Universitat Oberta de Catalunya

eHealth Center

uoc.edu

DEMO of patch **extraction** from images **& reconstruction** of patches into images

Àlex Bedmar Martínez

PhD Supervisors: Dr. Ferran Prados

NeuroADaS Lab Scientific Sessions - February 25th 2025

Workflow

NeuroADaS Lab

Extract patches from images

Upload them to the website

Categorize the images

Reconstruct the images from the categorized images

Upload docker image

NeuroADaS Lab

```
PS C:\...\docker_DEMO> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\...\docker_DEMO> docker tag patchextractor_demo-python-app bedmar001/patchextractor_demo-python-app
PS C:\...\docker_DEMO> docker push bedmar001/patchextractor_demo-python-app
Using default tag: latest
Using default tag: latest
7576b00d9bb1: Pushed
fd47d20a593b: Pushed
861d38f94d80: Pushed
861d38f94d80: Pushing [============================>                      ]  8.389MB/15.17MB
e844e4b3e270: Pushed
ff1cff00b1dd: Pushed
3b587922ad1a: Pushed
32b550be6cb6: Pushed
32b550be6cb6: Pushing [====================>                              ]  10.49MB/24.06MB
57e75b8db17c: Pushed
a492eee5e559: Pushed
a492eee5e559: Pushing [========>                                          ]  8.389MB/48.48MB
2fef56314ef4: Pushed
35af2a7690f2: Pushed
latest: digest: sha256:03691bc5eb00e8d1434f66e9a7c151a396f659fdce80353e80627bf23d811d8e size: 856
```

Requirements

NeuroADaS Lab

Requirements:

Dataset with images to extract patches from

Download docker image

NeuroADaS Lab

```
PS C:\...\yourdir> docker login
Authenticating with existing credentials...
Login Succeeded

PS C:\...\yourdir> docker pull bedmar001/patchextractor_demo-python-app
Using default tag: latest
latest: Pulling from bedmar001/patchextractor_demo-python-app
8586e048f063: Download complete
57e75b8db17c: Download complete
861d38f94d80: Download complete
e844e4b3e270: Download complete
a492eee5e559: Download complete
7576b00d9bb1: Download complete
3b587922ad1a: Download complete
2fef56314ef4: Download complete
32b550be6cb6: Download complete
533e5f89b4e1: Download complete
35af2a7690f2: Download complete
fd47d20a593b: Download complete
ff1cff00b1dd: Download complete
Digest: sha256:03691bc5eb00e8d1434f66e9a7c151a396f659fdce80353e80627bf23d811d8e
Status: Downloaded newer image for bedmar001/patchextractor_demo-python-app:latest
docker.io/bedmar001/patchextractor_demo-python-app:latest

PS C:\...\yourdir> docker run --rm -v ${PWD}/data:/usr/src/app/data -it bedmar001/patchextractor_demo-python-app
```

Commands

NeuroADaS Lab

```
PS C:\...\yourdir> docker run -it bedmar001/patchextractor_demo-python-app python
Python 3.10.16 (main, Feb  4 2025, 07:28:23) [GCC 12.2.0] on linux

>>> from patch_extractor_DEMO import patch_extractor

>>> from patch_extractor_DEMO import patch_extractor_4x4

>>> from patch_extractor_DEMO import patch_extractor_2d

>>> from patch_extractor_DEMO import patch_extractor_4x4_2d
```

	custom patches	4x4 patches
3D	patch_extractor	patch_extractor_4x4
2D	patch_extractor_2d	patch_extractor_4x4_2d

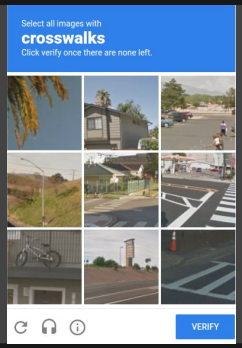
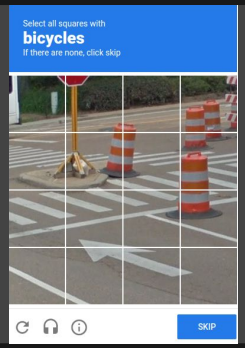
Commands

NeuroADaS Lab

	custom patches	4x4 patches
3D	patch_extractor	patch_extractor_4x4
2D	patch_extractor_2d	patch_extractor_4x4_2d

Commands

NeuroADaS Lab

	custom patches	4x4 patches
3D		
2D		

Parameters

NeuroADaS Lab

3D patches

```
patch_extractor(
    data_file="data/data_238.nii.gz",
    label_file="data/label_238.nii.gz",
    output_dir="output_patches_xyz",
    num_patches=2,
    patch_size_range=(100, 300),
    label_percentage_range=(5, 20),
    planes=('sagittal','coronal','axial',),
    layer_range={'sagittal': (50, 600),'coronal': (50, 400),'axial': (50, 400)},
    labels_present=[1, 4, (8, 15)]
)
```

```
patch_extractor_4x4(
    data_file = "data/data_238.nii.gz",
    label_file = "data/label_238.nii.gz",
    output_dir = "output_patches_4x4_xyz",
    num_patches = 5,
    patch_size_range = (400, 400),
    label_percentage_range = (7, 22),
    planes = ('sagittal', 'coronal', 'axial'),
    layer_range = {'sagittal': (10, 400), 'coronal': (10, 400), 'axial': (10, 400)},
    labels_present = (1, 3, 6, 8)
)
```

`data_file`: Path to the `.nii.gz` file containing the medical imaging data.

`label_file`: Path to the corresponding label `.nii.gz` file.

`output_dir`: Directory where extracted patches and metadata will be stored.

`num_patches`: Number of patches to extract.

`patch_size_range`: Tuple specifying the minimum and maximum patch size.

`label_percentage_range`: Range of valid label percentages to filter patches.

`planes`: Tuple specifying which planes (x='sagittal', y='coronal', z='axial') to extract patches from.

`layer_range`: Dictionary specifying layer ranges for each plane.

`labels_present`: List of required labels in the patch (supports ranges using tuples).

`generate_all_planes`: If `True`, extracts patches from all planes regardless of `plane` parameter.

Commands

NeuroADaS Lab

3D custom patches

```
PS C:\...\yourdir> docker run --rm -v ${PWD}/data:/usr/src/app/data -it bedmar001/patchextractor_demo-python-app
PS C:\...\yourdir> python3
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> from patch_extractor_DEMO import patch_extractor
>>>
>>> patch_extractor(
...     data_file="data/mouth/data_491.nii.gz",
...     label_file="data/mouth/label_491.nii.gz",
...     output_dir="output_patches_xyz",
...     num_patches=2,
...     patch_size_range=(100, 300),
...     label_percentage_range=(5, 20),
...     planes=('sagittal','coronal','axial',),
...     layer_range={'sagittal': (50, 600),'coronal': (50, 400),'axial': (50, 400)},
...     labels_present=[1, 2, (8, 15)]
... )

[X] Patch 1 skipped: Label percentage 0.00% or labels not in range.
[X] Saved patch 1/2: Label percentage 16.47%
[X] Patch 3 skipped: Label percentage 31.04% or labels not in range.
[X] Saved patch 2/2: Label percentage 9.27%

[Y] Patch 1 skipped: Label percentage 0.00% or labels not in range.
[Y] Patch 2 skipped: Label percentage 26.52% or labels not in range.
...
```

Output

NeuroADaS Lab

3D custom patches

Commands

NeuroADaS Lab

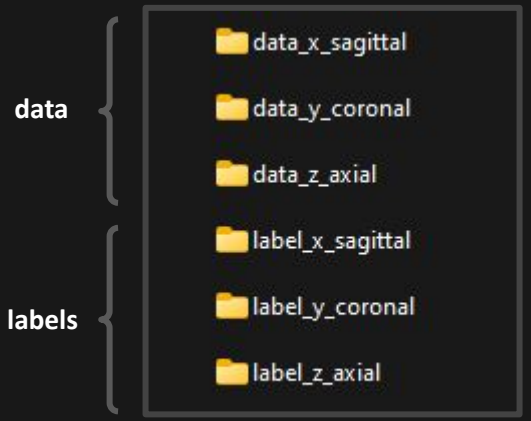
3D 4x4 patches

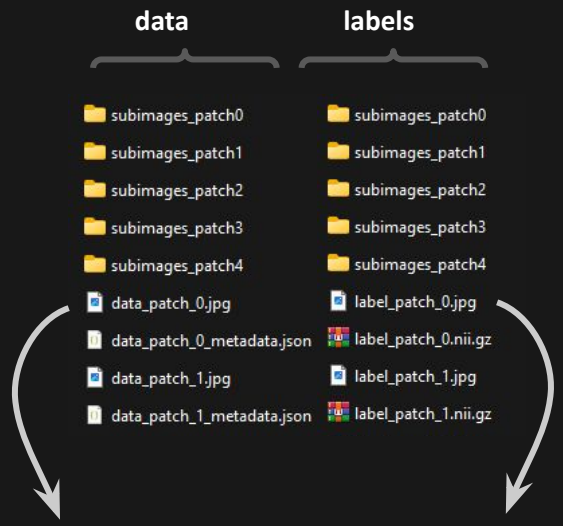
```
PS C:\...\yourdir> docker run --rm -v ${PWD}/data:/usr/src/app/data -it bedmar001/patchextractor_demo-python-app
PS C:\...\yourdir> python3
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> from patch_extractor_DEMO import patch_extractor
>>>
>>> patch_extractor_4x4(
...     data_file = "data/brain/flair_bfc.nii.gz",
...     label_file = "data/TIMILS/subj1/flair_bfc_filled_NeuroMorph_Parcellation.nii.gz",
...     output_dir = "output_patches_4x4_xyz",
...     num_patches = 5,
...     patch_size_range = (100, 150),
...     label_percentage_range = (7, 22),
...     planes = ('sagittal', 'coronal', 'axial'),
...     layer_range = {'sagittal': (10, 100), 'coronal': (10, 100), 'axial': (10, 100)},
...     labels_present = (1, 2, 3, 4)
... )

...

[X] Patch 1 skipped: Labels [0] not in range.
[X] Patch 2 skipped: Labels [0] not in range.
[X] Saved patch 1/5: Labels [0, 1, 2, 3, 4, 39, 40, 41, 42, 67, 72, 73, 74, 81, 85, 86, 89, 93, 94, 107, 108, 109, 110, 115, 116, 123, 124,
129, 130, 133, 134, 135, 136, 155, 156, 161, 168, 169, 170, 177, 178, 195, 196, 199, 200, 202]
[X] Patch 4 skipped: Labels [0] not in range.
```

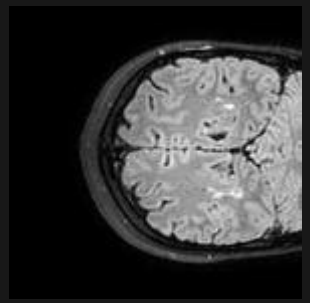
Output

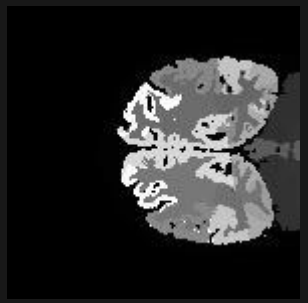
NeuroADaS Lab

3D 4x4 patches





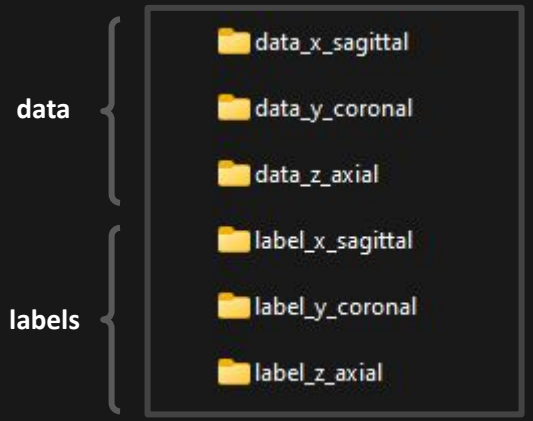


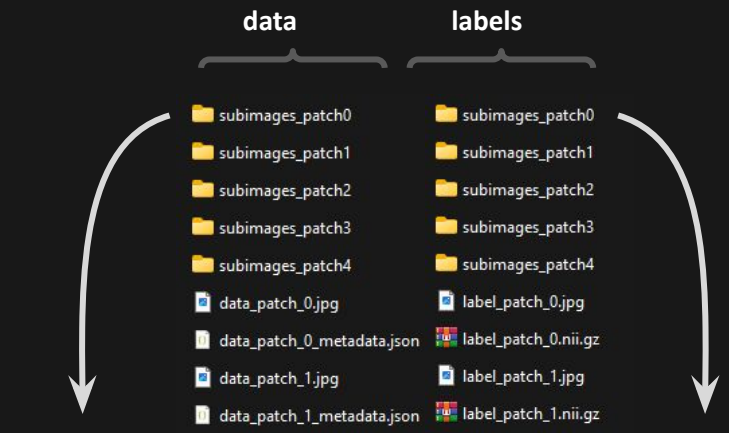


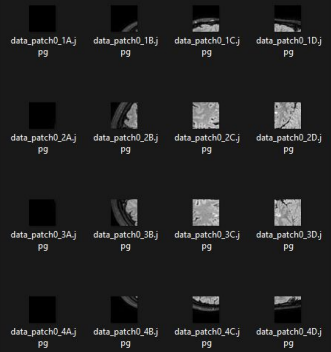
Output

NeuroADaS Lab

3D 4x4 patches



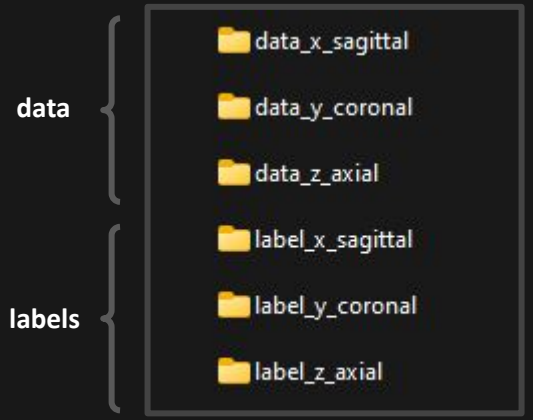


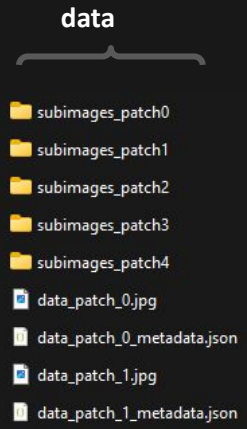


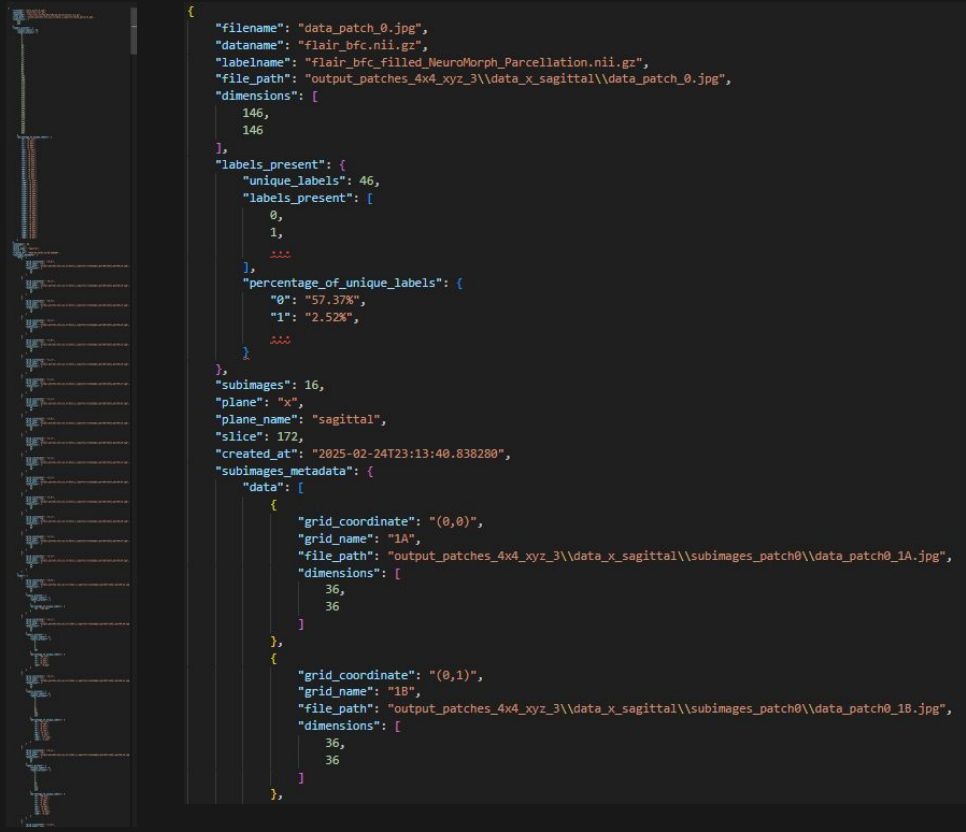
Output

NeuroADaS Lab

3D 4x4 patches








```
{
    "filename": "data_patch_0.jpg",
    "dataname": "flair_bfc.nii.gz",
    "labelname": "flair_bfc_filled_NeuroMorph_Parcellation.nii.gz",
    "file_path": "output_patches_4x4_xyz_3\\data_x_sagittal\\data_patch_0.jpg",
    "dimensions": [
        146,
        146
    ],
    "labels_present": {
        "unique_labels": 46,
        "labels_present": [
            0,
            1,
            ...
        ],
        "percentage_of_unique_labels": {
            "0": "57.37%",
            "1": "2.52%",
            ...
        }
    },
    "subimages": 16,
    "plane": "x",
    "plane_name": "sagittal",
    "slice": 172,
    "created_at": "2025-02-24T23:13:40.838280",
    "subimages_metadata": {
        "data": [
            {
                "grid_coordinate": "(0,0)",
                "grid_name": "1A",
                "file_path": "output_patches_4x4_xyz_3\\data_x_sagittal\\subimages_patch0\\data_patch0_1A.jpg",
                "dimensions": [
                    36,
                    36
                ]
            },
            {
                "grid_coordinate": "(0,1)",
                "grid_name": "1B",
                "file_path": "output_patches_4x4_xyz_3\\data_x_sagittal\\subimages_patch0\\data_patch0_1B.jpg",
                "dimensions": [
                    36,
                    36
                ]
            },
```

Parameters

NeuroADaS Lab

2D patches

```
patch_extractor_2d(
    data_file="datasets/lung/img/MCUCXR_0002_0.png",
    label_file="datasets/lung/mask/MCUCXR_0002_0.png",
    output_dir="output_patches_2d_TEST_2",
    num_patches=10,
    patch_size_range=(90, 100)
)
```

```
patch_extractor_4x4_2d(
    data_file = "datasets/lung/img/MCUCXR_0002_0.png",
    label_file = "datasets/lung/mask/MCUCXR_0002_0.png",
    output_dir= "output_patches_4x4_2d_TEST4",
    num_patches = 5,
    patch_size_range = (450, 500)
)
```

data_file: Path to the `.nii.gz` file containing the medical imaging data.

label_file: Path to the corresponding label `.nii.gz` file.

output_dir: Directory where extracted patches and metadata will be stored.

num_patches: Number of patches to extract.

patch_size_range: Tuple specifying the minimum and maximum patch size.

label_percentage_range: Range of valid label percentages to filter patches.

labels_present: List of required labels in the patch (supports ranges using tuples).

Commands

NeuroADaS Lab

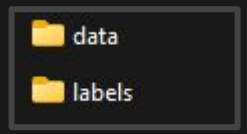
2D custom patches

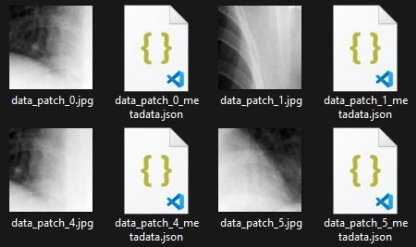
```
PS C:\...\yourdir> docker run --rm -v ${PWD}/data:/usr/src/app/data -it bedmar001/patchextractor_demo-python-app
PS C:\...\yourdir> python3
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> from patch_extractor_DEMO import patch_extractor
>>>
>>> extract_patches_2d(
...     data_file="data/lung/img/MCUCXR_0001_0.png",
...     label_file="data/lung/mask/MCUCXR_0001_0.png",
...     output_dir="output_patches_2d",
...     num_patches=10,
...     patch_size_range=(90, 100),
...     label_percentage_range=(10, 40),
...     labels_present=[255]
... )

...

Patch 1 skipped: Label percentage 56.21% out of range.
Patch 1 skipped: Label percentage 41.59% out of range.
Patch 1 skipped: Label percentage 0.00% out of range.
Saved patch 1/10: Label percentage 38.43%
Patch 2 skipped: Label percentage 0.00% out of range.
Saved patch 2/10: Label percentage 35.64%
```

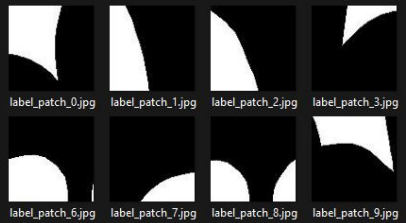
Output

NeuroADaS Lab

2D custom patches







metadata:


```
{
    "filename": "data_patch_0.jpg",
    "dataname": "MCUCXR_0001_0.png",
    "labelname": "MCUCXR_0001_0.png",
    "file_path": "output_patches_2d_TEST\\data\\data_patch_0.jpg",
    "dimensions": [
        91,
        91
    ],
    "label_present": 255,
    "label_percentage": "38.43%",
    "created_at": "2025-02-13T13:48:55.656295"
}
```

Commands

NeuroADaS Lab

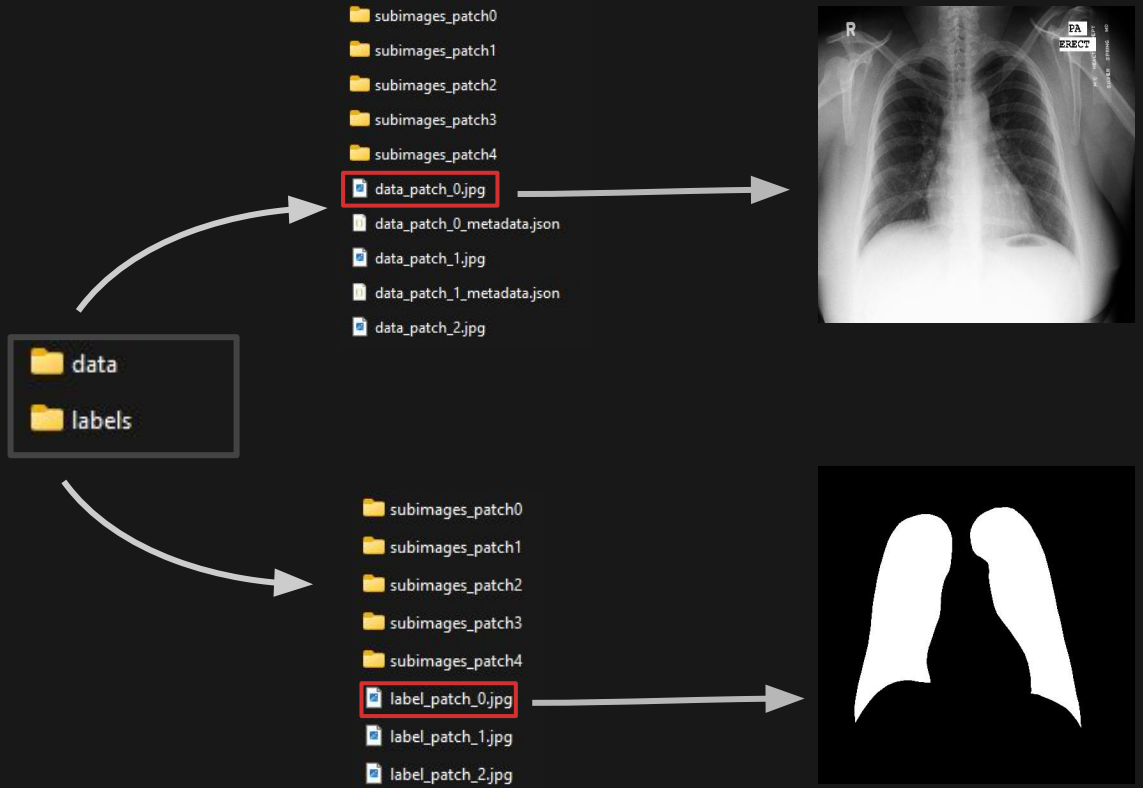
2D 4x4 patches

```
PS C:\...\yourdir> docker run --rm -v ${PWD}/data:/usr/src/app/data -it bedmar001/patchextractor_demo-python-app
PS C:\...\yourdir> python3
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> from patch_extractor_DEMO import patch_extractor
>>>
>>> patch_extractor_4x4_2d(
...     data_file = "datasets/Montgomery/img/MCUCXR_0002_0.png",
...     label_file = "datasets/Montgomery/mask/MCUCXR_0002_0.png",
...     output_dir= "output_patches_4x4_2d_TEST4",
...     num_patches = 5,
...     patch_size_range = (450, 500)
... )
```

Output

NeuroADaS Lab

2D 4x4 patches



Output

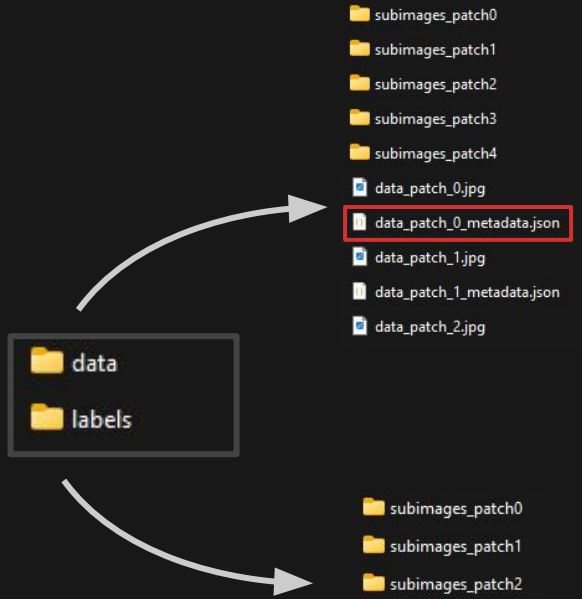
NeuroADaS Lab

2D 4x4 patches

Output

NeuroADaS Lab

2D 4x4 patches




```
{
    "filename": "data_patch_0.jpg",
    "dataname": "MCUCXR_0002_0.png",
    "labelname": "MCUCXR_0002_0.png",
    "patch_index": 0,
    "patch_size": 496,
    "position": {
        "x": 1,
        "y": 16
    },
    "labels_present": {
        "unique_labels": 2,
        "labels_present": [
            0,
            255
        ],
        "percentage_of_unique_labels": {
            "0": "78.35%",
            "255": "21.65%"
        }
    },
    "subimages_metadata": {
        "data": [
            {
                "grid_coordinate": "(0,0)",
                "grid_name": "1A",
                "file_path": "output_patches_4x4_2d\\data\\subimages_patch0\\data_patch0_1A.jpg",
                "dimensions": [
                    124,
                    124
                ]
            },
            {
                "grid_coordinate": "(0,1)",
                "grid_name": "1B",
                "file_path": "output_patches_4x4_2d\\data\\subimages_patch0\\data_patch0_1B.jpg",
                "dimensions": [
                    124,
                    124
                ]
            },
        "label": [
            {
                "grid_coordinate": "(0,0)",
                "grid_name": "1A",
                "file_path": "output_patches_4x4_2d\\labels\\subimages_patch0\\label_patch0_1A.jpg",
                "dimensions": [
                    124,
                    124
                ],
                "labels_present": {
                    "unique_labels": 2,
                    "labels_present": [
                        0,
                        255
                    ],
                    "percentage_of_unique_labels": {
                        "0": "98.09%",
                        "255": "1.91%"
                    }
                }
            },
```

Commands

NeuroADaS Lab

3D custom patches (bash script)

```bash
#!/bin/bash
INPUT_DIR="data"
OUTPUT_DIR="output_patches_xyz"

# generar una lista aleatoria de etiquetas
generate_labels_present() {
    local labels=($(seq 1 28)) # array de etiquetas validas
    local selected=()
    local num_elements=$((RANDOM % 7 + 1)) # numero random de elementos
(entre 1 y 7)

    for ((i=0; i<num_elements; i++)); do
        if (( RANDOM % 2 )); then
            # Seleccionar una única etiqueta
            selected+=("${labels[$((RANDOM % ${#labels[@]}))]}")
        else
            # Seleccionar un rango
            local start_idx=$((RANDOM % (${#labels[@]} - 1)))
            local end_idx=$((start_idx + RANDOM % (28 - start_idx) + 1))
            selected+=("(${labels[start_idx]}, ${labels[end_idx]})")
        fi
    done

    echo "[${selected[*]}]"
}
```

```bash
# buscar archivos de datos y etiquetas
for data_file in "$INPUT_DIR"/data_*.nii.gz; do
    label_file="${data_file/data_/label_}" # Reemplaza "data_" por "label_" para
obtener la etiqueta correspondiente

    if [[ -f "$label_file" ]]; then
        labels_present=$(generate_labels_present)

        python3 -c "
import patch_extractor

patch_extractor(
    data_file='$data_file',
    label_file='$label_file',
    output_dir='$OUTPUT_DIR',
    num_patches=2,
    patch_size_range=(100, 300),
    label_percentage_range=(5, 20),
    planes=('sagittal','coronal','axial'),
    layer_range={'sagittal': (50, 600), 'coronal': (50, 400), 'axial': (50, 400)},
    labels_present=$labels_present
)
"
    else
        echo "No se encontró etiqueta correspondiente para: $data_file"
    fi
done
```

Commands

NeuroADaS Lab

reconstruct_volume

```
PS C:\Users\alexb\OneDrive\Escritorio\NeuroADaS\docker_DEMO> docker run patchextractor_demo-python-app
PS C:\Users\alexb\OneDrive\Escritorio\NeuroADaS\docker_DEMO> python3
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from patches_to_nifti import reconstruct_volume
>>> reconstruct_volume("output_patches_xyz", "reconstructed_volumes_xyz")
Saved reconstructed patch volume: reconstructed_volumes_xyz\data_491_patch_277.nii.gz
Saved reconstructed patch volume: reconstructed_volumes_xyz\data_491_patch_440.nii.gz
Saved reconstructed patch volume: reconstructed_volumes_xyz\data_491_patch_117.nii.gz
Saved reconstructed patch volume: reconstructed_volumes_xyz\data_491_patch_306.nii.gz
Saved reconstructed patch volume: reconstructed_volumes_xyz\data_491_patch_260.nii.gz
Saved reconstructed patch volume: reconstructed_volumes_xyz\data_491_patch_214.nii.gz
```

Output:
(reconstructed_volumes_xyz)