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
#CroisonsLes from @Guill...
WindTurbineDetecNoGdal...
NoiseFreeTestImagesGray...
Line segment detection
video
images
SmartCoast
VariousForgeries
Create template
Template name
+ Create
No spaces here!

Templates

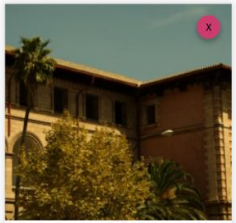
Template: denoisingTemplate

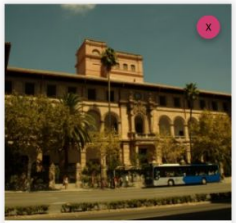
Template blobsets

Add new Blob | Delete Template


0
alley

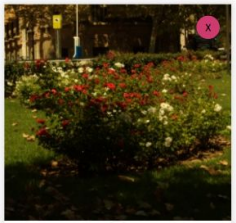
0
book


0
building1

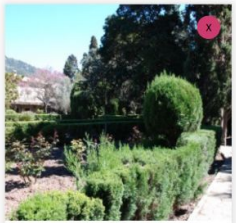

0
building2


0
computer

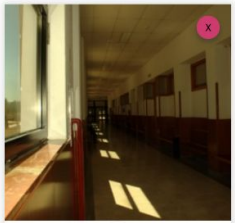

0
dice

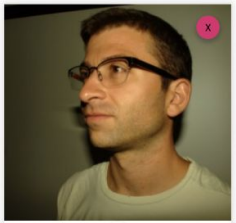

0
flowers1


0
flowers2


0
gardens


0
girl


0
hallway


0
man1