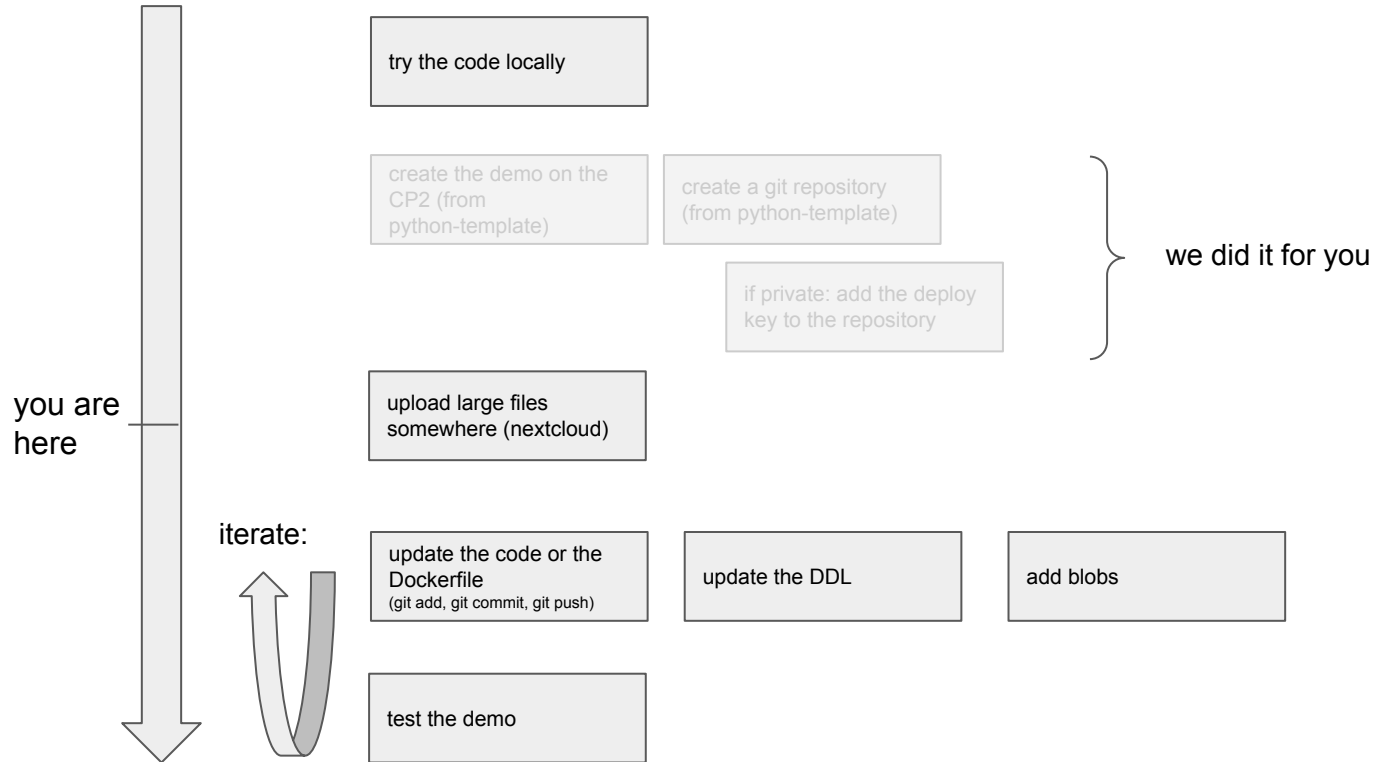


# Day 2

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/DEM0ID>

It is pre-filled with the Python template

Upload your code to your MLBriefs repository:

```
git clone git@github.com:mlbriefs/DEM0ID.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

```
1 # use one of the images from this repository: https://github.com/centreborelli/ipol-docker-images/
2 FROM registry.ipol.im/ipol:v1-py3.9
3
4 # install additional debian packages
5 COPY .ipol/packages.txt packages.txt
6 RUN apt-get update && apt-get install -y $(cat packages.txt) && rm -rf /var/lib/apt/lists/* && rm packages.txt
7
8 # copy the requirements.txt and install python packages
9 COPY requirements.txt requirements.txt
10 RUN pip3 install --no-cache-dir -r requirements.txt && rm requirements.txt
11
12 # copy the code to $bin
13 ENV bin /workdir/bin/
14 RUN mkdir -p $bin
15 WORKDIR $bin
16 COPY . .
17
18 # the execution will happen in the folder /workdir/exec
19 # it will be created by IPOL
20
21 # some QoL tweaks
22 ENV PYTHONDONTWRITEBYTECODE 1
23 ENV PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION python
24 ENV PATH $bin:$PATH
25
26 # $HOME is writable by the user `ipol`, but
27 ENV HOME /home/ipol
28 # chmod 777 so that any user can use the HOME, in case the docker is run with -u 1001:1001
29 RUN groupadd -g 1000 ipol && useradd -m -u 1000 -g 1000 ipol -d $HOME && chmod -R 777 $HOME
30 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/yLT6TiywXGB54t>

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/yLT6TiywXGB54t/download?path=%2F777777000141&files=weights.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

## Demo Editor

Title: Python Template Demo

[Demo Extras](#) [Archive](#) [Blobs](#) [Editors](#)

DDL editor

```
1 {  
2   "general": {  
3     "demo_title": "Python Demo Template",  
4     "requirements": "docker"  
5   },  
6   "build": {  
7     "url": "git@github.com:mlbriefs/template-python.git",  
8     "rev": "origin/main",  
9     "dockerfile": ".ipol/Dockerfile"  
10  },  
11  "inputs": [  
12    {  
13      "description": "input",  
14      "max_pixels": "3000*3000",  
15      "dtype": "x8i",  
16      "ext": ".png",  
17      "type": "image"  
18    }  
19  ],  
20  "params": [  
21    {
```

## Demo Editor

Title: Python Template Demo

[Demo Extras](#) [Archive](#) [Blobs](#) [Editors](#)

[Open demo](#) [Edit demo](#)

DDL editor

```
1 {  
2   "general": {  
3     "demo_title": "  
4     "requirements": "  
5   },  
6   "build": {  
7     "url": "git@git  
8     "rev": "origin/  
9     "dockerfile": "  
10  },  
11  "inputs": [  
12    {  
13      "descriptio  
14      "max_pixels  
15      "dtype": "x  
16      "ext": ".pn  
17      "type": "im  
18    }  
19  ],  
20  "params": [  
21    {  
22      "id": "sigma",  
23      "label": "sigma",  
24      "comments": "Noise standard deviation",  
25      "type": "range",  
26      "values": {
```

### Edit demo

Demo ID  
  
Demo ID is valid

Title

State

[Save](#) [Delete demo](#)

# 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

```
15 {  
16   "description": "input",  
17   "dtype": "x8i",  
18   "ext": ".png",  
19   "max_pixels": "10000*10000",  
20   "type": "image"  
21 }
```

```
13 "inputs": [  
14   {  
15     "description": "Time series to analyse",  
16     "ext": ".csv",  
17     "type": "data",  
18     "max_weight": "10*1024*1024"  
19   }  
20 ],
```

```
"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"  
  }  
],
```

# 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

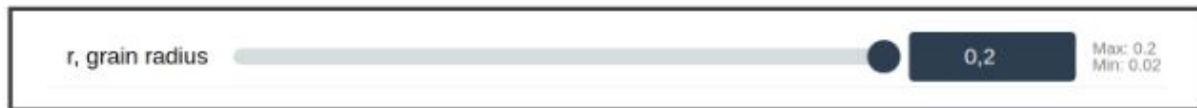


Figure 1: Range type example. It shows a slider with values from 0.02 to 0.2.



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



A text input field with a light blue border. Inside the field, the text "Example using text" is displayed on the left, and a dark blue button with the text "Example" is on the right.

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

# Parameters

[https://ipolcore.ipol.im/cp2/showDemo?demo\\_id=5555531082037&title=](https://ipolcore.ipol.im/cp2/showDemo?demo_id=5555531082037&title=)

Parameters

Effacer

Price	<div>30</div> <div>Max: 1000</div>	How much do you want to pay for the meal?
Dark saturation	<div><div></div></div> <div>0.015</div> <div>Max: 0.3</div>	Percentage of dark pixels to saturate.
Light saturation	<div><div></div></div> <div>0.015</div> <div>Max: 0.3</div>	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	<div>Dumplings</div>	Homemade with much love
What to drink	<div><input checked="" type="radio"/> Oolong <input type="radio"/> Green <input type="radio"/> Black</div>	but IPOL won't make the tea for you :(
Large portions?	<div><input checked="" type="checkbox"/></div>	

```

40 ~ "params": [
41 ~   {
42 ~     "id": "price",
43 ~     "type": "numeric",
44 ~     "label": "Price",
45 ~     "comments": "How much do you want to pay for the meal?",
46 ~     "values": {
47 ~       "min": 0,
48 ~       "max": 1000,
49 ~       "default": 30
50 ~     }
51 ~   },
52 ~   {
53 ~     "id": "s0",
54 ~     "label": "Dark saturation",
55 ~     "comments": "Percentage of dark pixels to saturate.",
56 ~     "type": "range",
57 ~     "values": {
58 ~       "default": 0.015,
59 ~       "max": 0.3,
60 ~       "min": 0,
61 ~       "step": 0.001
62 ~     }
63 ~   },
64 ~   {
65 ~     "id": "s1",
66 ~     "label": "Light saturation",
67 ~     "comments": "Percentage of light pixels to saturate.",
68 ~     "type": "range",
69 ~     "values": {
70 ~       "default": 0.015,
71 ~       "max": 0.3,
72 ~       "min": 0,
73 ~       "step": 0.001
74 ~     }
75 ~   },

```

```

76 ~   {
77 ~     "type": "label",
78 ~     "label": "Below are dummies to show the different kinds."
79 ~   },
80 ~   {
81 ~     "id": "food",
82 ~     "type": "selection_collapsed",
83 ~     "label": "What to eat",
84 ~     "comments": "Homemade with much love",
85 ~     "values": {
86 ~       "Soup": "soup",
87 ~       "Dumplings": "dumplings"
88 ~     },
89 ~     "default_value": "dumplings"
90 ~   },
91 ~   {
92 ~     "id": "drink",
93 ~     "type": "selection_radio",
94 ~     "label": "What to drink",
95 ~     "comments": "but IPOL won't make the tea for you :((",
96 ~     "values": {
97 ~       "Oolong": "oolong",
98 ~       "Green": "green",
99 ~       "Black": "black"
100 ~    },
101 ~     "default_value": "oolong"
102 ~   },
103 ~   {
104 ~     "id": "size",
105 ~     "type": "checkbox",
106 ~     "label": "Large portions?",
107 ~     "comment": "of course !",
108 ~     "default_value": "False"
109 ~   }
110 ~ ],

```



# Run

```
17         "type": "image"
18     }
19 ],
20 "params": [
21 ],
22 "run": "python $bin/code/comprint.py -i input_0.png -o ./output",
23 "results": [
24     {
25         "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
<input type="checkbox"/> 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

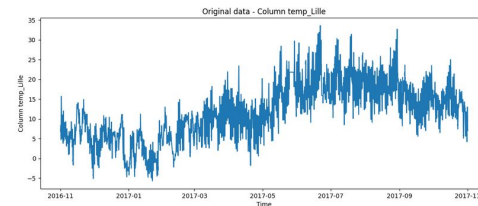
	Rule	Support	Confidence	Lift
0	mean_national_temp_low -> temp_Aix_low	0.4	1.0	0.5
1	temp_Aix_low -> mean_national_temp_low	0.4	1.0	0.5
2	mean_national_temp_low -> temp_Lille_low	0.4	1.0	0.5
3	temp_Lille_low -> mean_national_temp_low	0.4	1.0	0.5
4	temp_Aix_low -> temp_Lille_low	0.4	1.0	0.5
5	temp_Lille_low -> temp_Aix_low	0.4	1.0	0.5

Symbol Basket

	Support for low-valued deviant event	Support for high-valued deviant event	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
mean_Lille	0.10	NaN	NaN

Original data

Original data



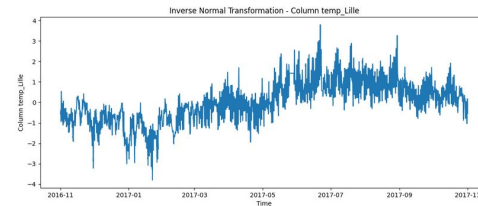
Zoom 1x

Inverse Normal Transformation

INT

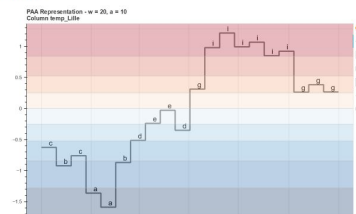
qq-plot

☐ 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"
    }
  },

```

## Experiments on Deep Single-Image Portrait Relighting

[Article](#) [Demo](#) [Archive](#)

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.

First Previous 1 2 3 4

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



Input image



Output



Estimated lighting



Target lighting

Reconstruct

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
k <sub>yx</sub>	0.15
k <sub>yz</sub>	0
k <sub>zz</sub>	0
k <sub>xz</sub>	0
k <sub>xyy</sub>	0



Input image



### Output



Estimated lighting



Target lighting

Reconstruct

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



Input image



Output



Estimated lighting

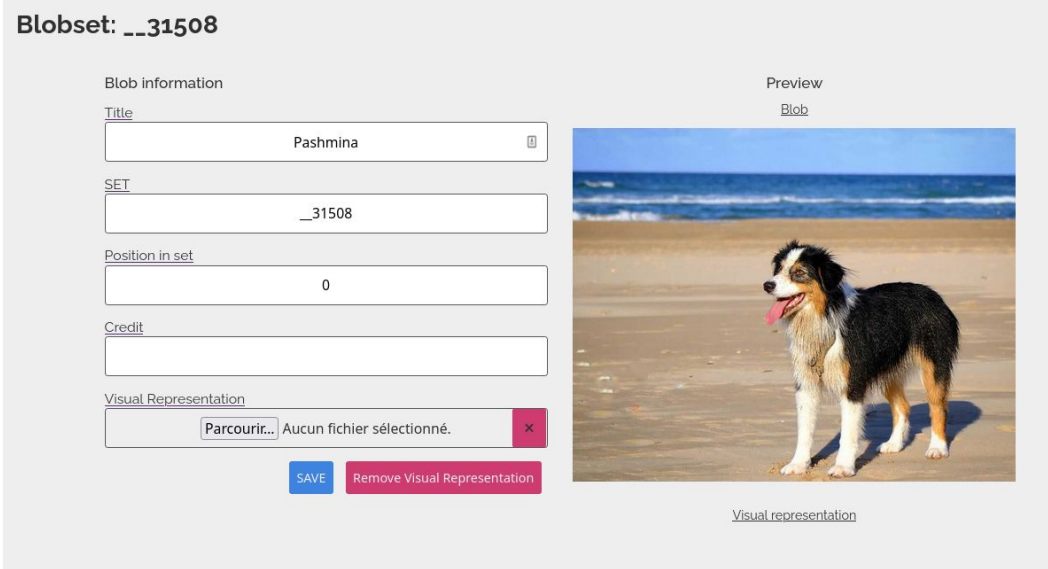
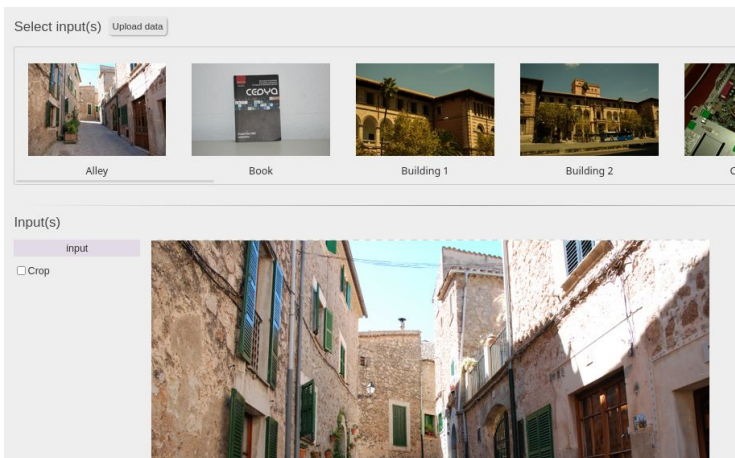
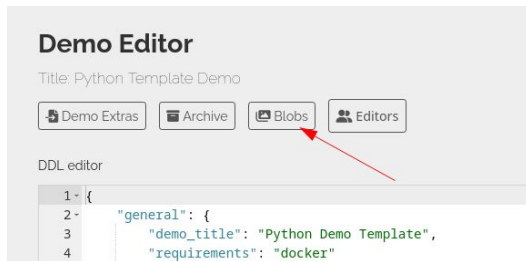


Target lighting

# 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

## Manage Blobs for demo

Demo blobsets

Add new Blob

Templates associated

Add template to demo



VariousForgeries



# Templates

[Demos](#)[Templates](#)[Status](#)[Welcome quentin\\_bammey](#)[Logout](#)

## List of Templates

[Create template](#)

denoisingTemplate	MicrotexturesTemplate	CartoonTextureTemplate	StandardTestImages	JPEGQuality	Forgery
Deblurring	retinexTemplate	contrastEnhancementTe..	PalettesTextureGenerator	BracketedExposureSeque..	FusionEvaluation
Statokinesigrams	KodakImageSuite	McMasterDataset	stereo_template	Statokinesigram	contrastEnhancement
FacedetectionTemplate	TextureTemplate	CloudDetection_Sentinelz..	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

The screenshot shows a grid of 48 template buttons, each with a stack icon and a name. The buttons are arranged in 8 rows and 6 columns. A 'Create template' button is located in the top right corner. A modal window titled 'Create template' is open in the center, featuring a text input field labeled 'Template name' and a '+ Create' button. An orange arrow points from the 'Create template' button in the top right to the modal. Another orange arrow points from the 'No spaces here!' text box to the 'Template name' input field.

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	Ch...	...	#CroisonsLes from @Guill...	WindTurbineDetecNoGdaL...
NoiseFreeTestImagesGray...	Line segment detection	M...	...	video	images
SmartCoast	VariousForgeries	Po...	...		

**No spaces here!**

# Templates

Template: denoisingTemplate

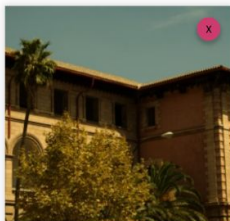
Template blobsets



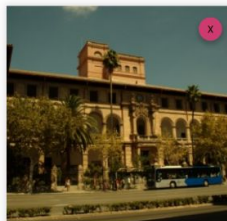
alley



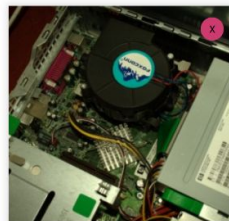
book



building1



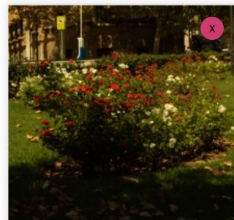
building2



computer



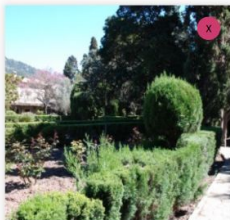
dice



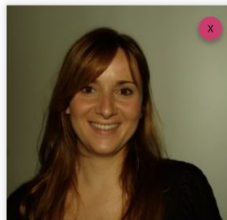
flowers1



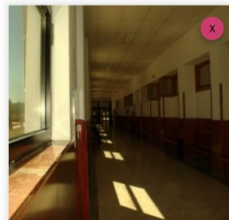
flowers2



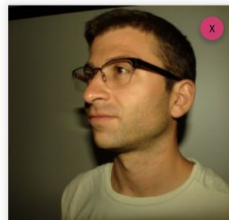
gardens



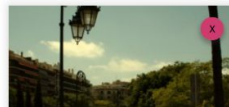
girl



hallway



man1



Add new Blob

Delete Template