



MONAI : Librería de Procesamiento de Imagen Médica

Autor: [Emilio Delgado Muñoz](#)

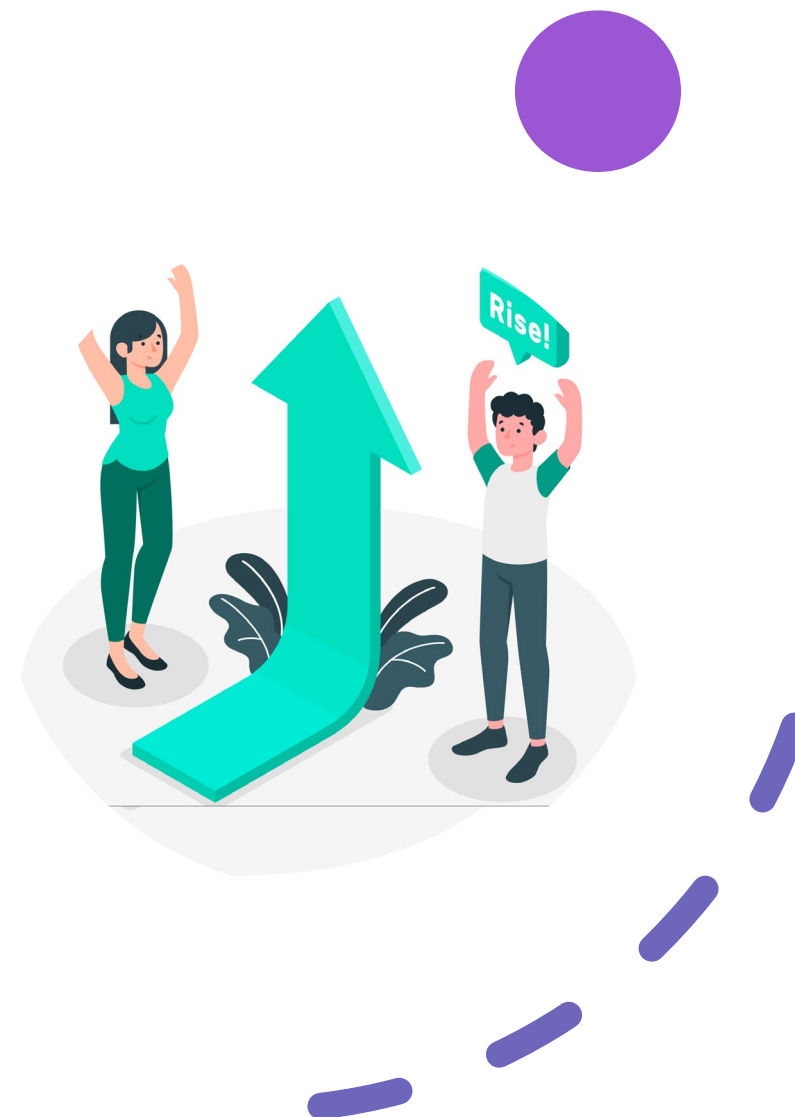
¿Qué es MONAI?

- *Open-source framework*
- *Librería Python*
- *Software de Alta Calidad*
- *Repositorio de Modelos Preentrenados*
- *Software Development Kit (SDK)*

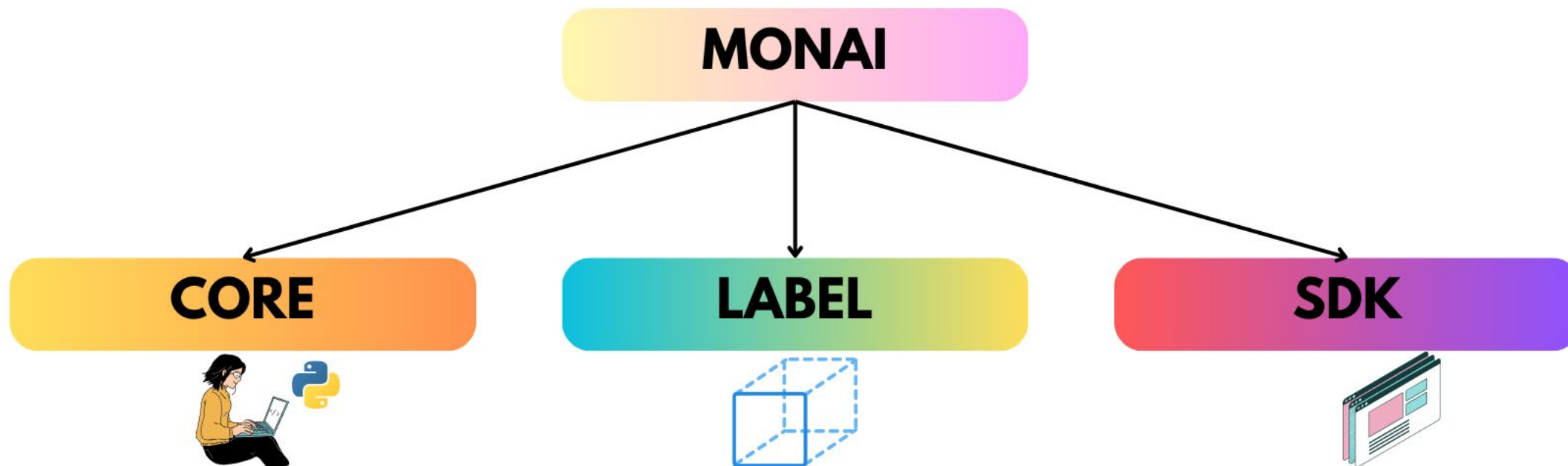


¿Cuáles son los beneficios de usar MONAI?

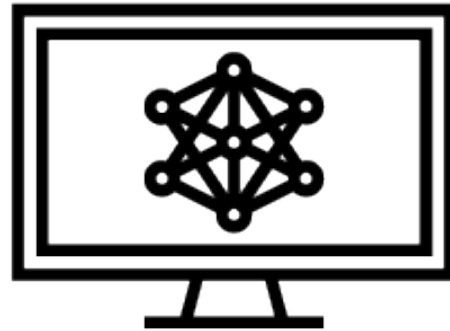
- **Diseño Open Source** (Cualquiera puede aportar)
- **Estandarizado**
- **User Friendly** (API Sencilla)
- **Integración Sencilla con otros frameworks**
- **Reproducible**



¿Qué puede ofrecernos MONAI?



Monai Core



Monai Core

Ventajas

- Carga y guardado de datos fácilmente para muchos estándares de imágenes médicas [NIFTI; DICOM; Numpy] (¡Solo preocúpate por las rutas de los archivos!)
- Gestión eficiente de la memoria (gestionada directamente por el framework)
- Muchos componentes específicos para el procesamiento de imágenes médicas
- Potente para crear pipelines de Deep Learning en imágenes médicas (funciones de pérdida, métricas y más, específicas del dominio)
- Contiene muchos conjuntos de datos de imágenes médicas (por ejemplo, MedNist, Decathlon y más)

Desventajas

1. API de alto nivel

Section Navigation

[Applications](#)[Auto3dseg](#)[Federated Learning](#)[Model Bundle](#)[Transforms](#)[Loss functions](#)[Network architectures](#)[Metrics](#)[Optimizers](#)[Data](#)[Engines](#)[Inference methods](#)[Event handlers](#)[Visualizations](#)[Utilities](#)[LossMetric](#)[Mean Dice](#)[Mean IoU](#)[Generalized Dice Score](#)[Area under the ROC curve](#)[Confusion matrix](#)[Hausdorff distance](#)[Average surface distance](#)[Surface dice](#)[PanopticQualityMetric](#)[Mean squared error](#)[Mean absolute error](#)[Root mean squared error](#)[Peak signal to noise ratio](#)[Structural similarity index measure](#)[Multi-scale structural similarity index measure](#)[Fréchet Inception Distance](#)[Maximum Mean Discrepancy](#)[Cumulative average](#)

Monai Core

Data

Data

Generic Interfaces

Dataset

`class monai.data.Dataset(data, transform=None) # [source]`

A generic dataset with a length property and an optional callable data transform when fetching a data sample. If passing slicing indices, will return a PyTorch Subset, for example: `data: Subset = dataset[1:4]`, for more details, please check: <https://pytorch.org/docs/stable/data.html#torch.utils.data.Subset>

For example, typical input data can be a list of dictionaries:

Dataset

- Dataset
- IterableDataset
- DatasetFunc
- ShuffleBuffer
- CSVIterableDataset
- PersistentDataset
- GDSDataset
- CacheNTransDataset
- LMDBDataset
- CacheDataset
- SmartCacheDataset
- ZipDataset
- ArrayDataset

Metrics

Surface dice

`monai.metrics.compute_surface_dice(y_pred, y, class_thresholds, include_background=False, distance_metric='euclidean', spacing=None, use_subvoxels=False) [source]`

This function computes the (Normalized) Surface Dice (NSD) between the two tensors `y_pred` (referred to as \hat{Y}) and `y` (referred to as Y). This metric determines which fraction of a segmentation boundary is correctly predicted. A boundary element is considered correctly predicted if the closest distance to the reference boundary is smaller than or equal to the specified threshold related to the acceptable amount of deviation in pixels. The NSD is bounded between 0 and 1.

This implementation supports multi-class tasks with an individual threshold τ_c for each class c . The class-specific NSD for batch index b , $NSD_{b,c}$, is computed using the function:

$$NSD_{b,c}(Y_{b,c}, \hat{Y}_{b,c}) = \frac{|D'_{Y_{b,c}}| + |D'_{\hat{Y}_{b,c}}|}{|D_{Y_{b,c}}| + |D_{\hat{Y}_{b,c}}|} \quad (1)$$

On this page

- FROC
- Metric
- Variance
- LabelQualityScore
- IterationMetric
- Cumulative
- CumulativeIterationMetric
- LossMetric
- Mean Dice
- Mean IoU
- Generalized Dice Score
- Area under the ROC curve
- Confusion matrix
- Hausdorff distance
- Average surface distance

Transforms

Vanilla Transforms

Crop and Pad

PadListDataCollate

`class monai.transforms.PadListDataCollate(method=Method.SYMMETRIC, mode=PytorchPadMode.CONSTANT, **kwargs) [source]`

Same as MONAI's `list_data_collate`, except any tensors are centrally padded to match the shape of the biggest tensor in each dimension. This transform is useful if some of the applied transforms generate batch data of different sizes.

This can be used on both list and dictionary data. Note that in the case of the dictionary data, it may add the transform information to the list of invertible transforms if input batch have different spatial shape, so need to call static method: `inverse` before inverting other transforms.

Note that normally, a user won't explicitly use the `__call__` method. Rather this would be passed to the `DataLoader`. This means that `__call__` handles data as it comes out of a `DataLoader`, containing batch dimension. However, the `inverse` operates on dictionaries containing images of

On this page

- Generic Interfaces
- Functionals
- Vanilla Transforms
- Crop and Pad
- PadListDataCollate
- Pad
- SpatialPad
- BorderPad
- DivisiblePad
- Crop
- SpatialCrop
- CenterSpatialCrop
- RandSpatialCrop
- RandSpatialCropSamples
- CropForeground
- RandWeightedCrop
- RandCropByPosNegLabel

Loss Functions

GeneralizedDiceLoss

`class monai.losses.GeneralizedDiceLoss(include_background=True, to_onehot_y=False, sigmoid=False, softmax=False, other_act=None, w_type=Weight.SQUARE, reduction=LossReduction.MEAN, smooth_nr=1e-05, smooth_dr=1e-05, batch=False) [source]`

Compute the generalised Dice loss defined in:

Sudre, C. et. al. (2017) Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. DLMIA 2017.

Adapted from: [NiftyNet](#)

`__init__(include_background=True, to_onehot_y=False, sigmoid=False,`

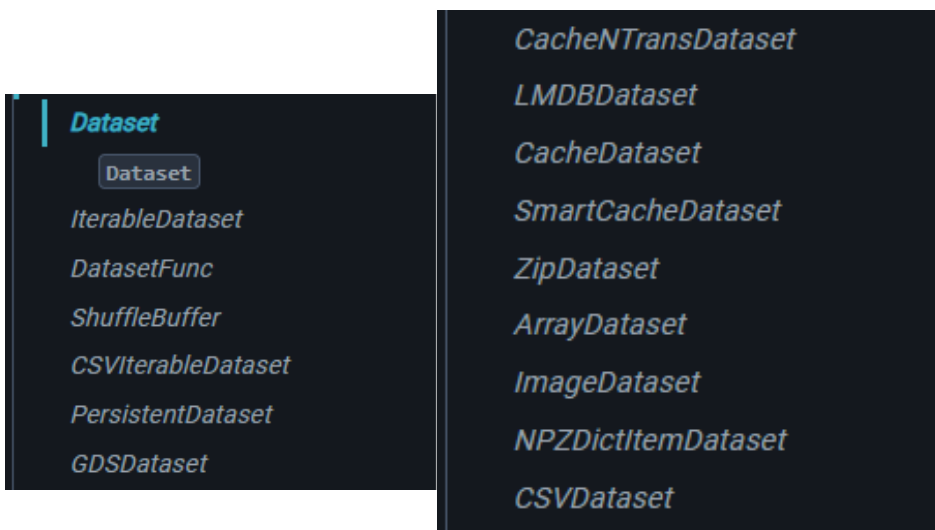
On this page

Segmentation Losses

- DiceLoss
- MaskedDiceLoss
- MaskedDiceLoss
- GeneralizedDiceLoss
- GeneralizedWassersteinDiceLoss
- DiceCELoss
- DiceFocalLoss
- GeneralizedDiceFocalLoss
- FocalLoss
- TverskyLoss
- ContrastiveLoss
- HausdorffDTLoss

Monai Core

Easy Data Management API



MONAI te permite adaptar la gestión de datos de forma sencilla para administrar eficientemente tus recursos:

- **¿Tienes poca RAM?** Usa PersistentDataset o LMDBDataset.
- **¿Tienes más RAM disponible?** Usa CacheDataset o SmartCacheDataset para mejorar la velocidad de las operaciones de entrada/salida durante el entrenamiento.

Monai Core

Arquitecturas codificadas
disponibles

UNETR

```
class monai.networks.nets.UNETR(in_channels, out_channels, img_size,  
feature_size=16, hidden_size=768, mlp_dim=3072, num_heads=12, pos_embed='conv',  
proj_type='conv', norm_name='instance', conv_block=True, res_block=True,  
dropout_rate=0.0, spatial_dims=3, qkv_bias=False, save_attn=False) [source]
```

UNETR based on: "Hatamizadeh et al., UNETR: Transformers for 3D Medical Image Segmentation
<<https://arxiv.org/abs/2103.10504>>"

```
__init__(in_channels, out_channels, img_size, feature_size=16,  
hidden_size=768, mlp_dim=3072, num_heads=12, pos_embed='conv',  
proj_type='conv', norm_name='instance', conv_block=True, res_block=True,  
dropout_rate=0.0, spatial_dims=3, qkv_bias=False, save_attn=False) [source]
```

Parameters:

- **in_channels** – dimension of input channels.
- **out_channels** – dimension of output channels.
- **img_size** – dimension of input image.
- **feature_size** – dimension of network feature size. Defaults to 16.
- **hidden_size** – dimension of hidden layer. Defaults to 768.
- **mlp_dim** – dimension of feedforward layer. Defaults to 3072.
- **num_heads** – number of attention heads. Defaults to 12.
- **proj_type** – patch embedding layer type. Defaults to "conv".

SEResNet50

SEResNet101

SEResNet152

SEResNext50

SEResNext101

HighResNet

DynUNet

UNet

AttentionUnet

UNETR

UNETR

SwinUNETR

BasicUNet

BasicUNetPlusPlus

FlexibleUNet

VNet

RegUNet

GlobalNet

LocalNet

AutoEncoder

VarAutoEncoder

ViT

ViTAutoEnc

Monai Core

Métodos de inferencia

MONAI ofrece múltiples métodos de inferencia basados en los enfoques de **Split & Merge** y **Sliding Window**.

Sliding Window Inference Function

```
monai.inferers.sliding_window_inference(inputs, roi_size, sw_batch_size,
    predictor, overlap=0.25, mode=BlendMode.CONSTANT, sigma_scale=0.125,
    padding_mode=PytorchPadMode.CONSTANT, cval=0.0, sw_device=None, device=None,
    progress=False, roi_weight_map=None, process_fn=None, buffer_steps=None,
    buffer_dim=-1, with_coord=False, *args, **kwargs)
```

[\[source\]](#)

Sliding window inference on *inputs* with *predictor*.

The outputs of *predictor* could be a tensor, a tuple, or a dictionary of tensors. Each output in the tuple or dict value is allowed to have different resolutions with respect to the input. e.g., the input patch spatial size is [128,128,128], the output (a tuple of two patches) patch sizes could be ([128,64,256], [64,32,128]). In this case, the parameter *overlap* and *roi_size* need to be carefully chosen to ensure the output ROI is still an integer. If the predictor's input and output spatial sizes are not equal, we recommend choosing the parameters so that $overlap * roi_size * output_size / input_size$ is an integer (for each spatial dimension).

When *roi_size* is larger than the inputs' spatial size, the input image are padded during inference. To maintain the same spatial sizes, the output image will be cropped to the original input size.

☰ On this page

Inferers

Splitters

Mergers

**Sliding Window Inference
Function**

`sliding_window_inference()`

[Show Source](#)

Monai Core

Runners

MONAI define el concepto de "**Runners**". Un *Runner* es un **pipeline completo** para segmentar conjuntos de datos de imágenes médicas.

nnUNet

```
class monai.apps.nnunet.nnUNetV2Runner(input_config,
trainer_class_name='nnUNetTrainer', work_dir='work_dir',
export_validation_probabilities=True) \[source\]
```

`nnUNetV2Runner` provides an interface in MONAI to use *nnU-Net* V2 library to analyze, train, and evaluate neural networks for medical image segmentation tasks.

`nnUNetV2Runner` can be used in two ways:

1. with one line of code to execute the complete pipeline.
2. with a series of commands to run each modules in the pipeline.

The output of the interface is a directory that contains:

1. converted dataset met the requirement of nnU-Net V2
2. data analysis results
3. checkpoints from the trained U-Net models
4. validation accuracy in each fold of cross-validation
5. the predictions on the testing datasets from the final algorithm ensemble and potential post-processing

☰ On this page

Datasets

Clara MMARs

Utilities

Deepgrow

Pathology

Detection

Reconstruction

FastMRIReader

`FastMRIReader`

ConvertToTensorComplex

ComplexAbs

RootSumOfSquares

ComplexMul

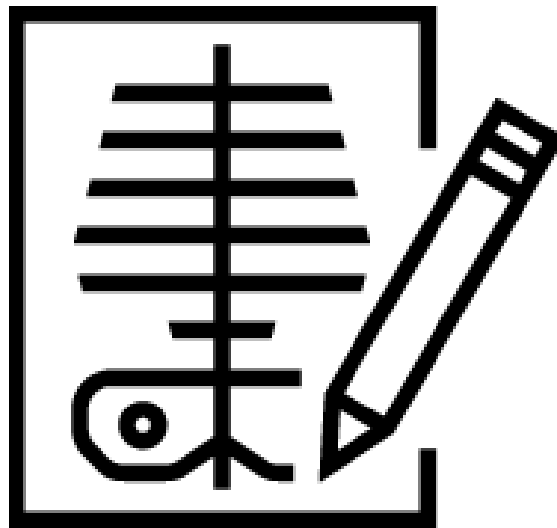
ComplexConj

Auto3DSeg

nnUNet

[Show Source](#)

Monai Label



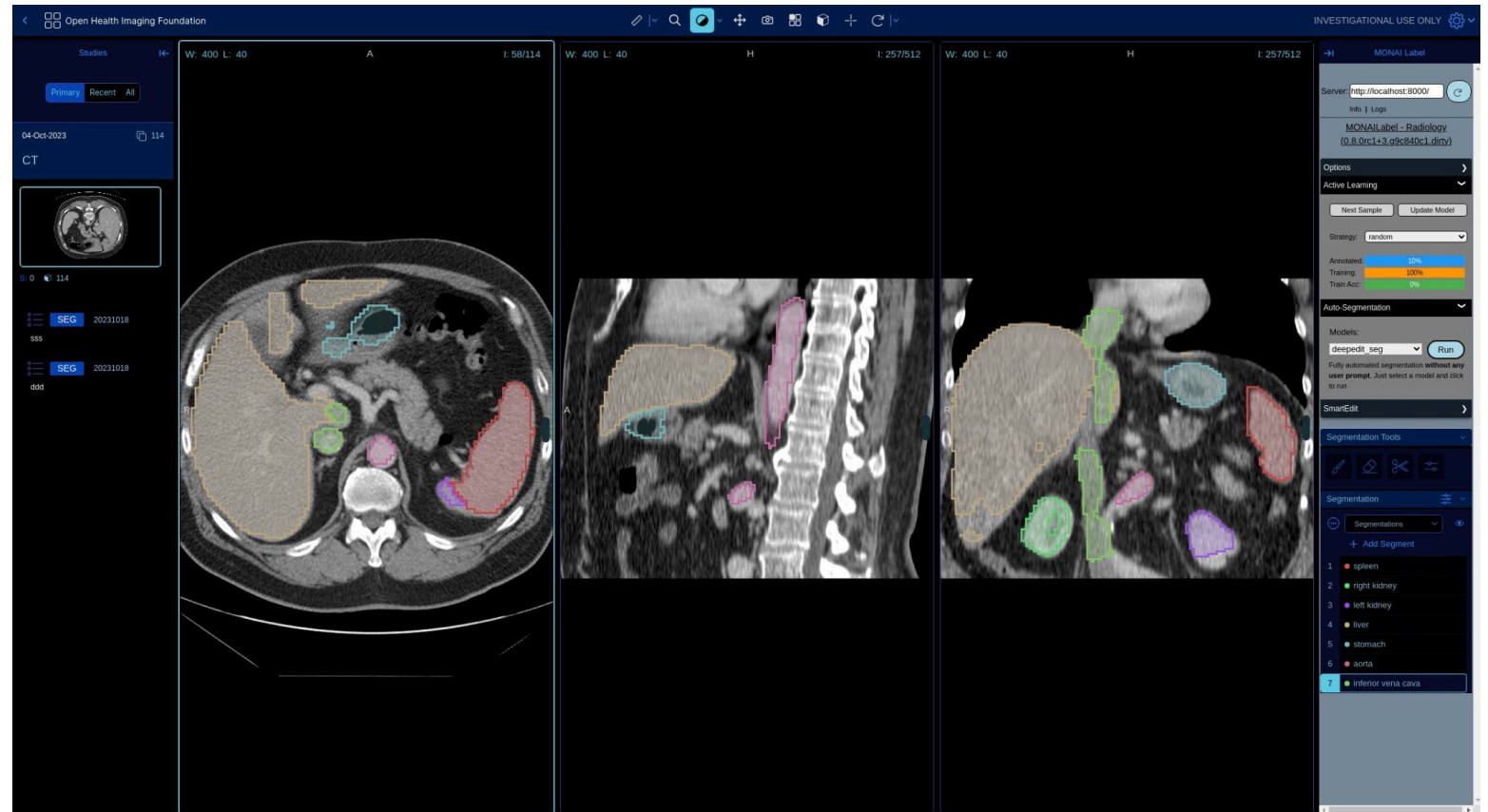
Monai Label

MONAI define **MONAI Label** como:

“Una herramienta inteligente de etiquetado y aprendizaje de imágenes que utiliza asistencia por IA para reducir el tiempo y esfuerzo necesarios para anotar nuevos conjuntos de datos. A través de la interacción con el usuario, MONAI Label entrena o utiliza un modelo de IA para una tarea específica y lo actualiza continuamente a medida que recibe nuevas imágenes anotadas.”

Monai Label

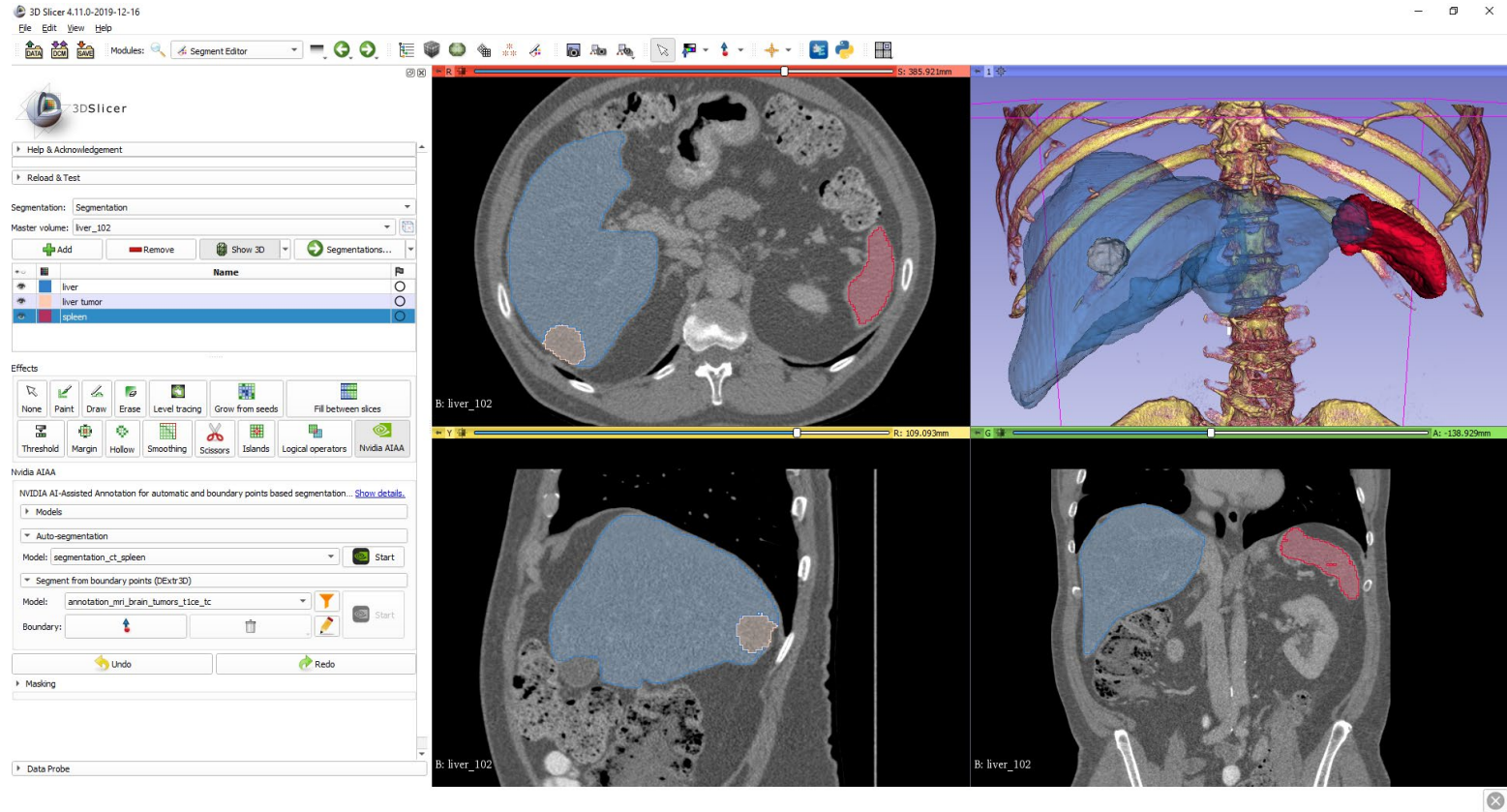
[OHIF Viewer](#) (Monai Open Source Viewer)



Source: <https://github.com/Project-MONAI/MONAILabel/tree/main/plugins/ohifv3>

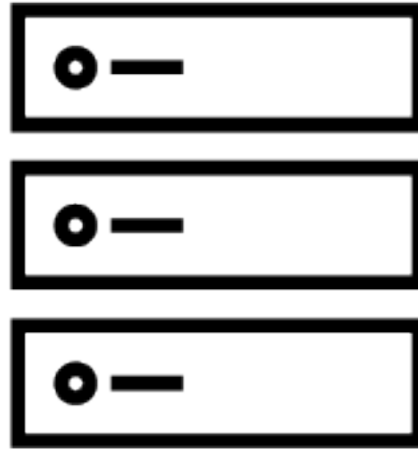
Monai Label

3D Slicer Viewer



Source: <https://www.slicer.org/>

Monai Deploy SDK



Monai Deploy SDK

MONAI define **MONAI SDK** (o **MONAI Deploy**) como:

“MONAI Deploy aspira a convertirse en el estándar de facto para desarrollar, empaquetar, probar, desplegar y ejecutar aplicaciones de IA médica en entornos clínicos. MONAI Deploy establece un conjunto de pasos intermedios donde investigadores y médicos pueden ganar confianza en las técnicas y enfoques utilizados con IA, permitiendo así un flujo de trabajo iterativo.”

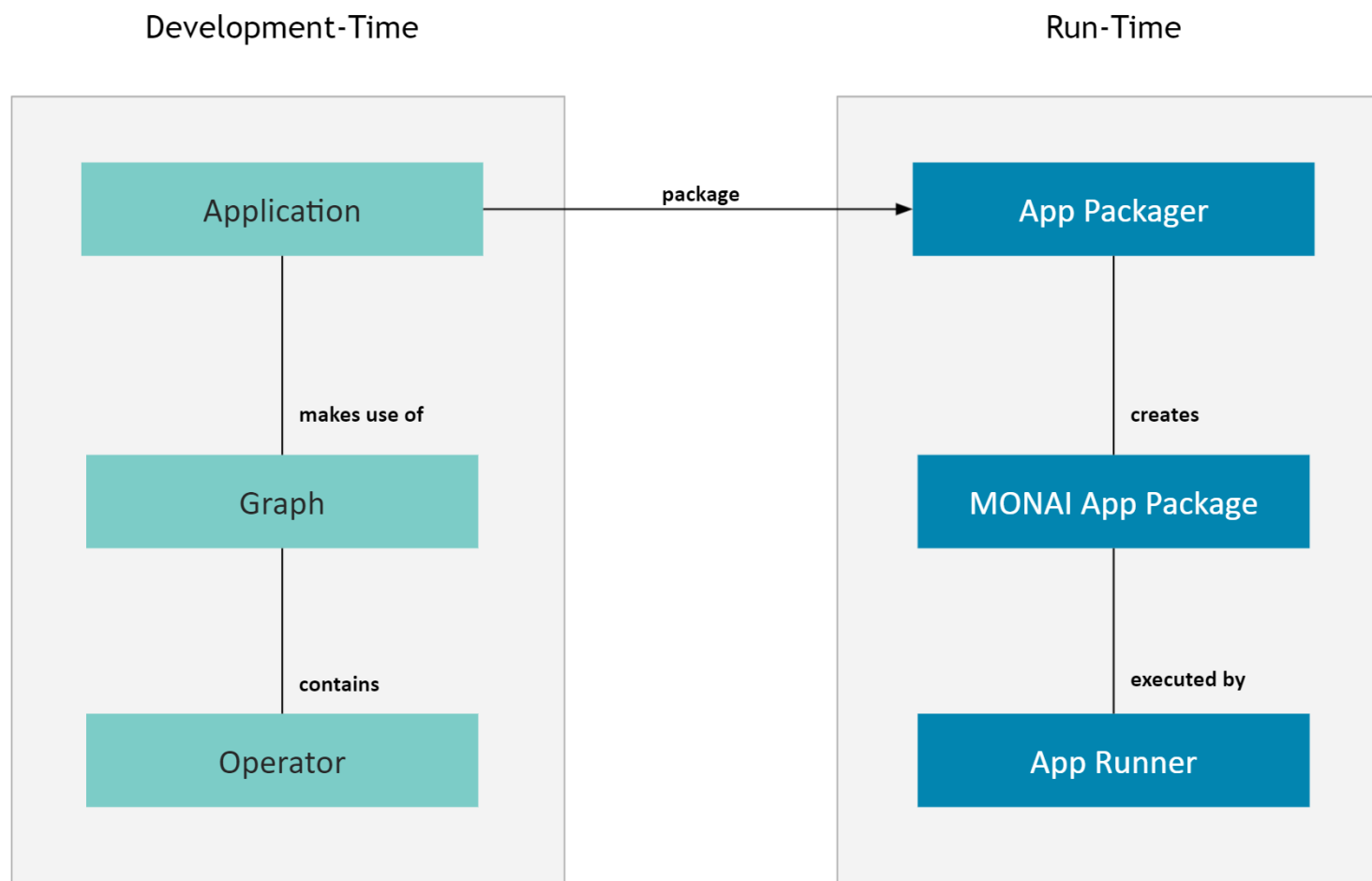
Monai Deploy SDK

MONAI SDK fue diseñado con los siguientes cuatro objetivos fundamentales:

- **Usabilidad:** Proporciona una API intuitiva para estructurar el código como un flujo de trabajo, es decir, como un grafo dirigido de operadores.
- **Composición:** Ofrece una colección de operadores como bloques de construcción para abordar tareas comunes en el ámbito sanitario.
- **Portabilidad:** Permite empaquetar la aplicación, los modelos asociados y los metadatos relevantes en un paquete autocontenido y fácil de trasladar entre entornos.
- **Preparación para producción:** Está diseñado para ejecutar aplicaciones en distintos entornos, desde el desarrollo hasta su uso en producción clínica.

Monai Deploy SDK

Diagrama de Arquitectura



Concepts

Development-Time Inference = inference application in the healthcare domain

Graph = Serie of operators (Transforms or Operations) [See Monai Core]

Run-Time = Use of final application

Source: <https://docs.monai.io/projects/monai-deploy-app-sdk/en/stable/introduction/architecture.html>

Monai Generative Models

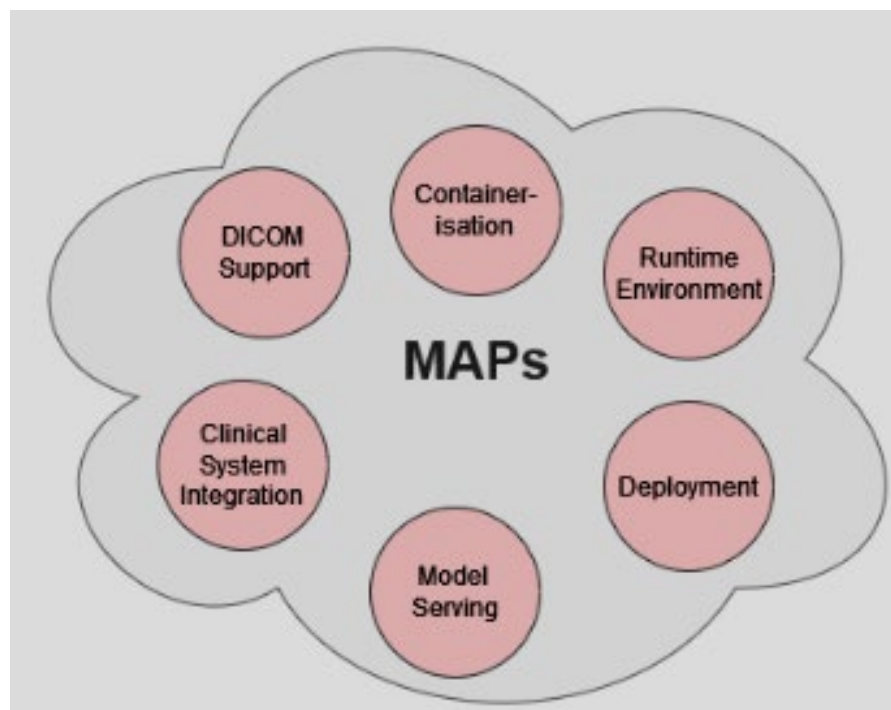
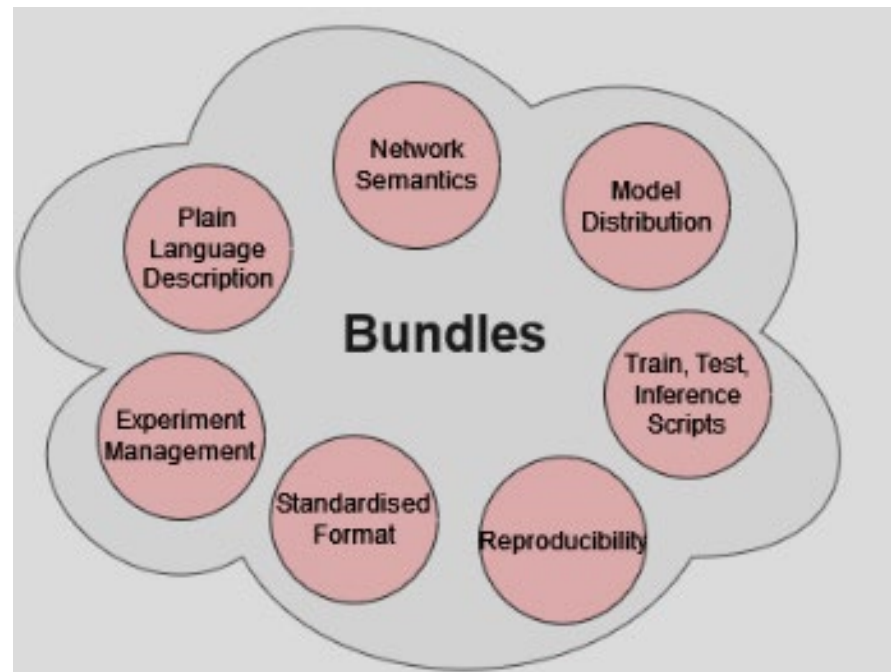


MONAI Model Zoo



Bundle vs. MAPs

- Bundle**: Es una forma de distribuir modelos entrenados con MONAI junto con sus metadatos, código, documentación y otros recursos, todo ello siguiendo una especificación y una estructura de archivos definida.
- MAP (Model Application Package)**: Se centra en la **despliegue**, **contenedorización**, **integración con sistemas clínicos existentes** y otras áreas clave para poner los modelos en uso real en entornos clínicos.



What is Model ZOO?

MONAI Model Zoo es un repositorio de modelos preentrenados en formato bundle

All Models

Brats mri axial slices generative diffusion

MONAI team

A generative model for creating 2D brain MRI axial slices from Gaussian noise based on BraTS dataset

[Model Details](#)

Brats mri generative diffusion

MONAI team

A generative model for creating 3D brain MRI from Gaussian noise based on BraTS dataset

[Model Details](#)

Brats mri segmentation

MONAI team

A pre-trained model for volumetric (3D) segmentation of brain tumor subregions from multimodal MRIs based on BraTS 2018 data

[Model Details](#)

Breast density classification

Center for Augmented Intelligence in Imaging, Mayo Clinic Florida

A pre-trained model for classifying breast images (mammograms)

[Model Details](#)

Endoscopic inbody classification

NVIDIA DLMED team

A pre-trained binary classification model for endoscopic inbody classification task

[Model Details](#)

Endoscopic tool segmentation

NVIDIA DLMED team

A pre-trained binary segmentation model for endoscopic tool segmentation

[Model Details](#)

Lung nodule ct detection

MONAI team

A pre-trained model for volumetric (3D) detection of the lung lesion from CT image on LUNA16 dataset

[Model Details](#)

Mednist gan

MONAI Team

This example of a GAN generator produces hand xray images like those in the MedNIST dataset

[Model Details](#)

Mednist reg

MONAI team

This is an example of a ResNet and spatial transformer for hand xray image registration

[Model Details](#)

What is Model ZOO?

Lung nodule ct detection

Download

×

Model Metadata:

Overview: A pre-trained model for volumetric (3D) detection of the lung lesion from CT image on LUNA16 dataset

Author(s): MONAI team

References:

Lin, Tsung-Yi, et al. 'Focal loss for dense object detection. ICCV 2017

Downloads: 2079

File Size: 148.1MB

Version: 0.5.9

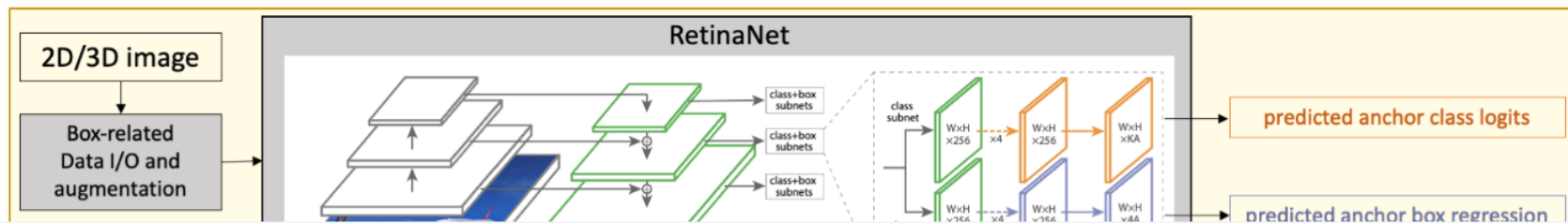
Model README:

Model Overview

A pre-trained model for volumetric (3D) detection of the lung nodule from CT image.

This model is trained on LUNA16 dataset (<https://luna16.grand-challenge.org/Home/>), using the RetinaNet (Lin, Tsung-Yi, et al. "Focal loss for dense object detection." ICCV 2017.

<https://arxiv.org/abs/1708.02002>).



Download

Close

What is Model ZOO?

MONAI Bundle Commands

In addition to the Pythonic APIs, a few command line interfaces (CLI) are provided to interact with the bundle. The CLI supports flexible use cases, such as overriding configs at runtime and predefining arguments in a file.

For more details usage instructions, visit the [MONAI Bundle Configuration Page](#).

Execute training:

```
python -m monai.bundle run --config_file configs/train.json
```

Please note that if the default dataset path is not modified with the actual path in the bundle config files, you can also override it by using `--dataset_dir`:

```
python -m monai.bundle run --config_file configs/train.json --dataset_dir <actual dataset path>
```

Override the `train` config to execute evaluation with the trained model:

```
python -m monai.bundle run --config_file "['configs/train.json', 'configs/evaluate.json']"
```

Execute inference on resampled LUNA16 images by setting `"whether_raw_luna16": false` in `inference.json`:

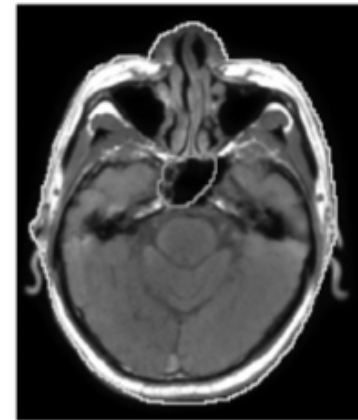
```
python -m monai.bundle run --config_file configs/inference.json
```

With the same command, we can execute inference on original LUNA16 images by setting `"whether_raw_luna16": true` in `inference.json`. Remember to also set `"data_list_file_path": "$@bundle_root + '/LUNA16_datsplit/mhd_original/dataset_fold0.json'"` and change `"dataset_dir"`.

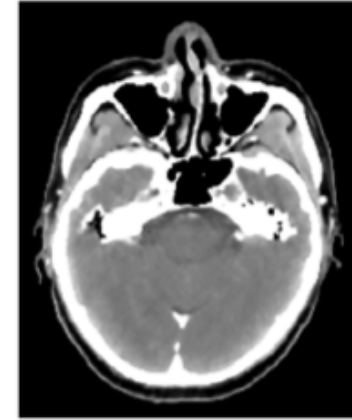
Monai Generative Models

MONAI incorporará **modelos generativos** que podrán utilizarse en tareas como:

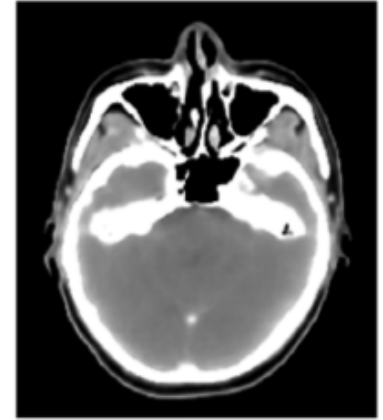
- **Síntesis de imágenes**
- **Transferencia de dominio**
- **Reconstrucción de imágenes por RMN (MRI)**
- **Detección de anomalías**



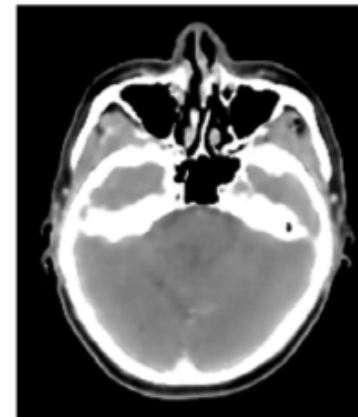
MRI



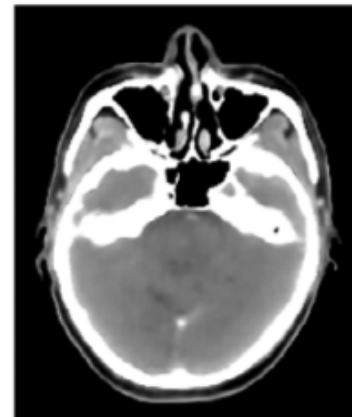
RealCT



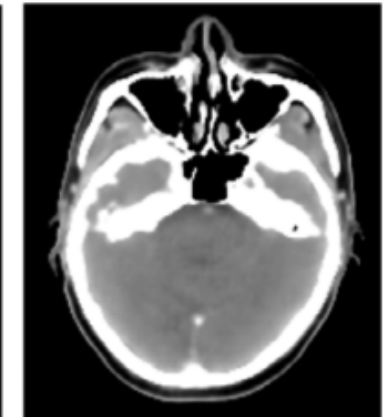
MSE-Only



MSE+GLCM(GTVnx)



MSE+GLCM(Outline)



MSE+GLCM(Outline+GTVnx)

Source: <https://aapm.onlinelibrary.wiley.com/doi/full/10.1002/mp.16850>

Código



<https://github.com/emi-dm/Medical-Image-PGTD>

Recursos útiles

- <https://monai.io/>
- <https://github.com/Project-MONAI/tutorials>
- <https://www.youtube.com/playlist?list=PLtoSVSQ2XzyCobzE6NvwjNpITsQyPUtfs>
- <https://colab.research.google.com/drive/1x8VjJXpkK7aYVrzt2l3eaYs9ewpHDaJr?usp=sharing> (Keras 3.0 integration with Monai)
- https://github.com/Project-MONAI/tutorials/blob/main/modules/3d_image_transforms.ipynb

¿Alguna pregunta?

