Deep learning methods for 2D in-vivo dose reconstruction with EPID detector

7 June 2024, Pisa - Workshop on AI-based in-vivo dosimetry with EPID

lorenzo.marini@phd.unipi.it

Overview

- **Pipeline of the analysis**
  - Data exploration
  - Data preprocessing
  - Deep learning architecture and training phase
  - Predictions of the model
  - Gamma-index analysis

- **Preliminary results**

- **Possible improvements**
  - Data augmentation technique
  - k-Cross validation
  - Custom loss function
  - Ensemble learning

- **Conclusions**

Pipeline of the analysis

Data exploration → **Data preprocessing** → **Data partition**

Data exploration
- number of sample
- pixel spacing
- physical size
- pixel units…

Data preprocessing
- Match of physical size
- Resize of the images
- Pixel normalization

Data partition
- 70% training
- 10% validation
- 20% test
- Data augmentation

↓

Training phase → **Prediction of the model** → **Evaluation of the performance**

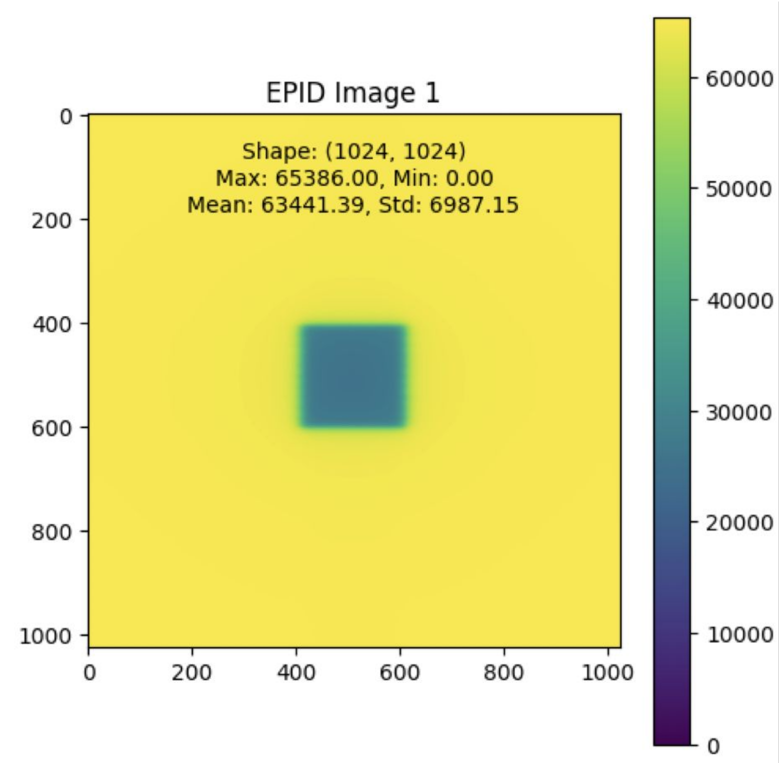
Training phase
- Unet model
- Loss function
- Epochs

Prediction of the model
- Prediction of the trained models on test dataset

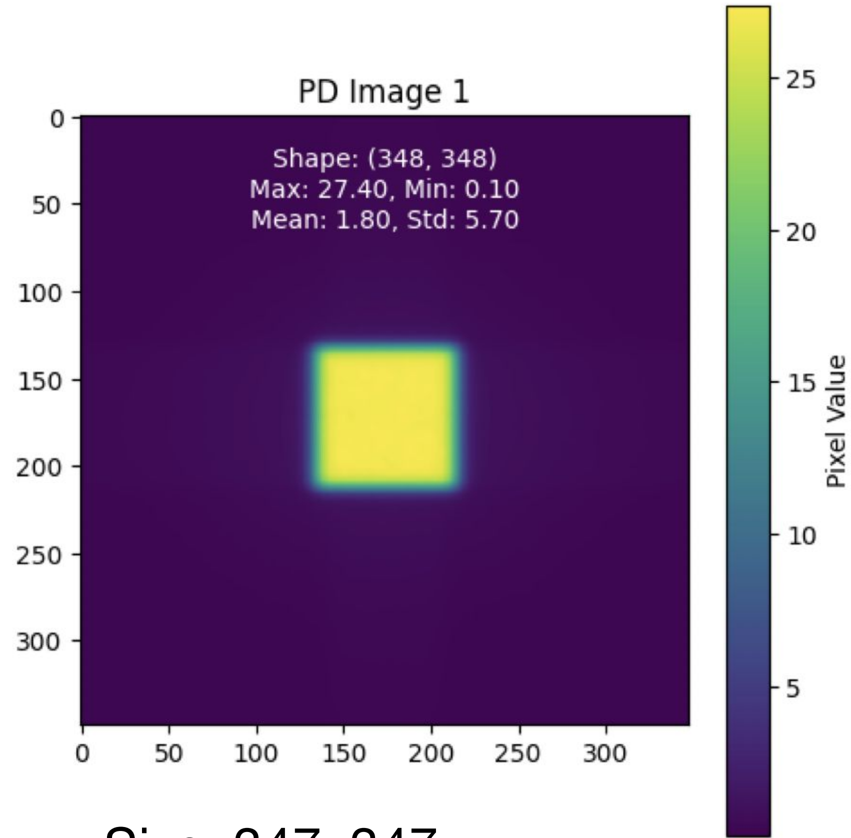
Evaluation of the performance
- Standard gamma-index analysis 3%/3mm

Data exploration

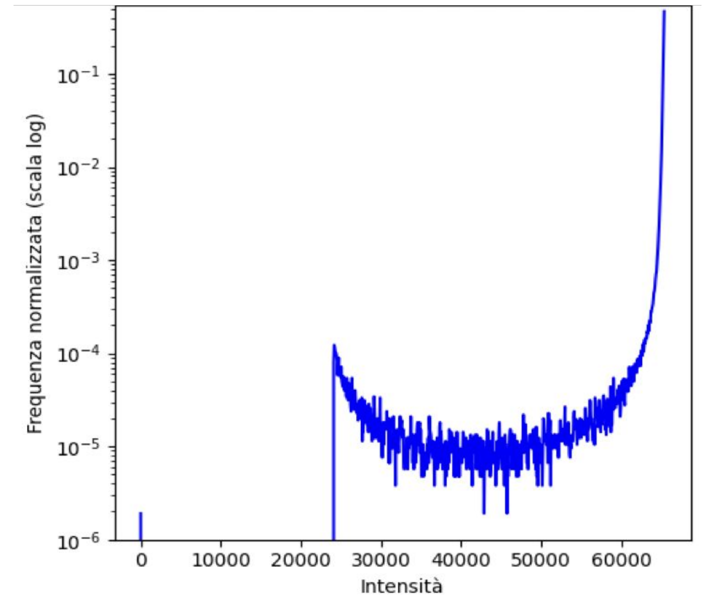
We collected **one hundred pairs** of EPID-PD images from various phantoms representing different material densities (lung, solid water, titanium, and bone), along with corresponding simulated dose images).

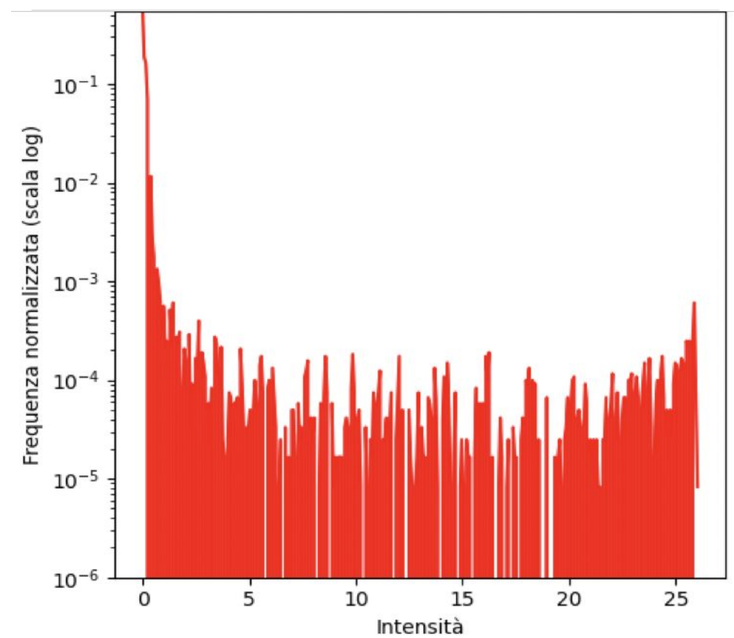


- Size: 1024x1024
- 16 bit
- Pixel spacing: [0.405, 0.405] mm
- Grey-scale pixel value

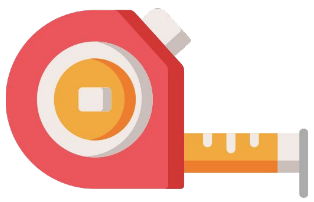


- Size: 347x347
- Range: [0, ≈30]
- Pixel spacing: [1.0, 1.0] mm
- Gray dose pixel value [cGy]





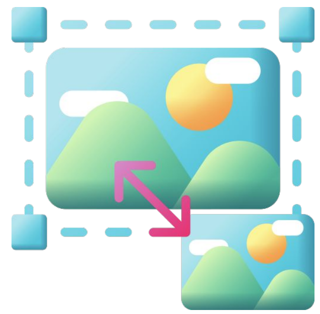
Data preprocessing



(1)

Matching of physical size

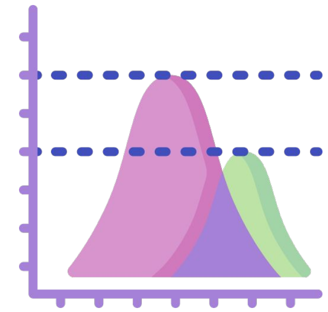
Do the images have the same dimensions?



(2)

Image resizing (256x256)

A trade-off between resolution and computing capacity



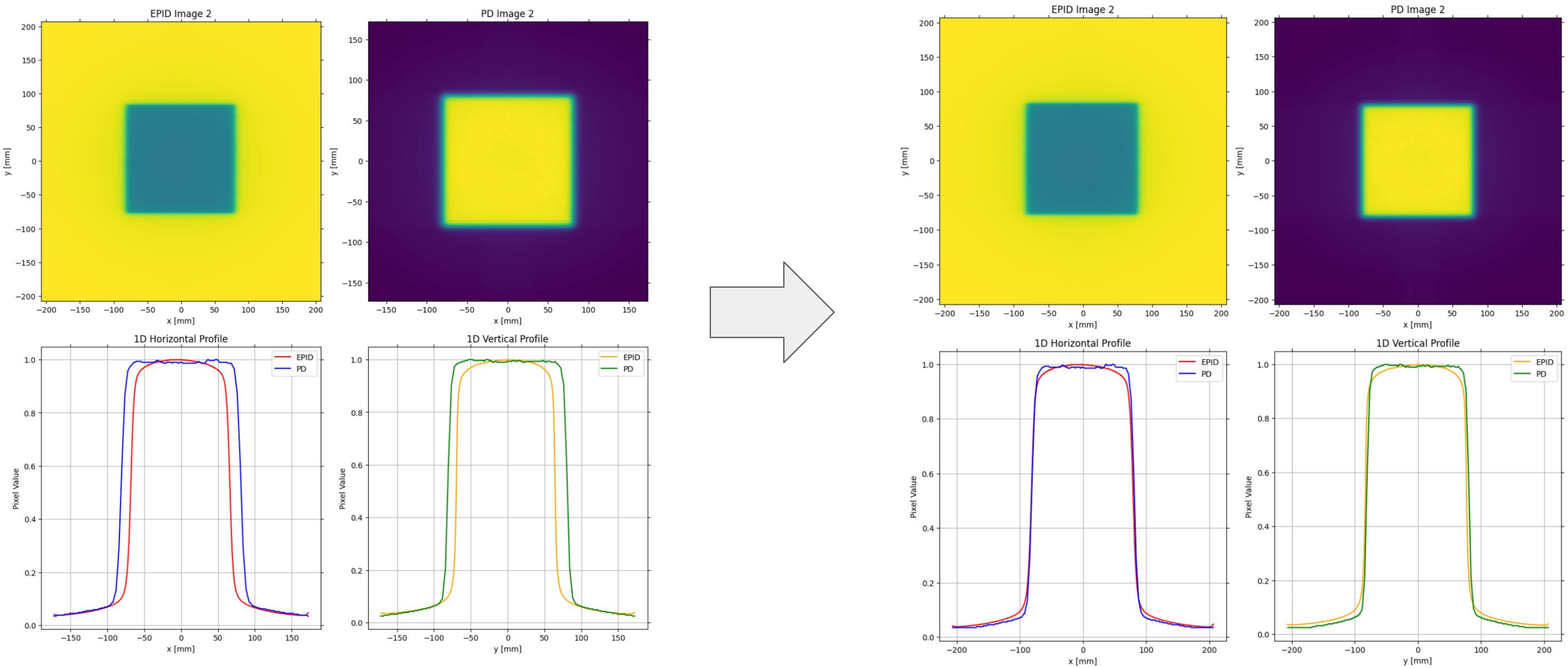
(3)

Pixel scaling → normalization

It can help the training phase of the network

1) Matching of physical size

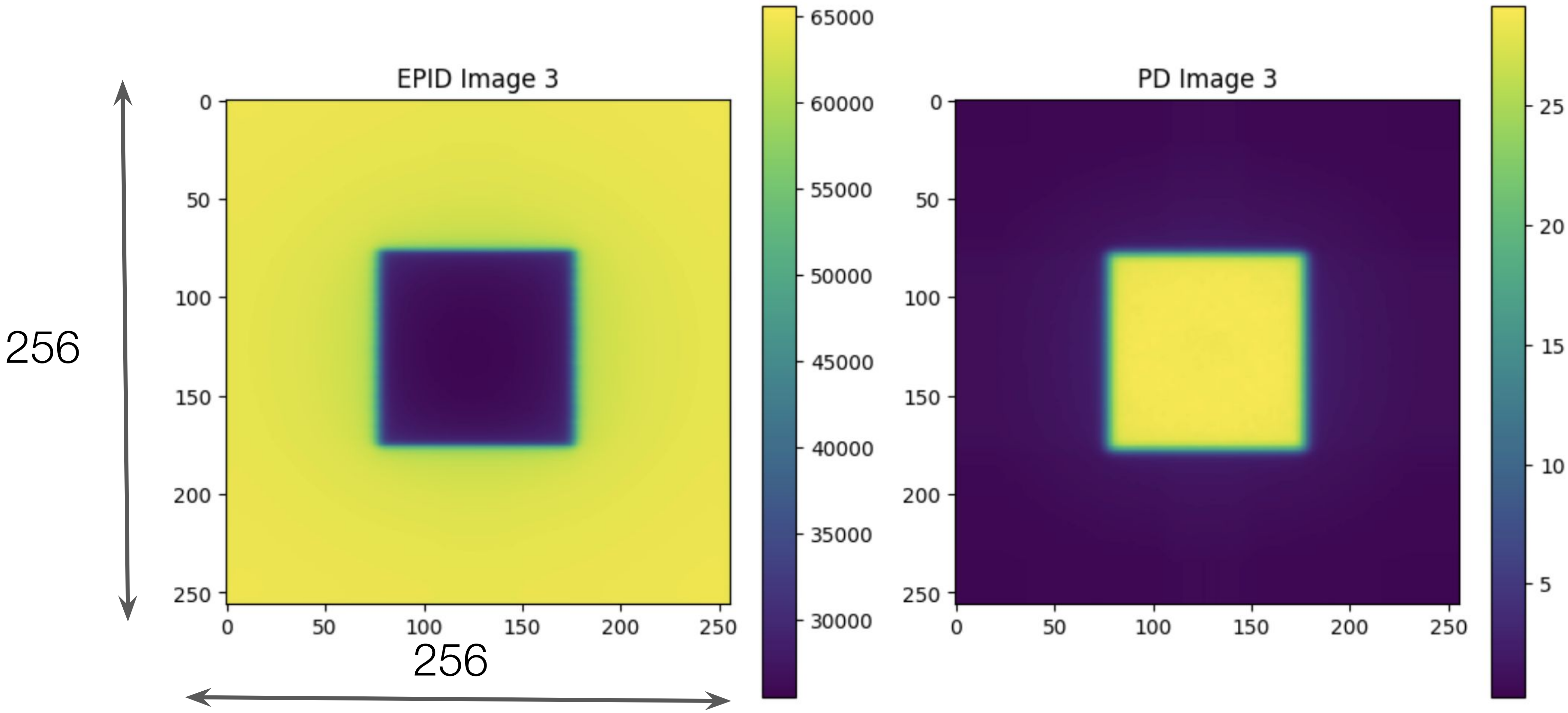
Do the images have the same dimensions?



- The **spatial dimensions** of the PDs differ from those of the EPIDs.
- Additionally, the squares within the PDs appear **slightly larger**.
- Some **background** pixels may are **missing**, possibly due to being cut off by the TPS.
- To address this problem, we add background pixels, ensuring that both images have the same spatial dimensions.

2) Image resizing → (256x256)

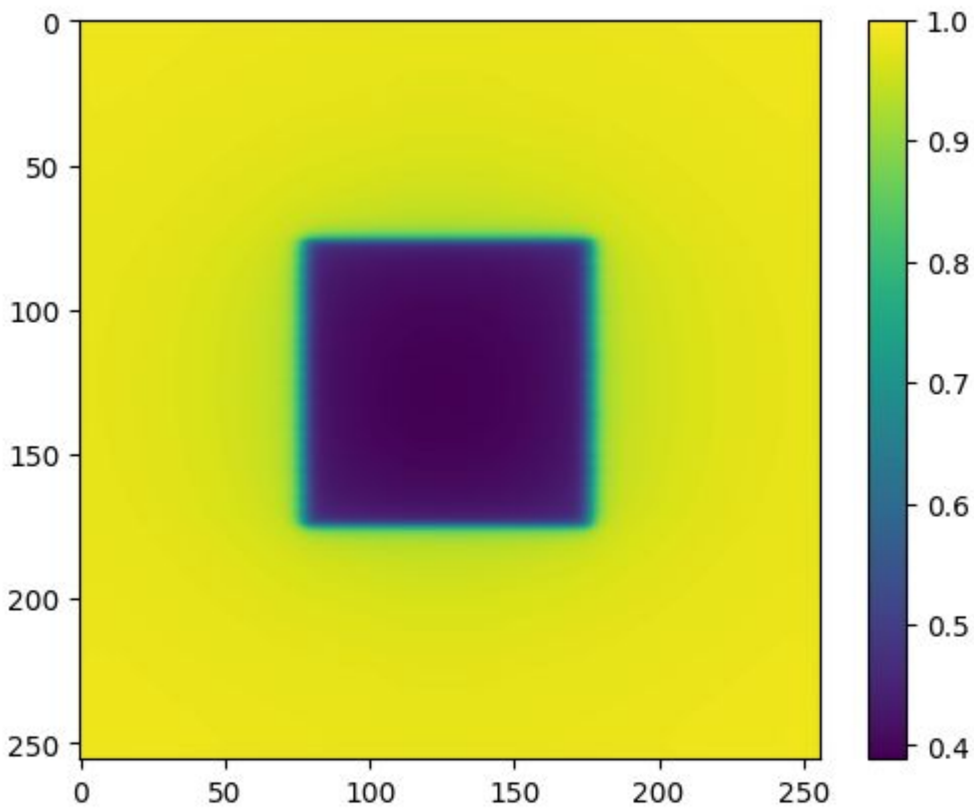
A trade-off between resolution and computing capacity



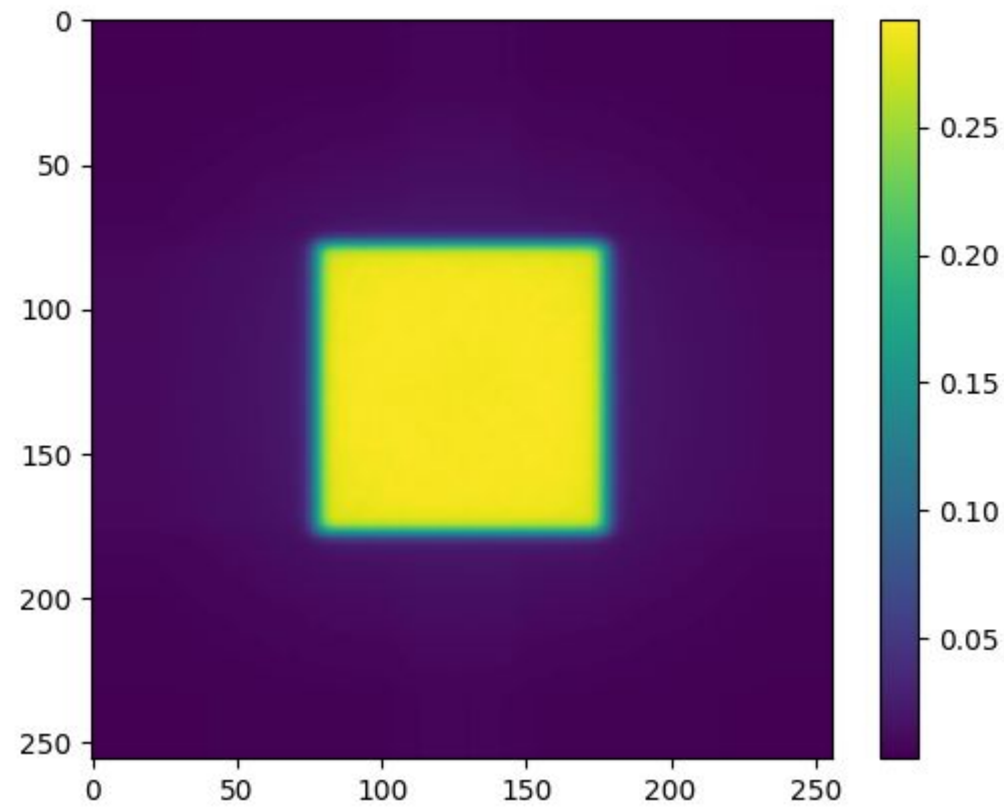
- **Image resizing** (using OpenCV, Python) refers to the scaling of images.
- Reduces the number of pixels, speeding up neural network training and reducing model complexity (**Training efficiency**).
- Lowers computational and memory requirements by decreasing image size.

3) Pixel scaling

It can help the training phase of the network



EPID → EPID/2^{16}

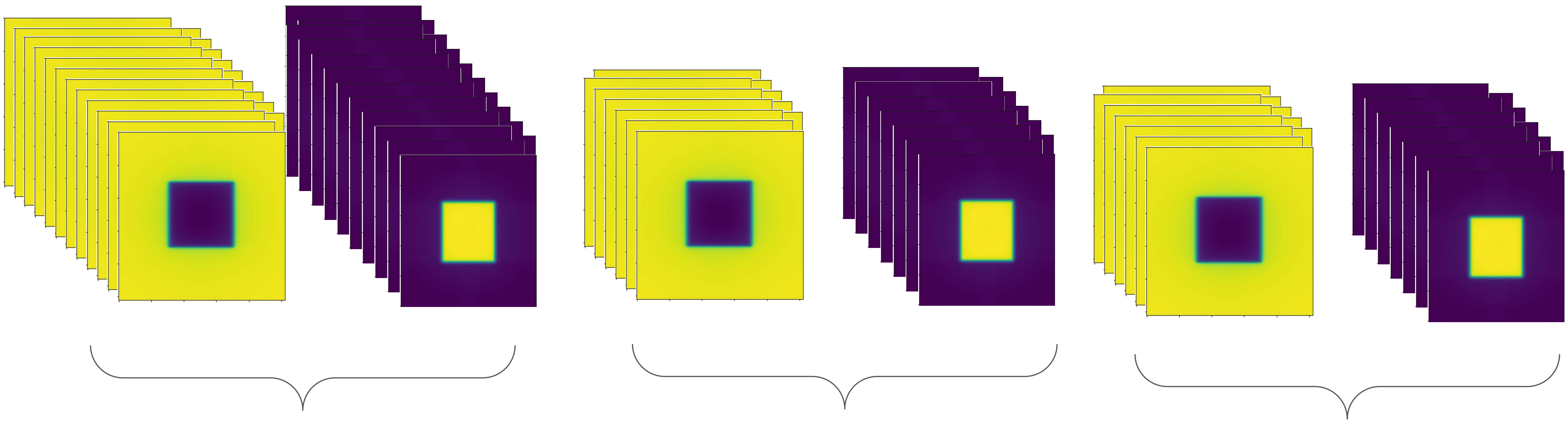


PD → PD/100

- Normalization is the process of converting an actual range of values which a numerical pixel can take, into a **standard range of values**, typically in the **interval [0, 1]**.
- *Why do we normalize?* It is not a strict requirement. However, in practice, it can lead to an **increased speed of learning** (Gradient descent, weight updates and numerical overflow)

Data partition

Once the dataset is ready to be processed by the neural network for the training phase, we split it into three different set:

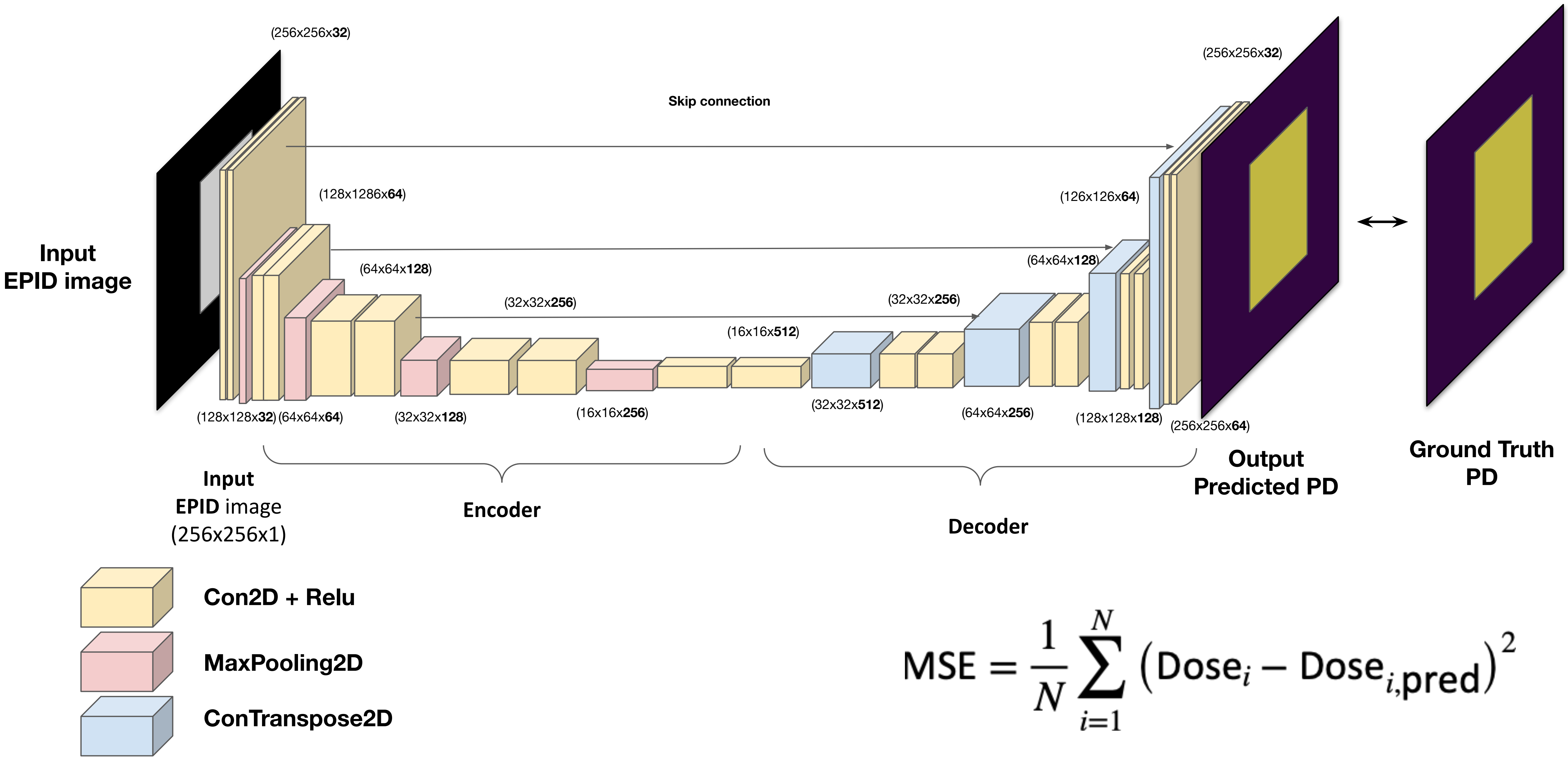


Training set
(70%)

Validation set
(15%)

Test set
(15%)

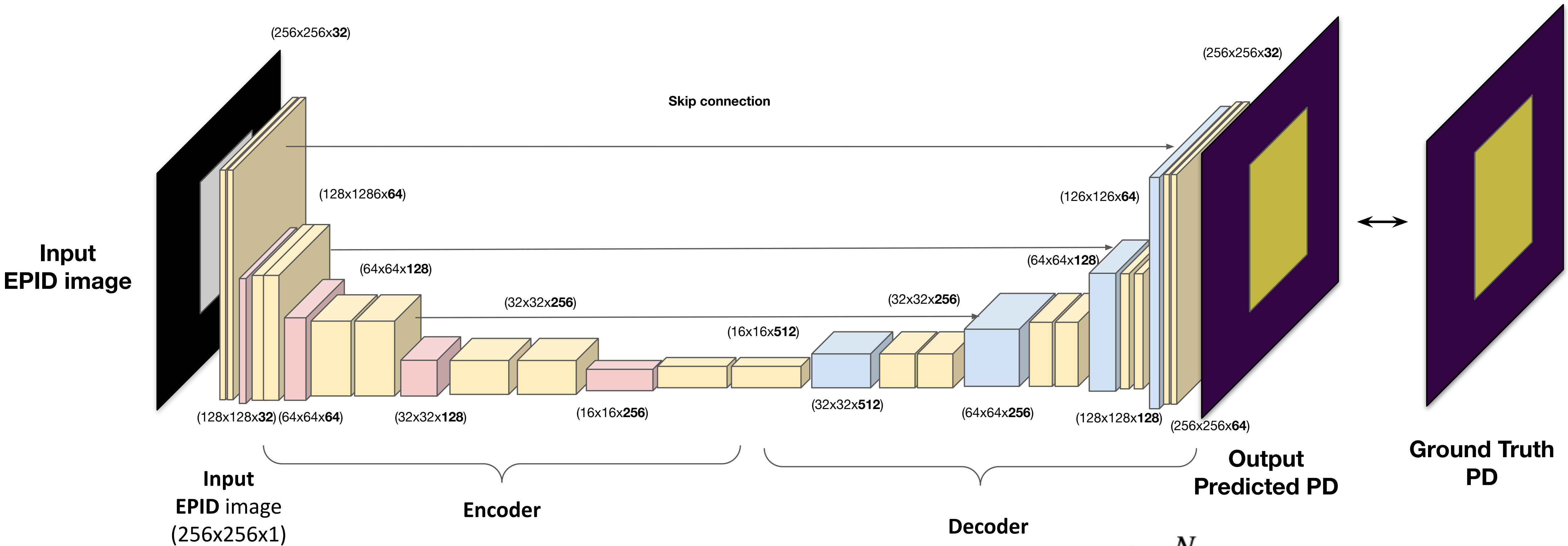
Unet architecture



$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Dose}_i - \mathrm{Dose}_{i,\mathrm{pred}}\right)^2$$

We developed a **U-net architecture** aiming at mapping EPID images into PD ones.
It is an **image regression problem**, i.e., a machine-learning technique that has the ability to predict continuous values within a specific range.

Deep learning architecture



$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\text{Dose}_i - \text{Dose}_{i,\text{pred}}\right)^2$$

Conv Block:

- **Conv2D:** Applies convolution with 3x3 filters and 'same' padding.
- **BatchNormalization**: Normalizes the activations of the previous layer to improve training stability.
- **Activation:** applies ReLU activation function to introduce non-linearity.
- **SpatialDropout2D**: Randomly drops spatial units to prevent overfitting.

Unet parameters

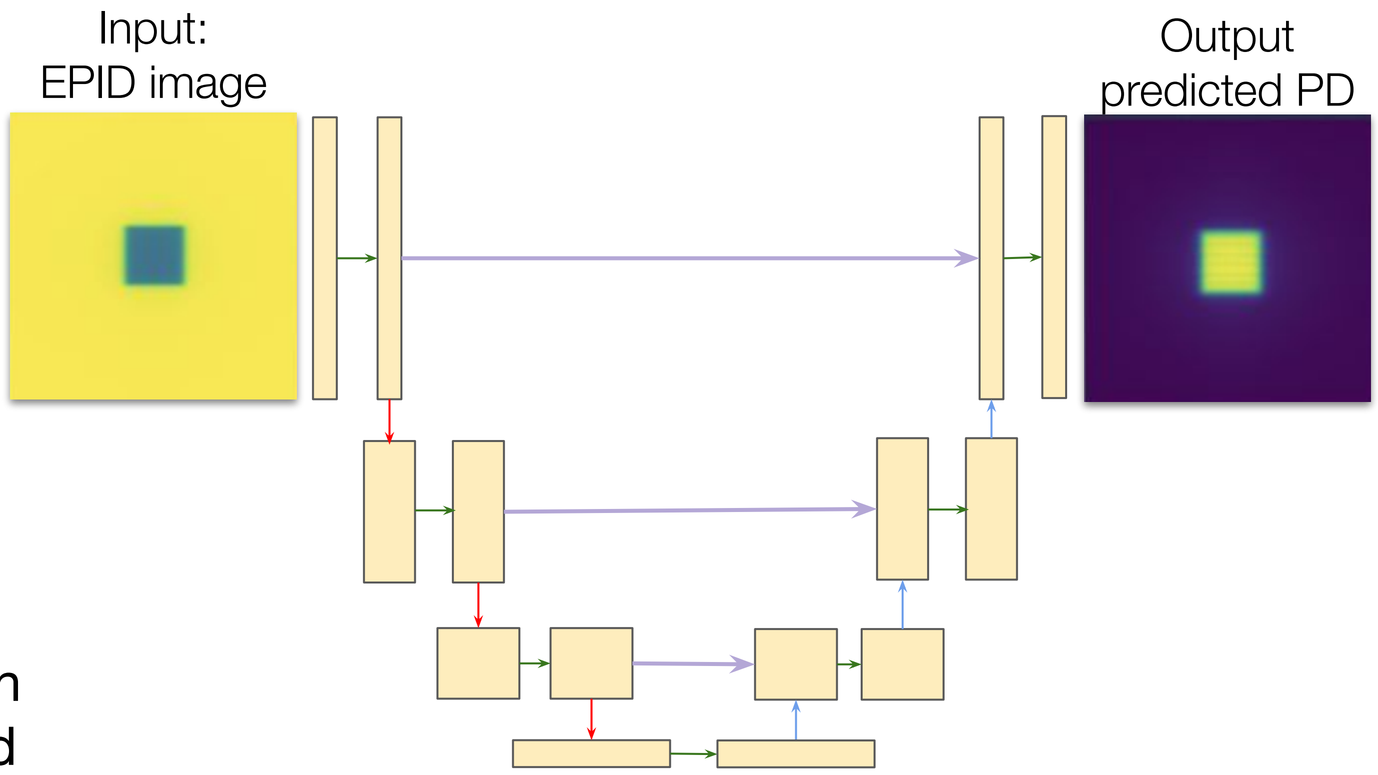
UNet Architecture:
Downsampling: Successive conv_block and MaxPooling2D((2, 2)) layers to reduce spatial dimensions and extract features.

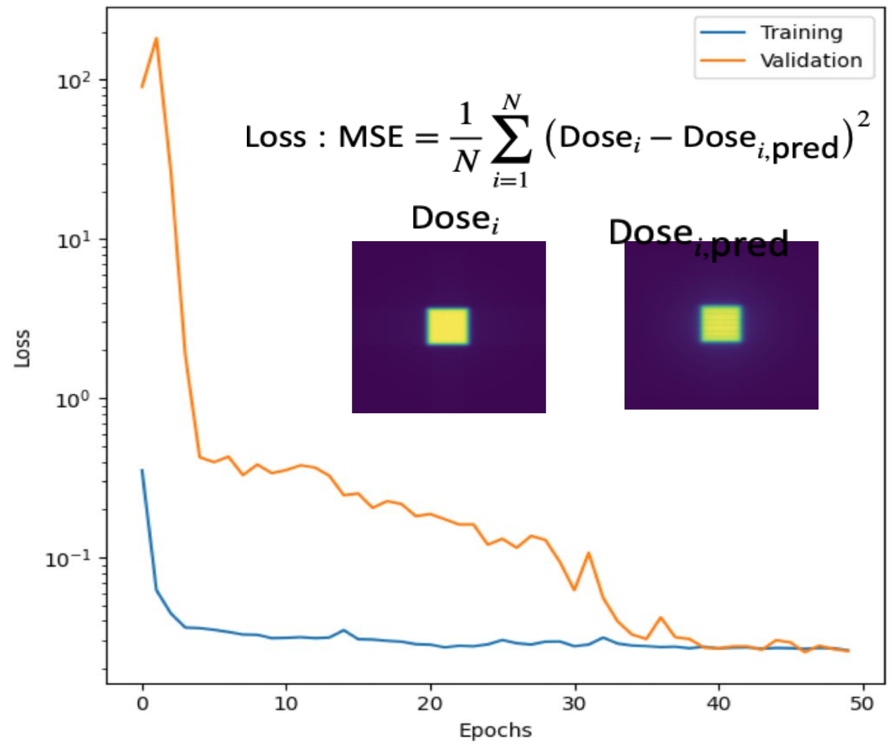
Bottleneck: Deepest part of the network with highest feature extraction (256 filters).

Upsampling: Conv2DTranspose layers to increase spatial dimensions, concatenating with corresponding downsampled layers for detailed reconstruction.

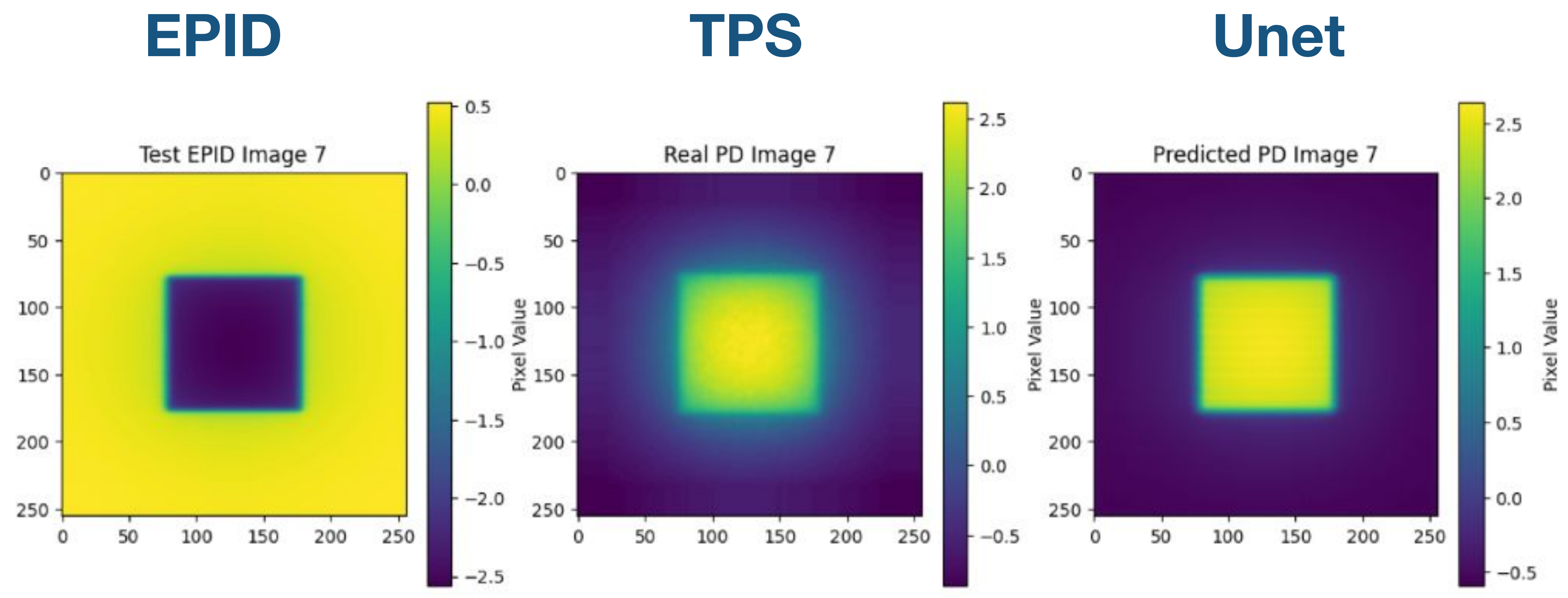
Output Layer:
Conv2D(1, (1, 1), activation=**'linear'**): Generates the final output with a single channel using linear activation.
Resizing(256, 256): Resizes the output to a fixed dimension of 256x256.

Compilation: Compiles the model with Adam optimizer, mean squared error loss, and mean absolute error as a metric.

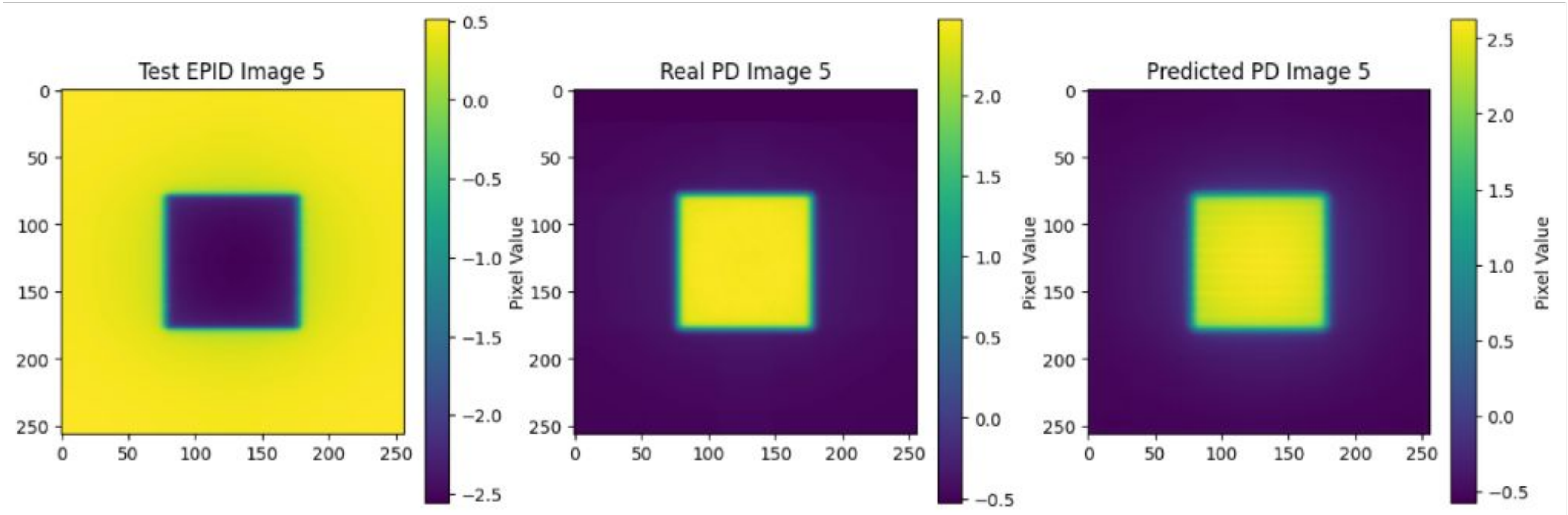




Predictions of the model

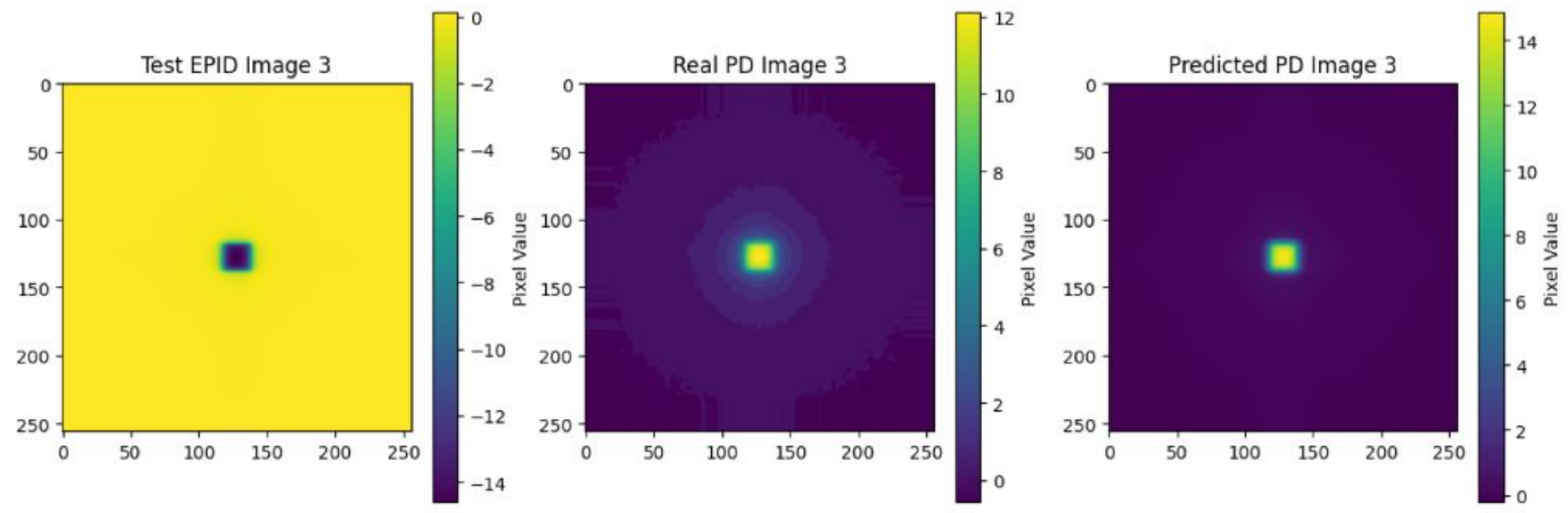


EPID TPS Unet



Predictions of the model

EPID | **TPS** | **Unet**

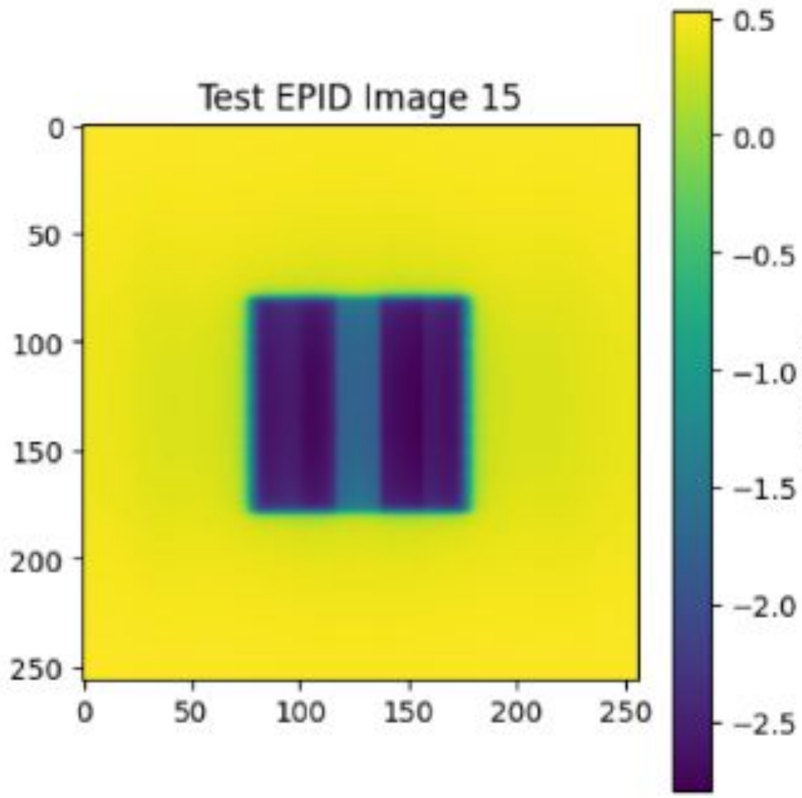


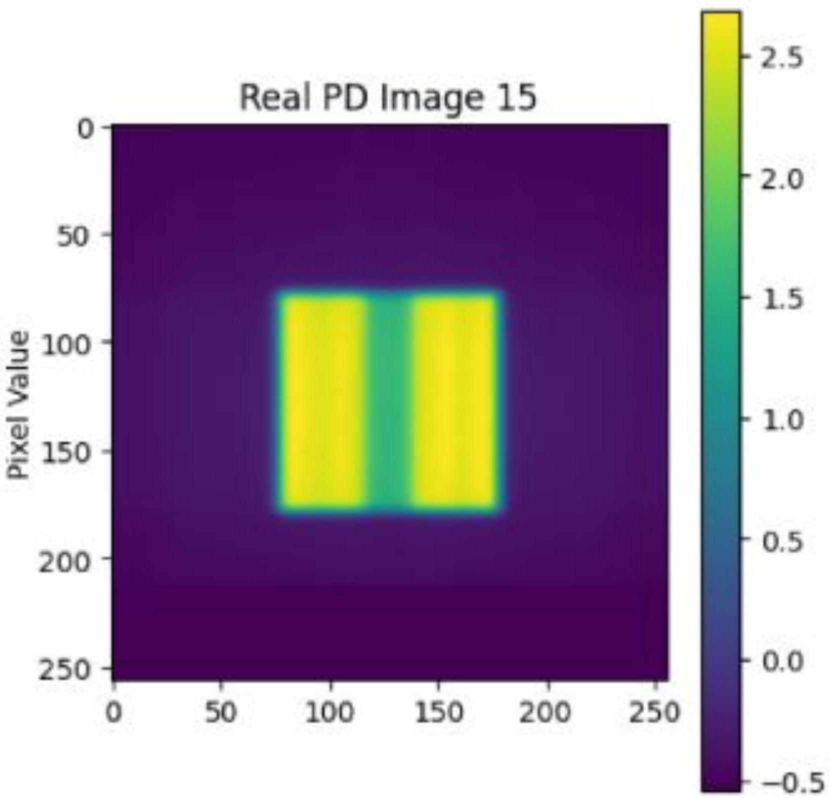
Predictions of the model

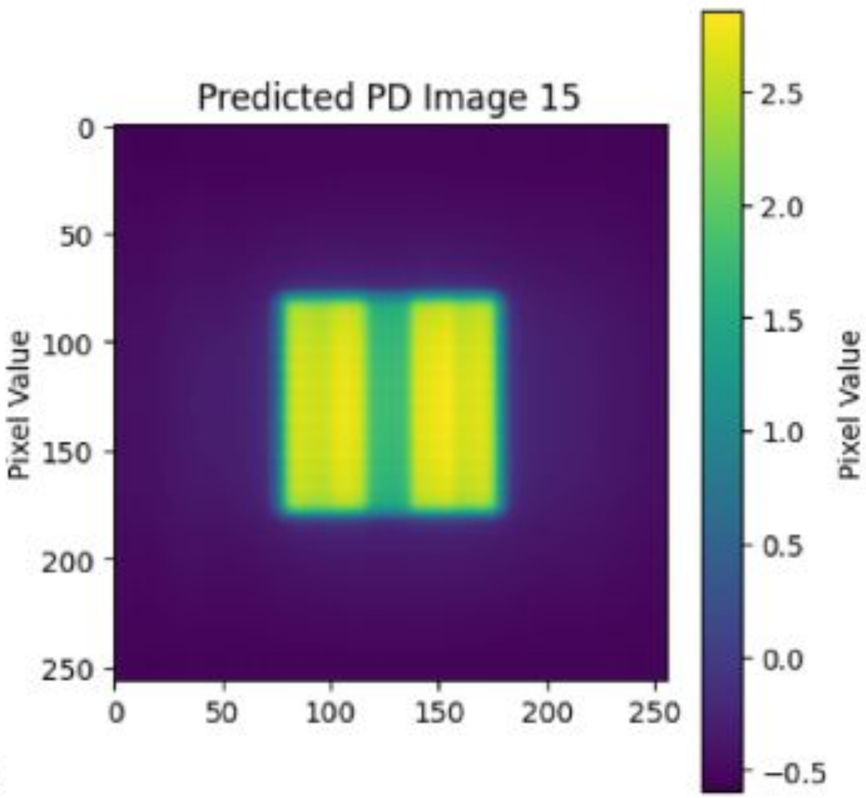
EPID

TPS

Unet

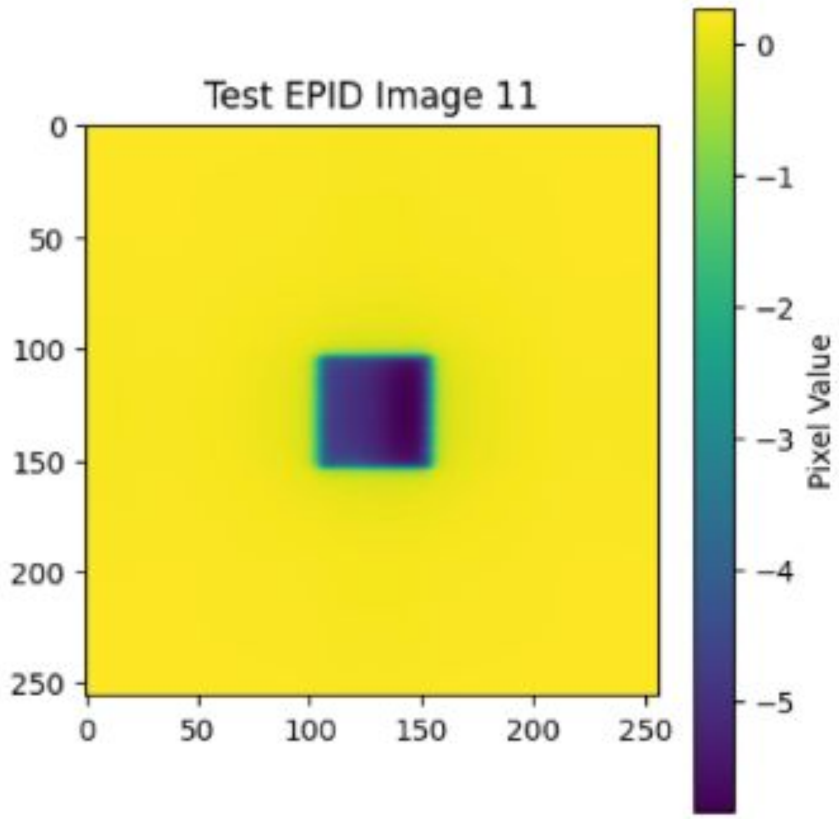




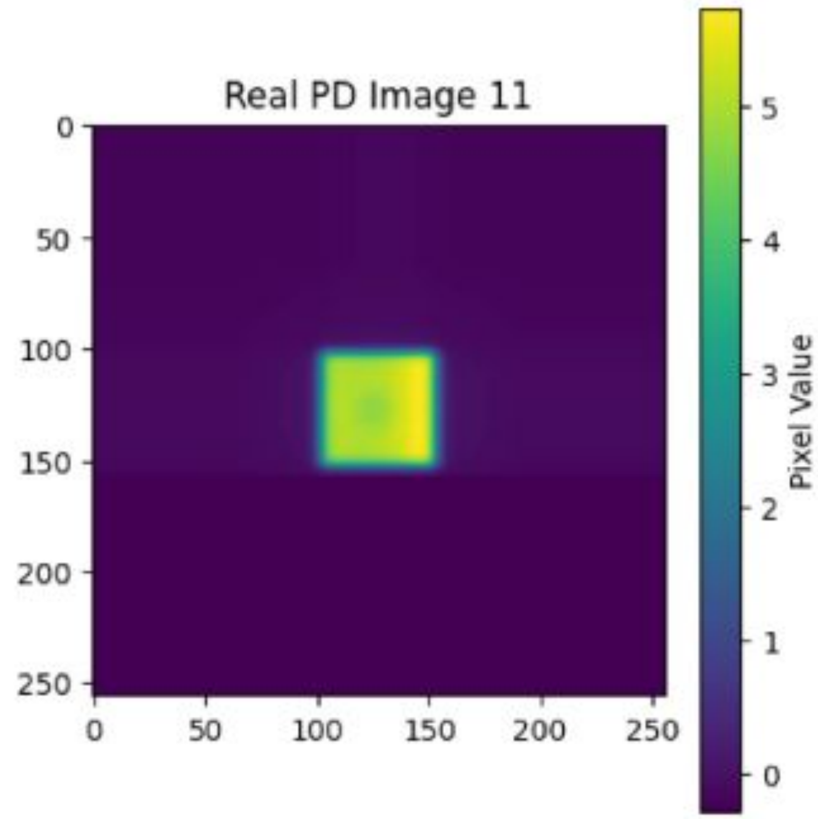


Predictions of the model

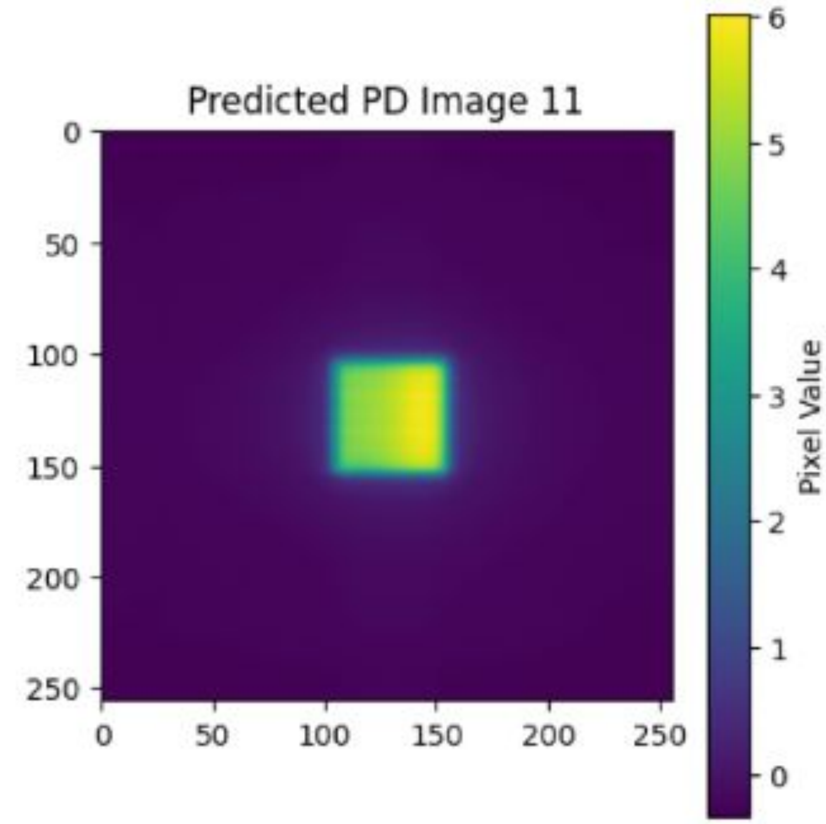
EPID



TPS

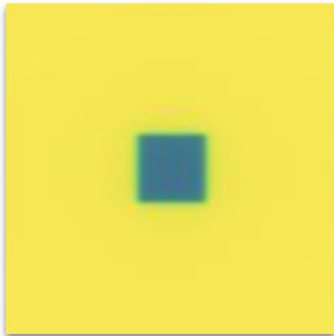


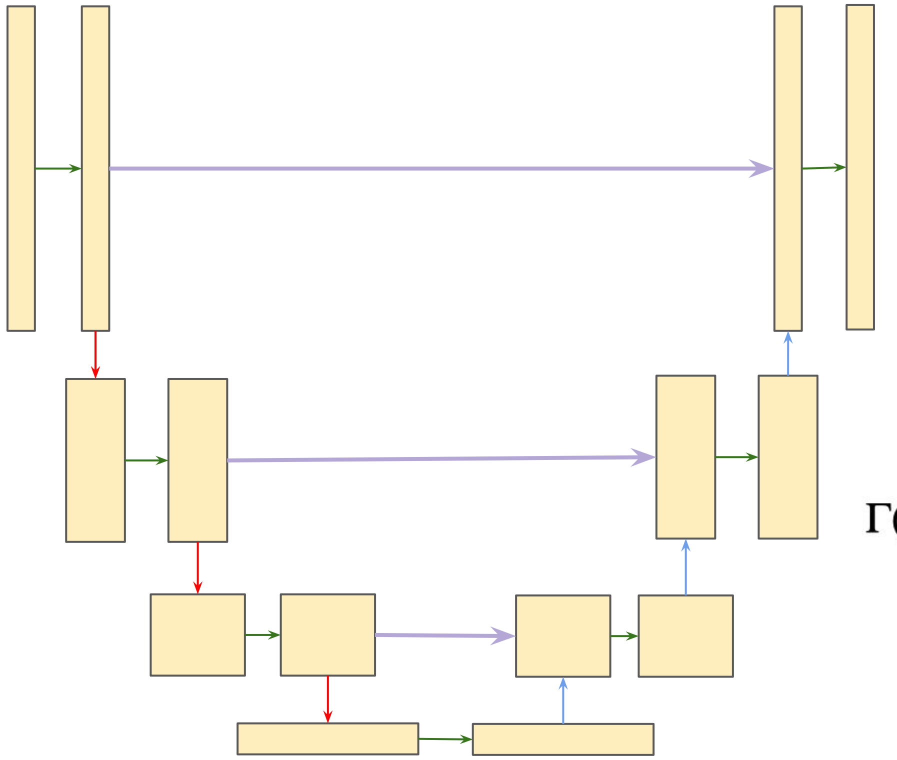
Unet



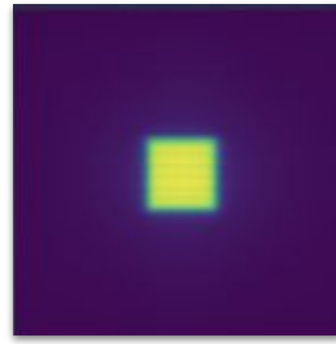
Gamma-index analysis

Input: EPID image





Output: predicted PD

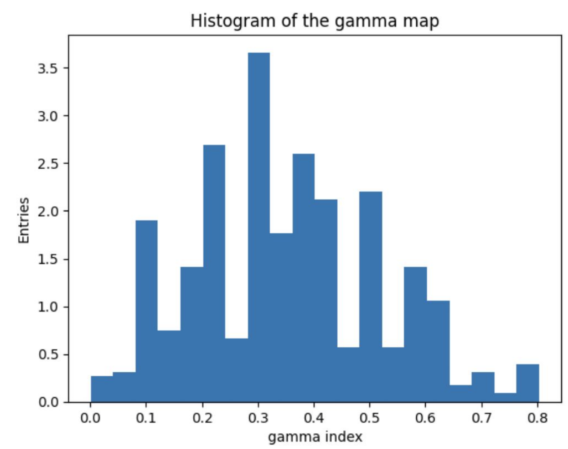


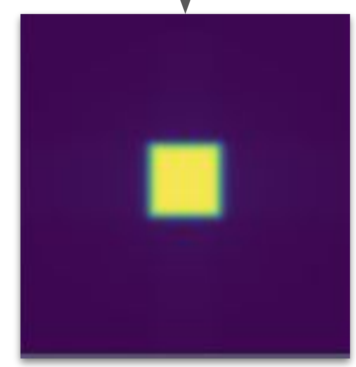
VS

$$\Gamma(\mathbf{r}_{\text{real}}, \mathbf{r}_{\text{pred}}) = \sqrt{\frac{\Delta \mathbf{r}^2(\mathbf{r}_{\text{real}}, \mathbf{r}_{\text{pred}})}{\delta r^2} + \frac{\Delta D^2(\mathbf{r}_{\text{real}}, \mathbf{r}_{\text{pred}})}{\delta D^2}}$$

$$\gamma(\mathbf{r}_{\text{real}}, \mathbf{r}_{\text{pred}}) = \min\{\Gamma(\mathbf{r}_{\text{real}}, \mathbf{r}_{\text{pred}})\}\, \forall \mathbf{r}_{\text{real}}$$

$$\gamma\text{-passing rate} = \frac{\#\gamma < 1}{\#\text{tot}\gamma}$$

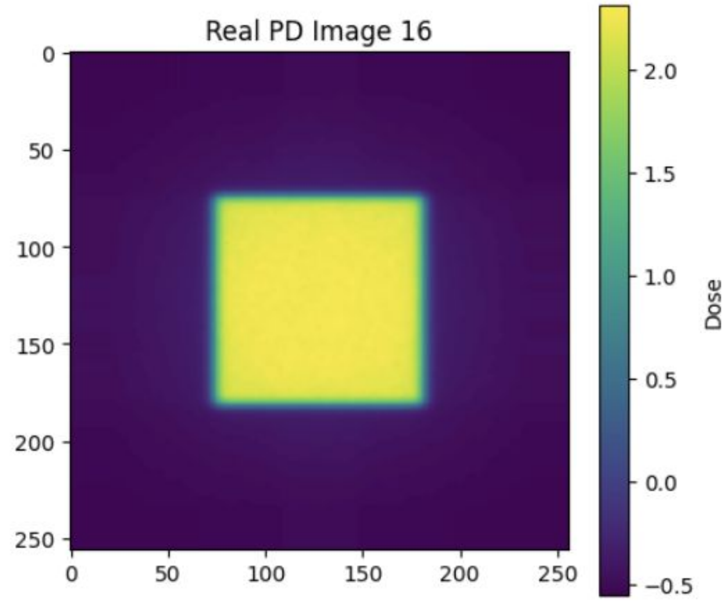


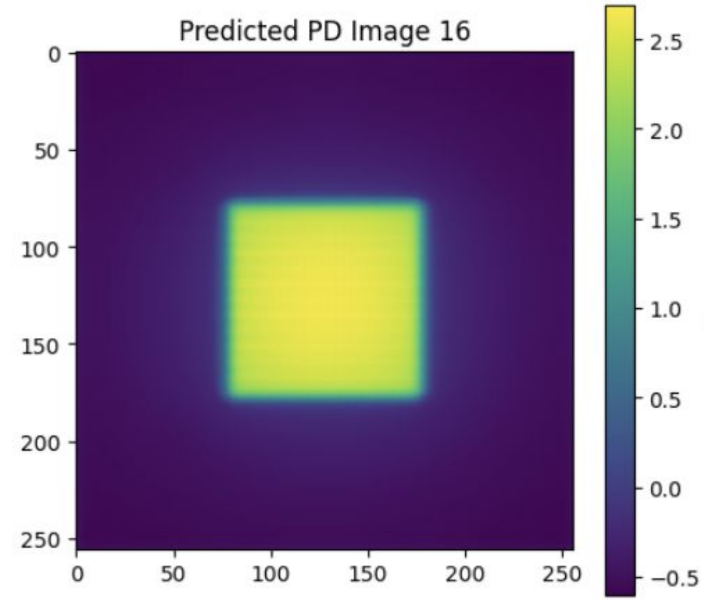


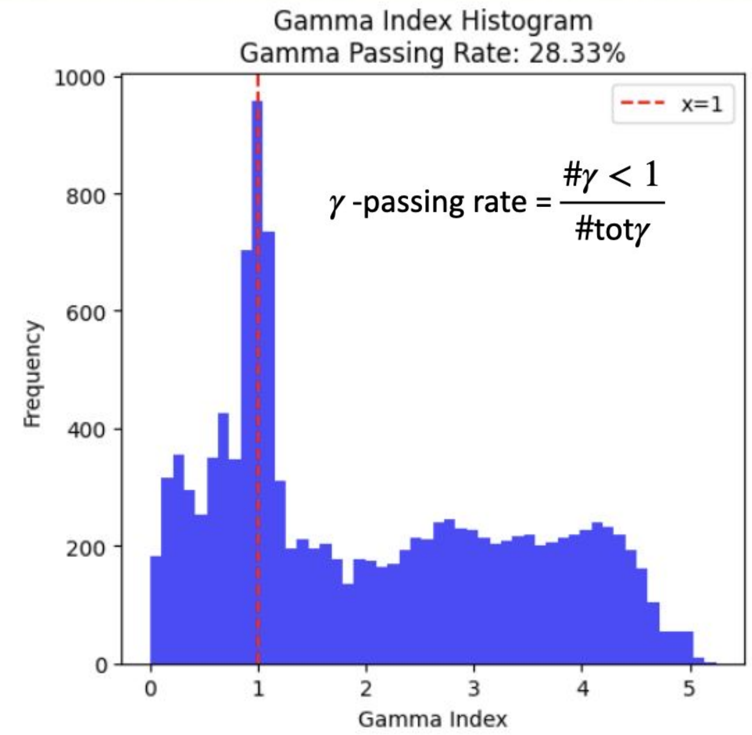
Ground truth: real PD

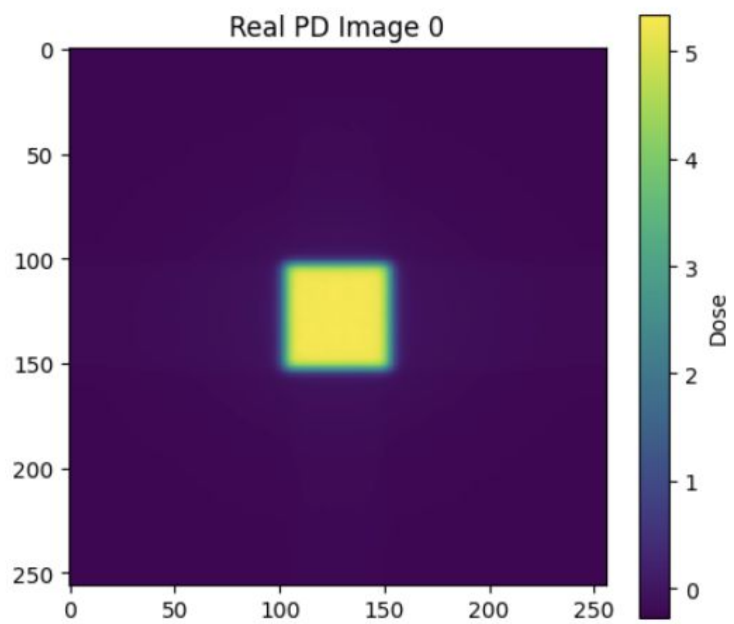
A standard gamma analysis of 3%/3 mm was performed on the PD predicted by the DL network and the simulated PD in the test dataset, leading to a mean gamma pass rates of **(82.30 ± 4.80)%**.

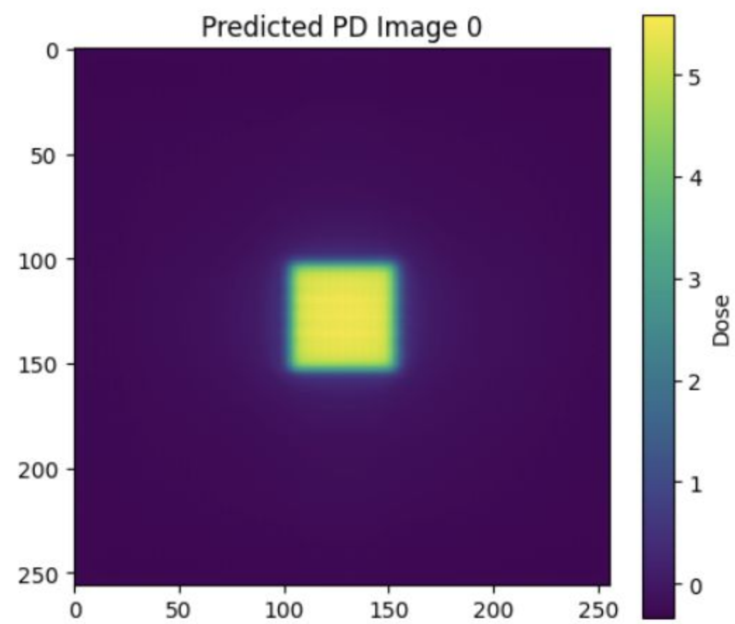
Gamma-index: worse and best cases

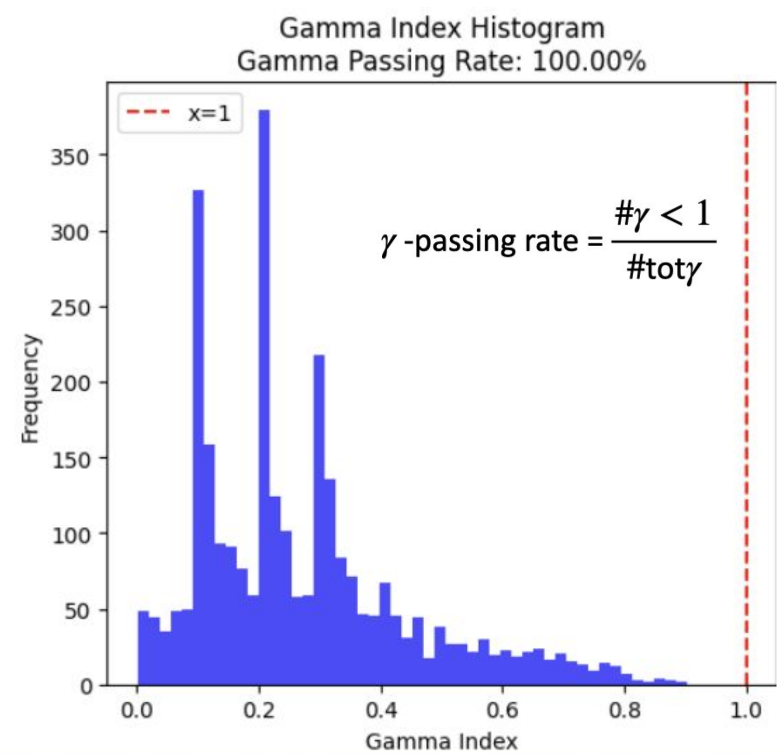




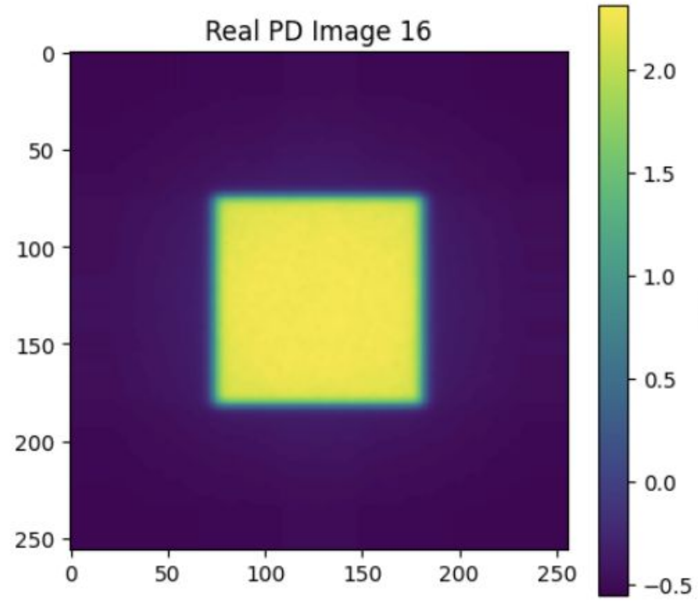


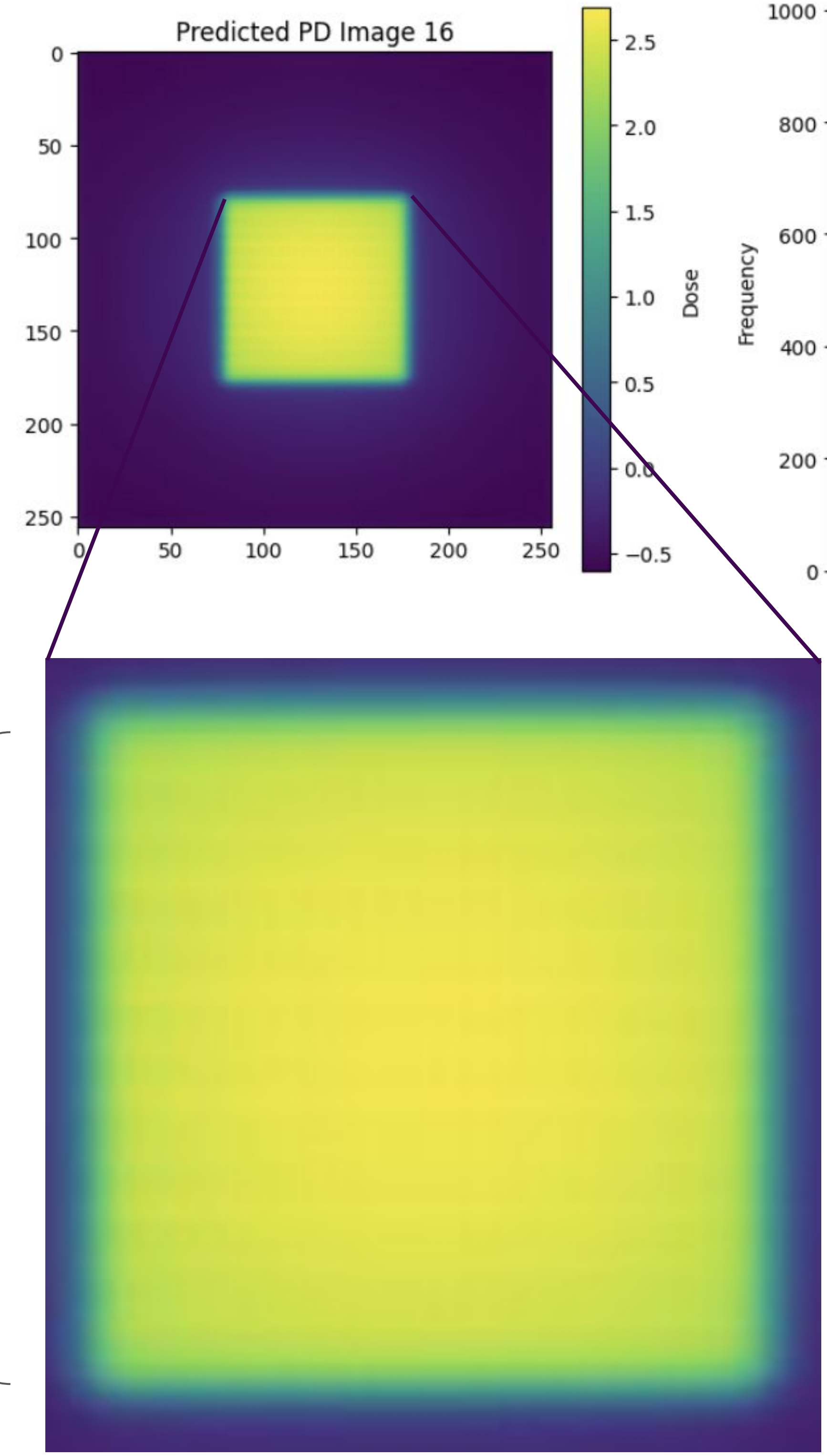


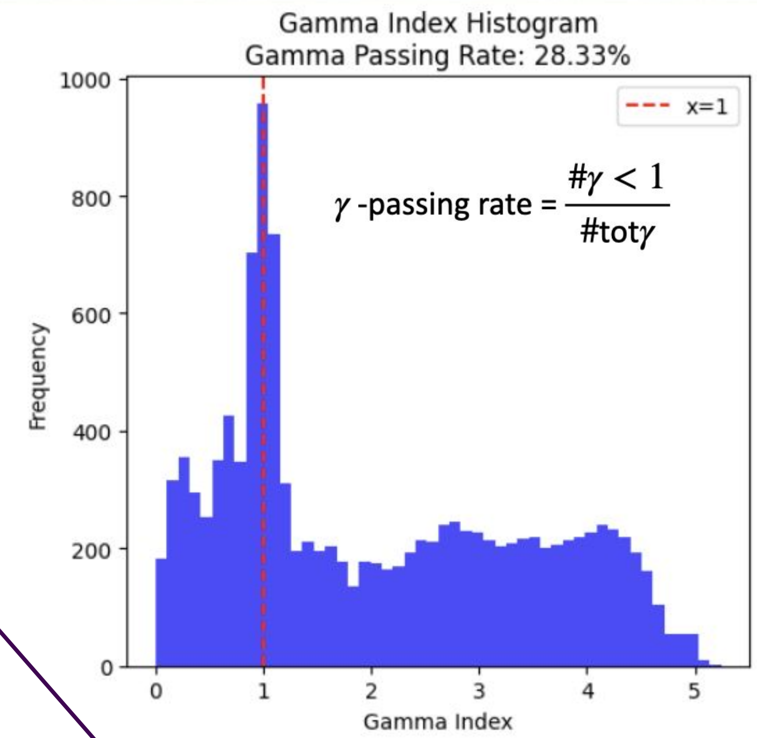




Gamma-index: worse and best cases





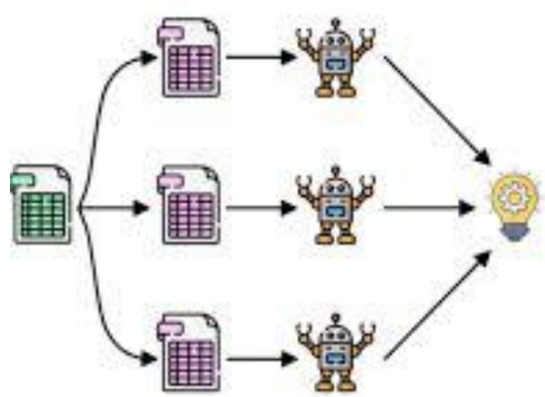


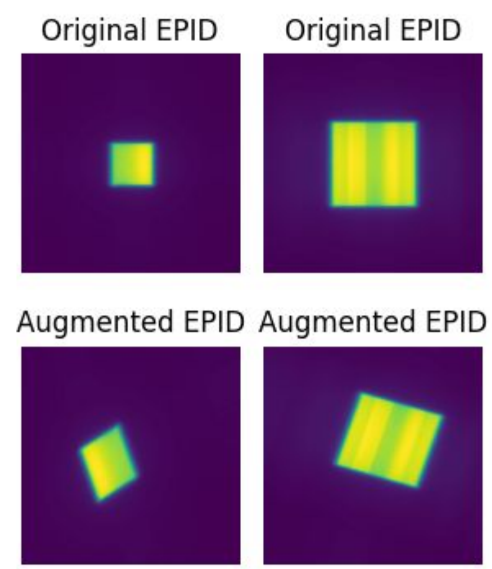
The reconstructed PDs show some **horizontal patterns**.
The causes need to be investigated…

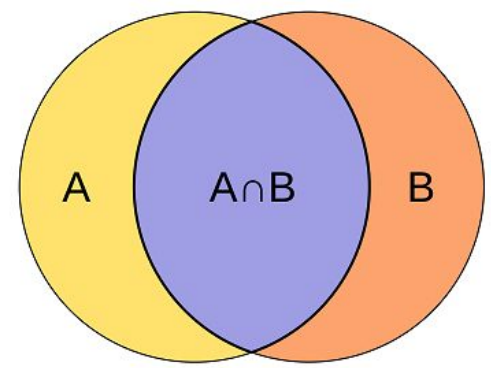
Improvements

Possible improvements:

- **Dataset:**
  - Correction of artefacts
  - Increase the data size
- **Deep learning model**
  - Data augmentation
  - Cross validation
  - Ensemble learning
  - Custom loss function



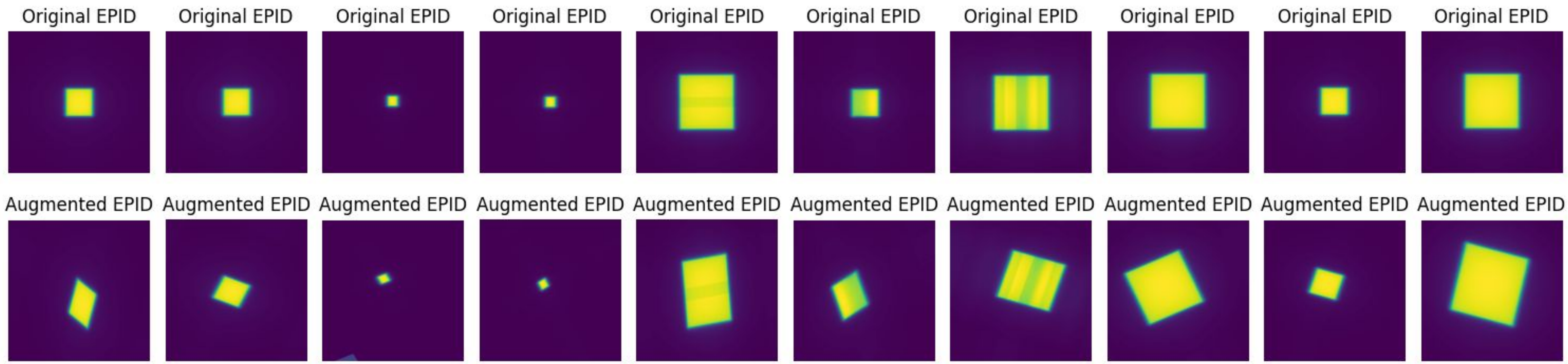




$$\text{Dice coefficient}(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|}$$

Data augmentation

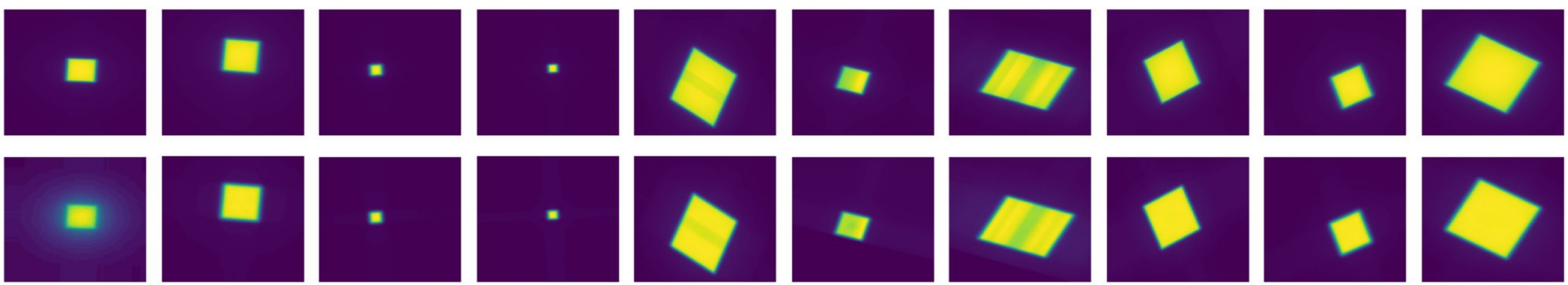
```python
data_augmentation = ImageDataGenerator(
    rotation_range=30,        # Rotazione massima di 30 gradi
    width_shift_range=0.1,    # Spostamento massimo del 10% della larghezza
    height_shift_range=0.1,   # Spostamento massimo del 10% dell'altezza
    shear_range=0.1,          # Applicazione di taglio massimo del 10%
    zoom_range=0.3,           # Zoom massimo del 30%
    horizontal_flip=True,     # Ribaltamento orizzontale casuale
    vertical_flip=True,       # Ribaltamento verticale casuale
    fill_mode='nearest'       # Modalità di riempimento dei pixel oltre i bordi
)
```



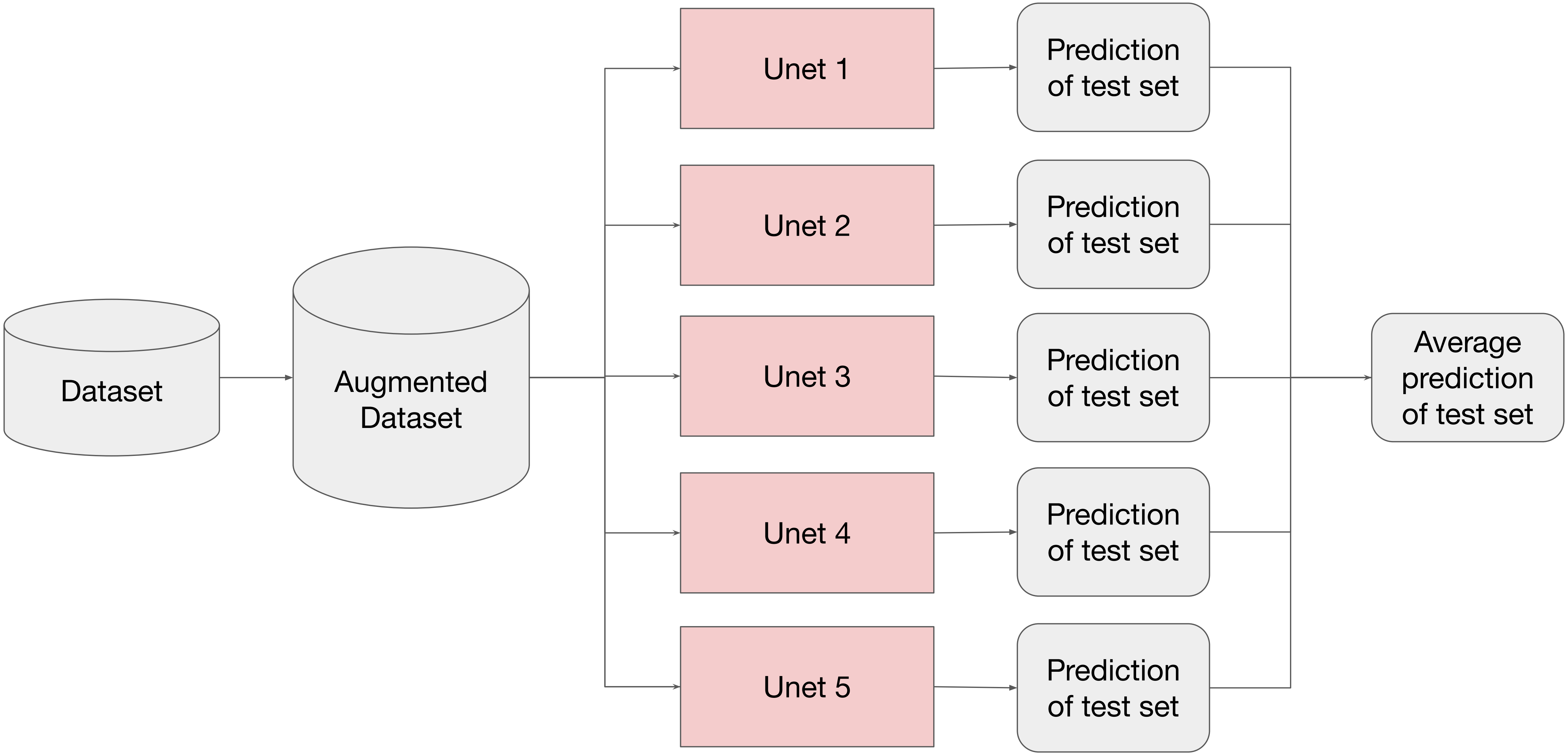
Warning: it is important that each pair of epid-pd images is subject to the <u>same</u> geometric transformation

EPID

PD



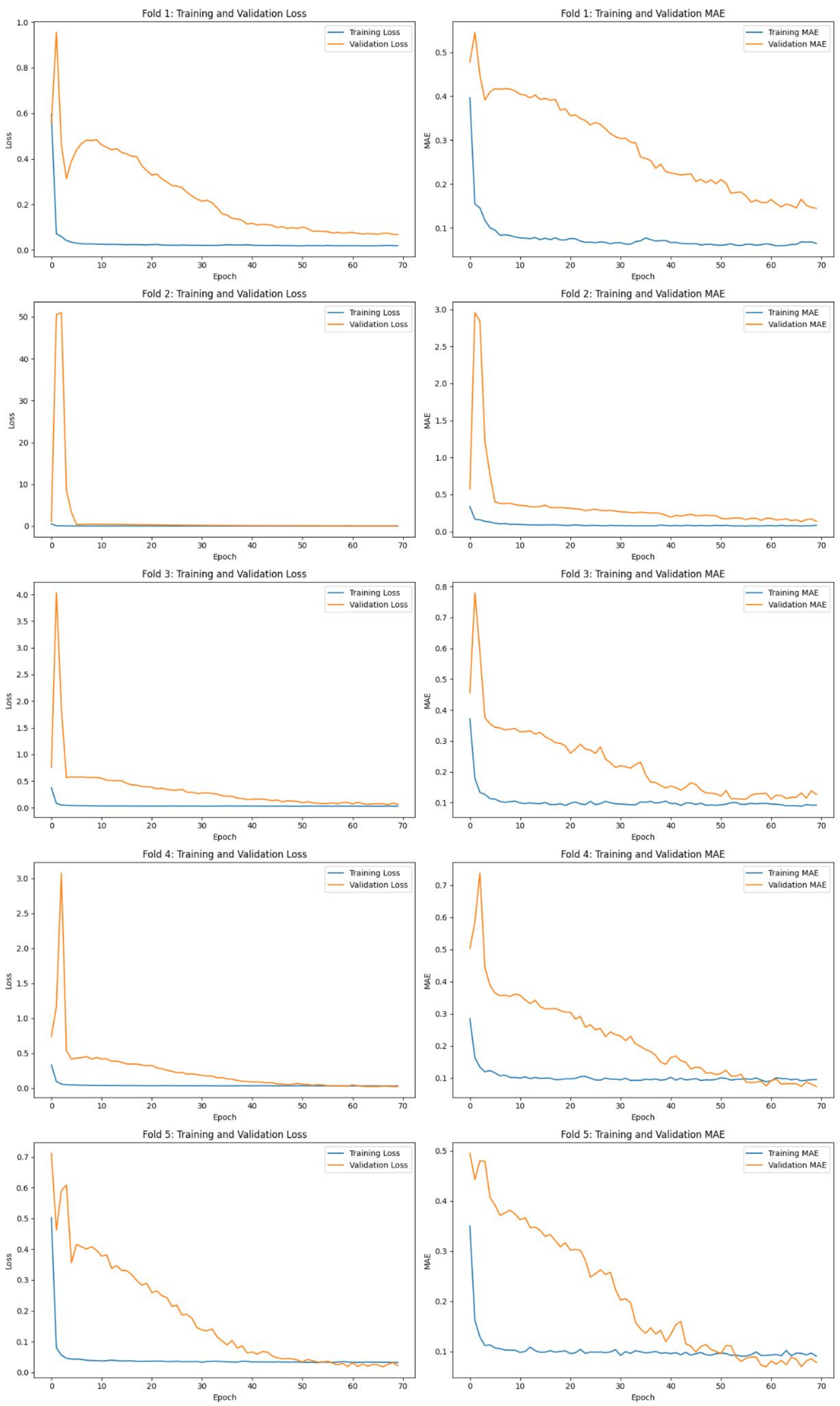
Cross validation & ensemble learning



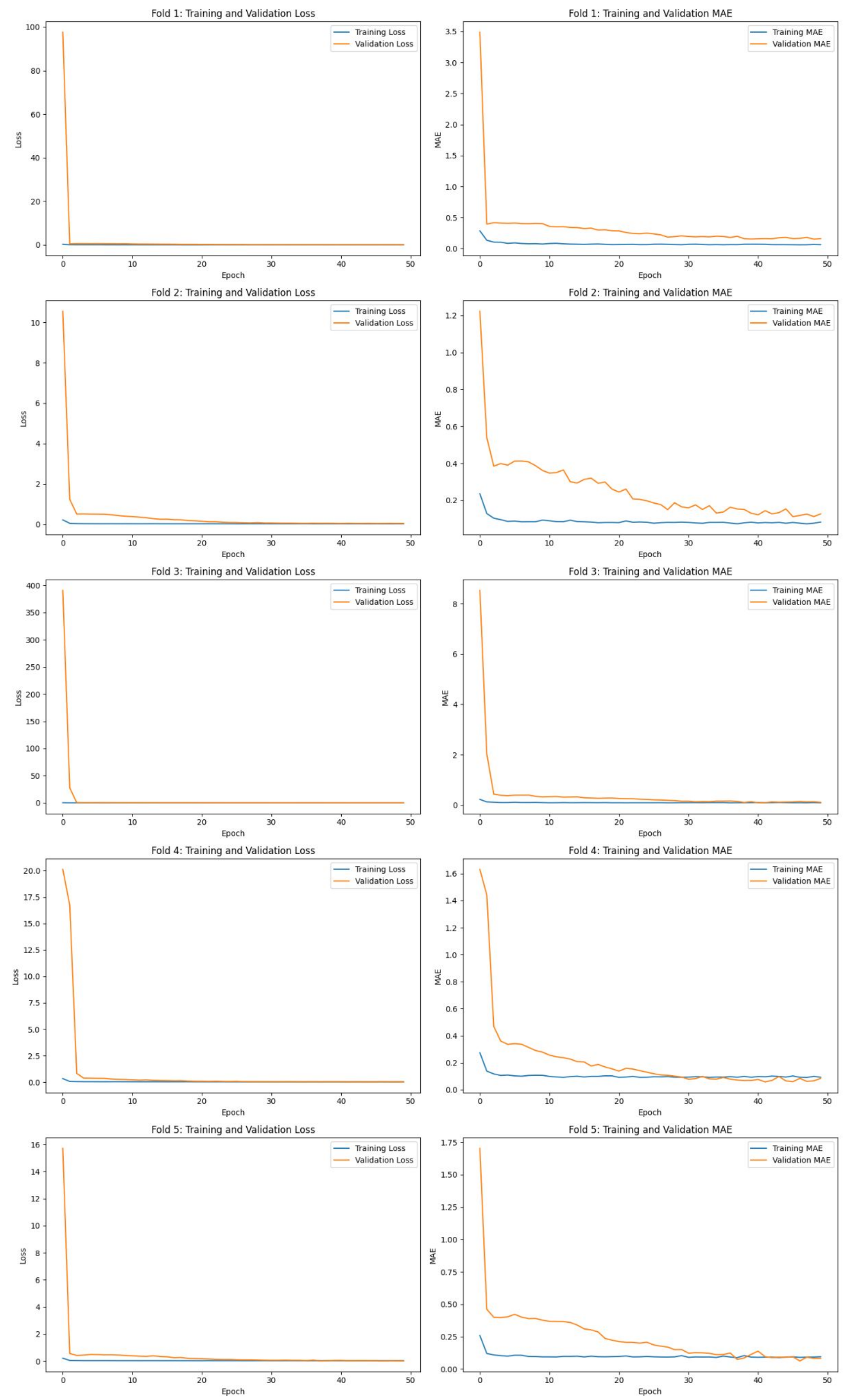
- CV is employed for training a neural network on an image dataset to ensure that the model is **robust** and **generalizes well** to unseen data.
- This technique evaluates the model's performance on different subsets of the dataset, reducing the **risk of overfitting** and providing a more reliable estimate of its ability to predict new images.

Training phase

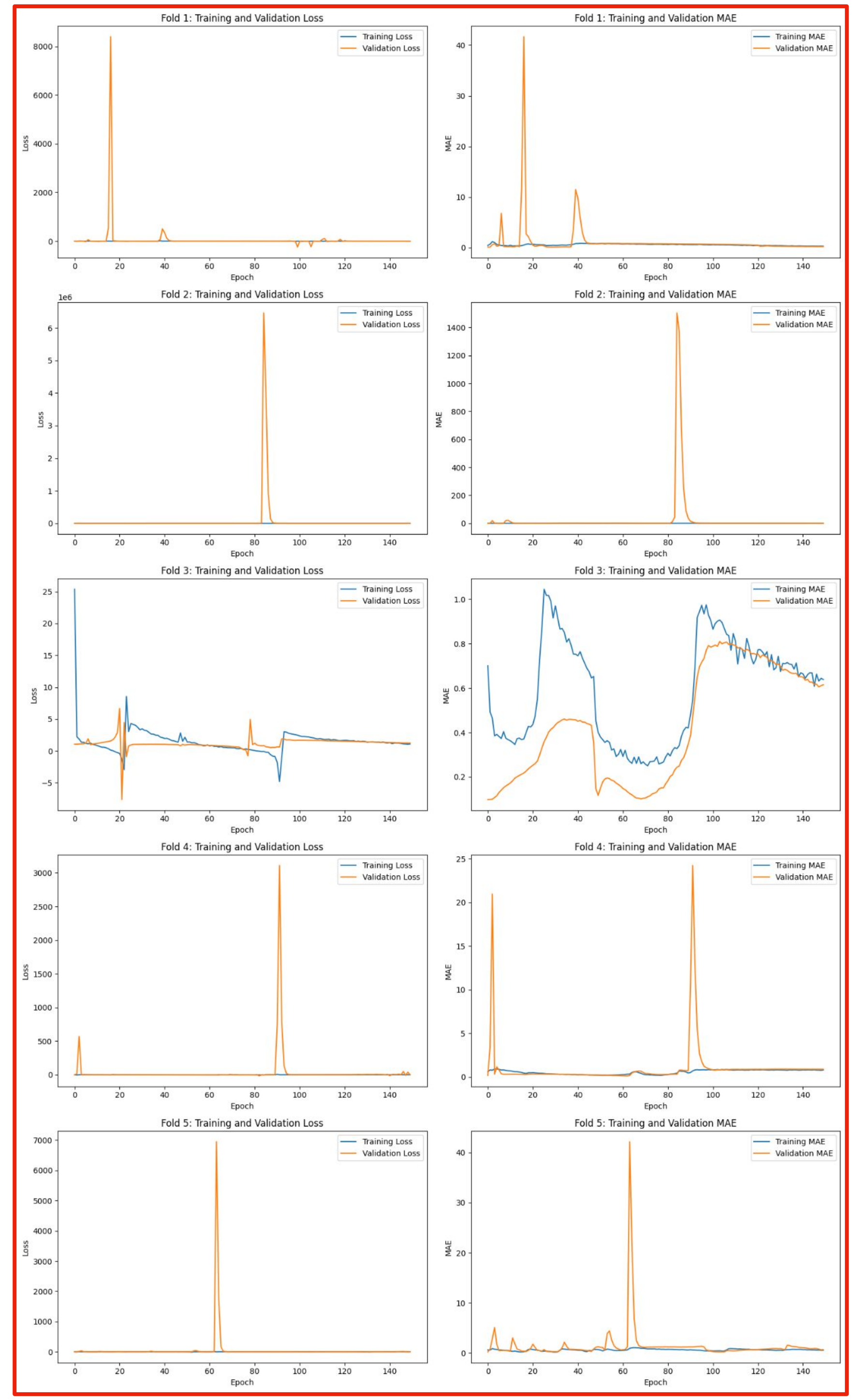
Cross-validation



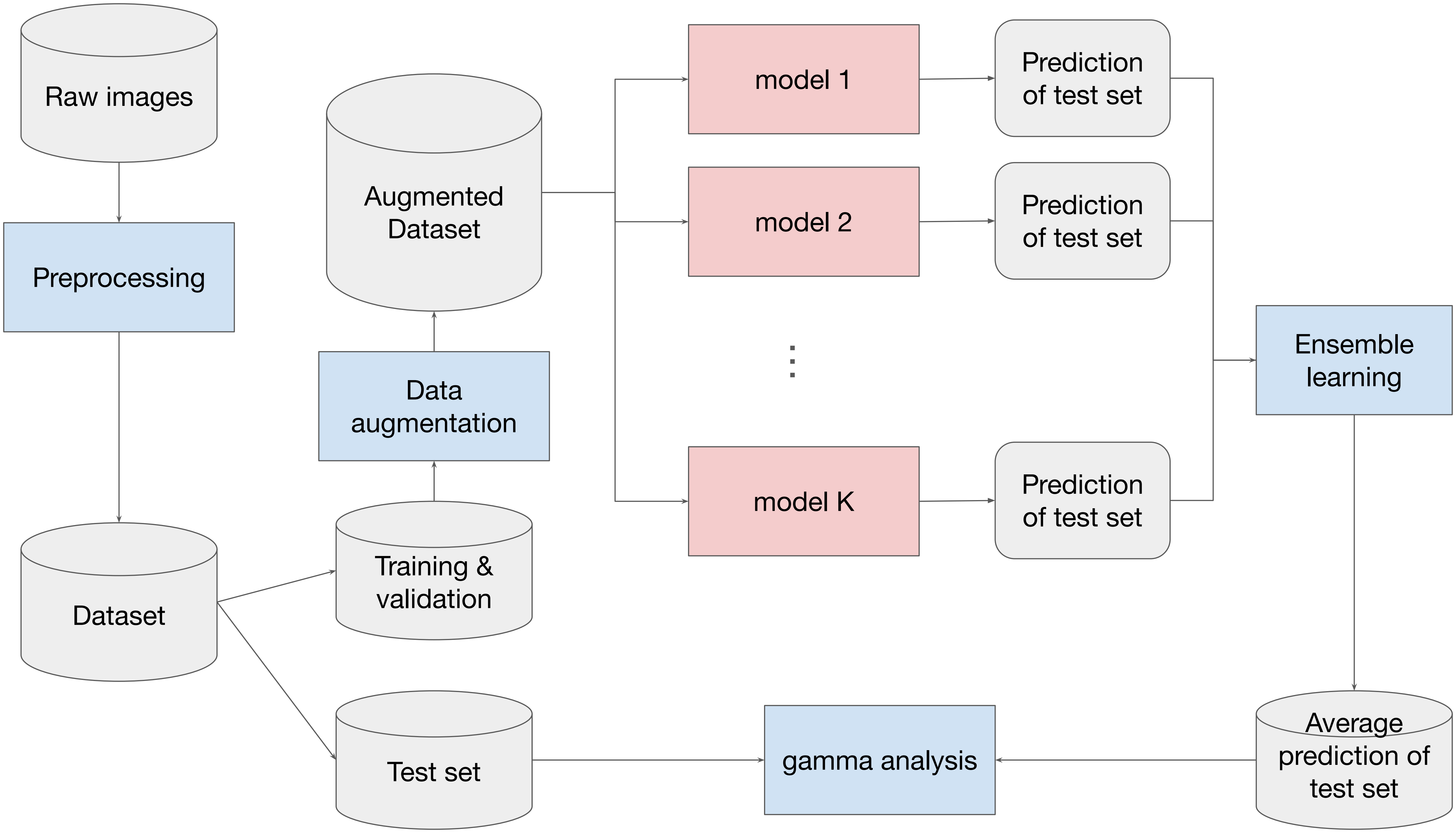
Cross-valid + data aug



Data aug + cross-valid + custom loss

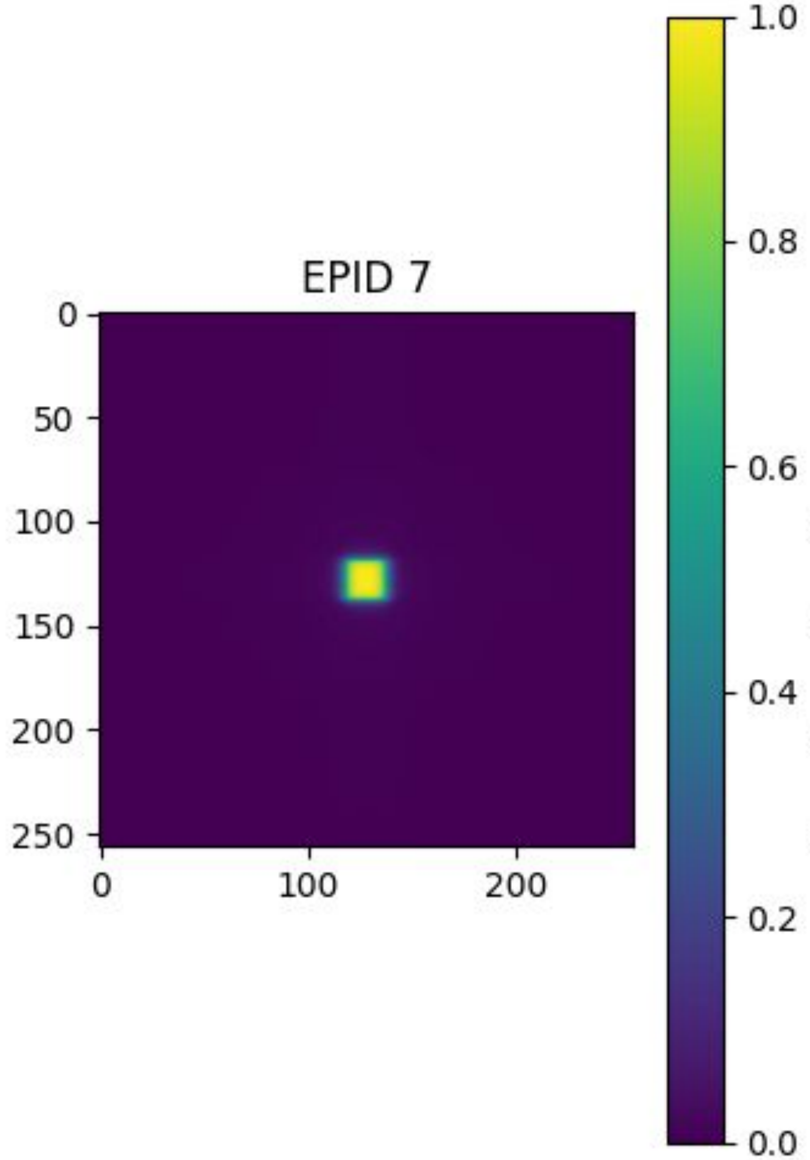


Pipeline of the analysis

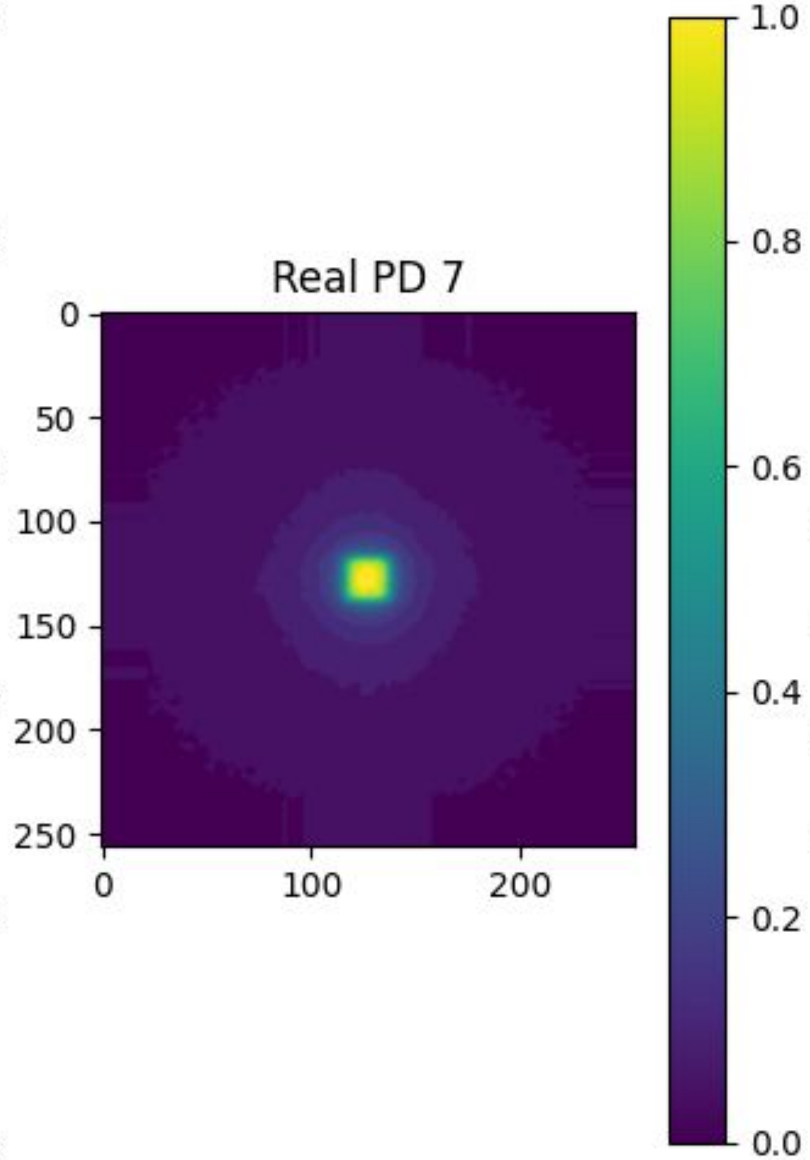


Predictions of the model

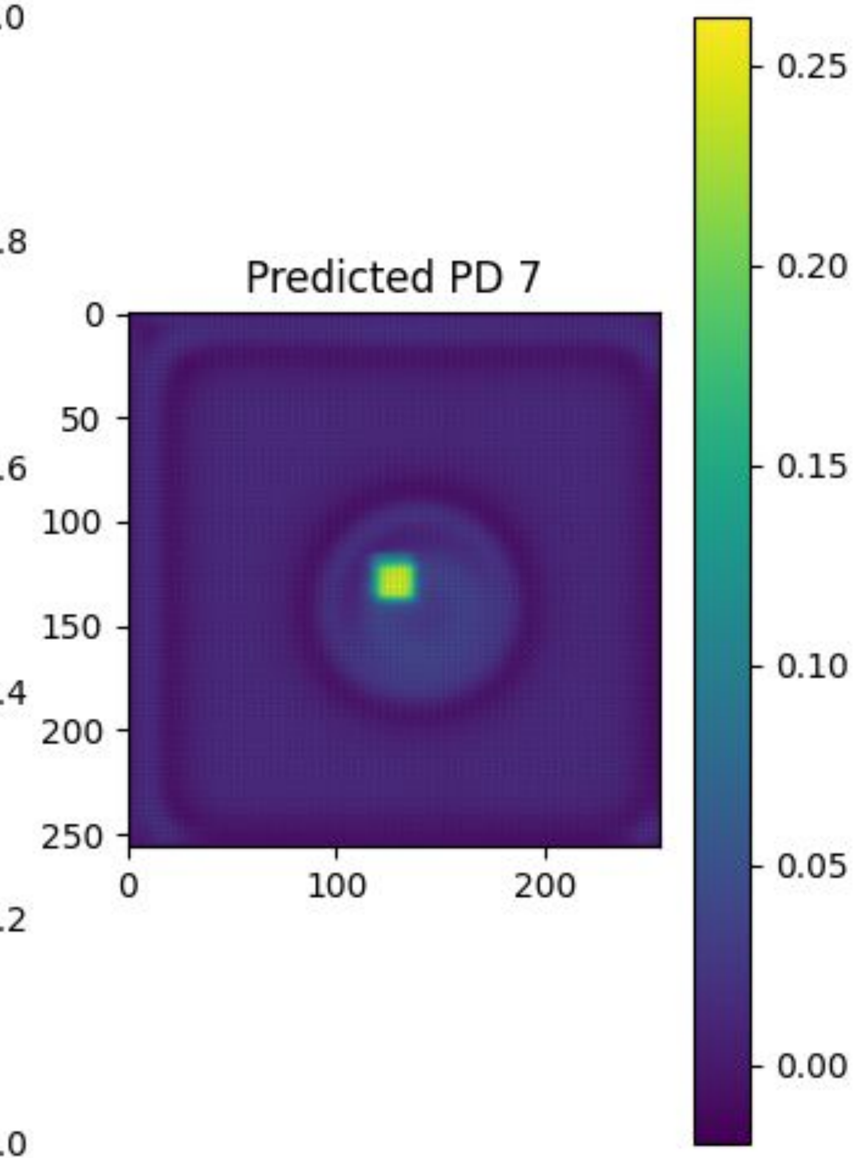
EPID



TPS

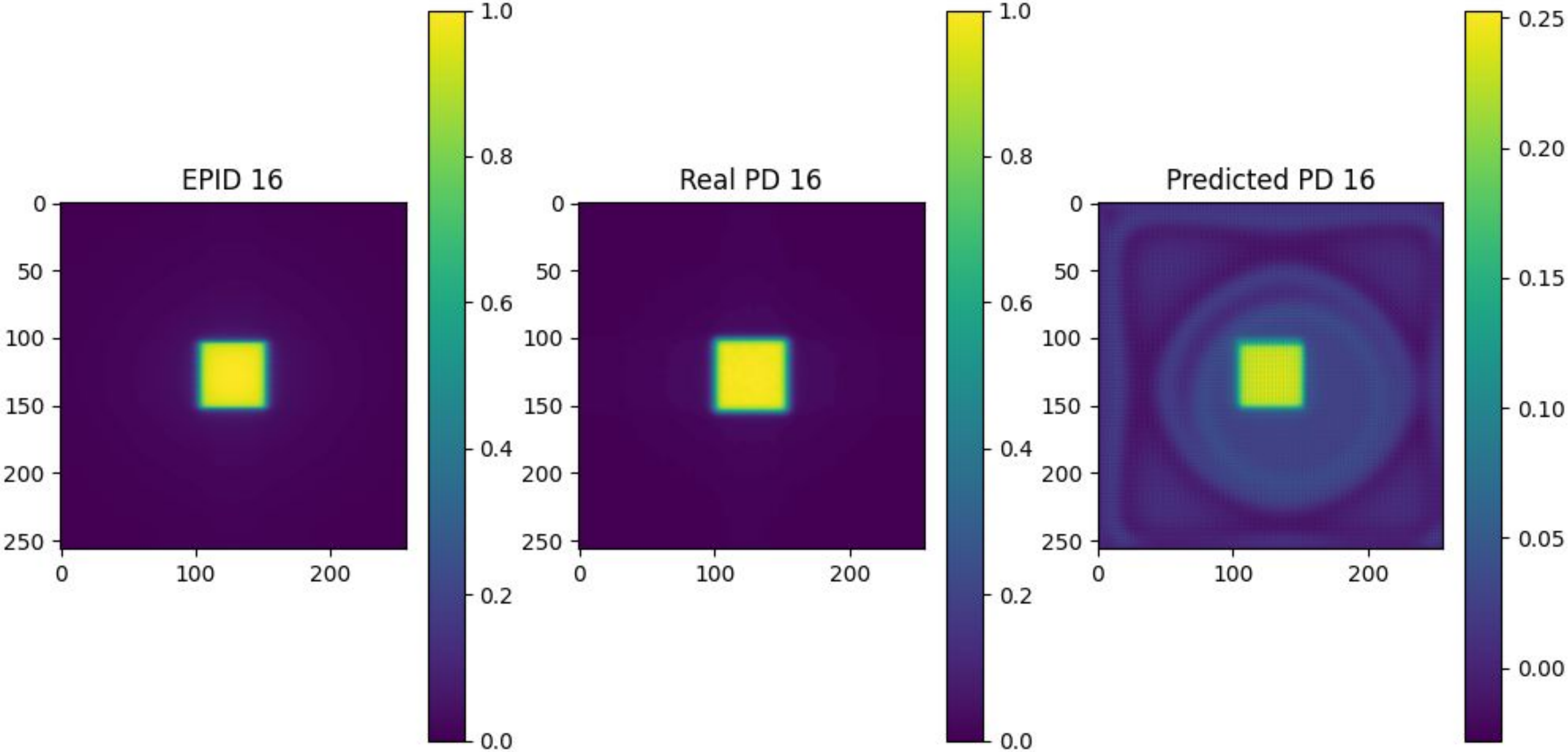


Unet



Predictions of the model

EPID **TPS** **Unet**



Conclusions

In this study, we converted the measured EPID response into the actual PD through a DL network and compared it with the simulated PD calculated by TPS.

A standard **gamma analysis of 3%/3mm** was performed on the PD predicted by the DL network and the simulated PD in the test dataset, leading to a mean gamma pass rates of **(82.30 ± 4.80)%**.

Additional techniques, such as **data augmentation**, k-**cross validation**, and **ensemble learning**, do not produce significantly better results.

- We need to understand in which cases the network performs better and in which ones it performs worse.
- It is possible to use **different types of architectures**.
- **Increase the size** and variety of the dataset

References and contacts

Thank you for the attention!

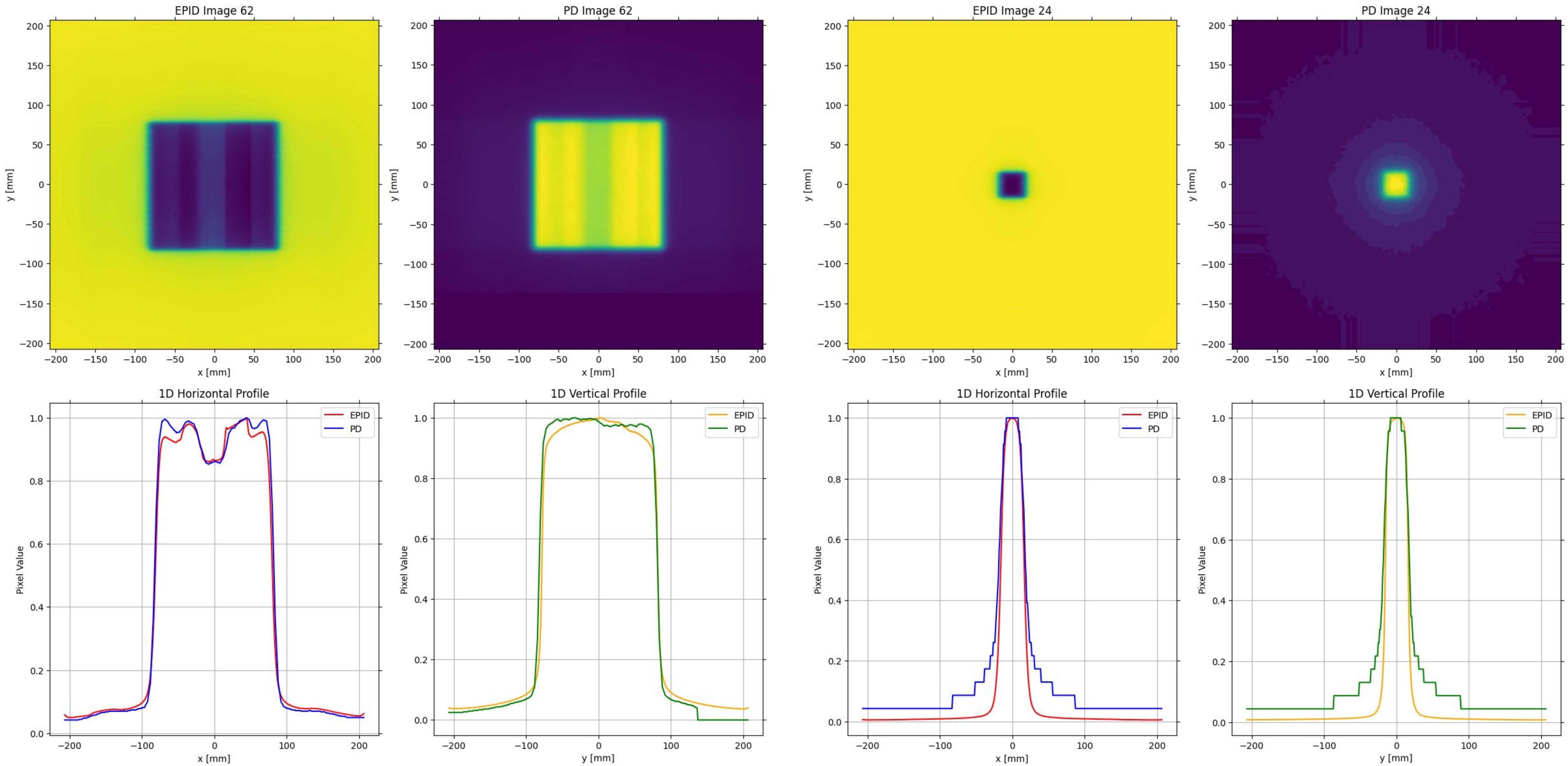
Papers:

1. Fan Z Zhang Q Zhang X Yang R et al. Zhang J, Cheng Z. *A feasibility study for in vivo treatment verification of IMRT using Monte Carlo dose calculation and deep learning-based modelling of EPID detector response*. Radiat Oncol, (17):1–12, 2022.
2. Parodi et. al, *Towards real-time EPID-based 3D in vivo dosimetry for IMRT with Deep Neural Networks: A feasibility study,* Physica Medica, Volume 114, October 2023, 103148

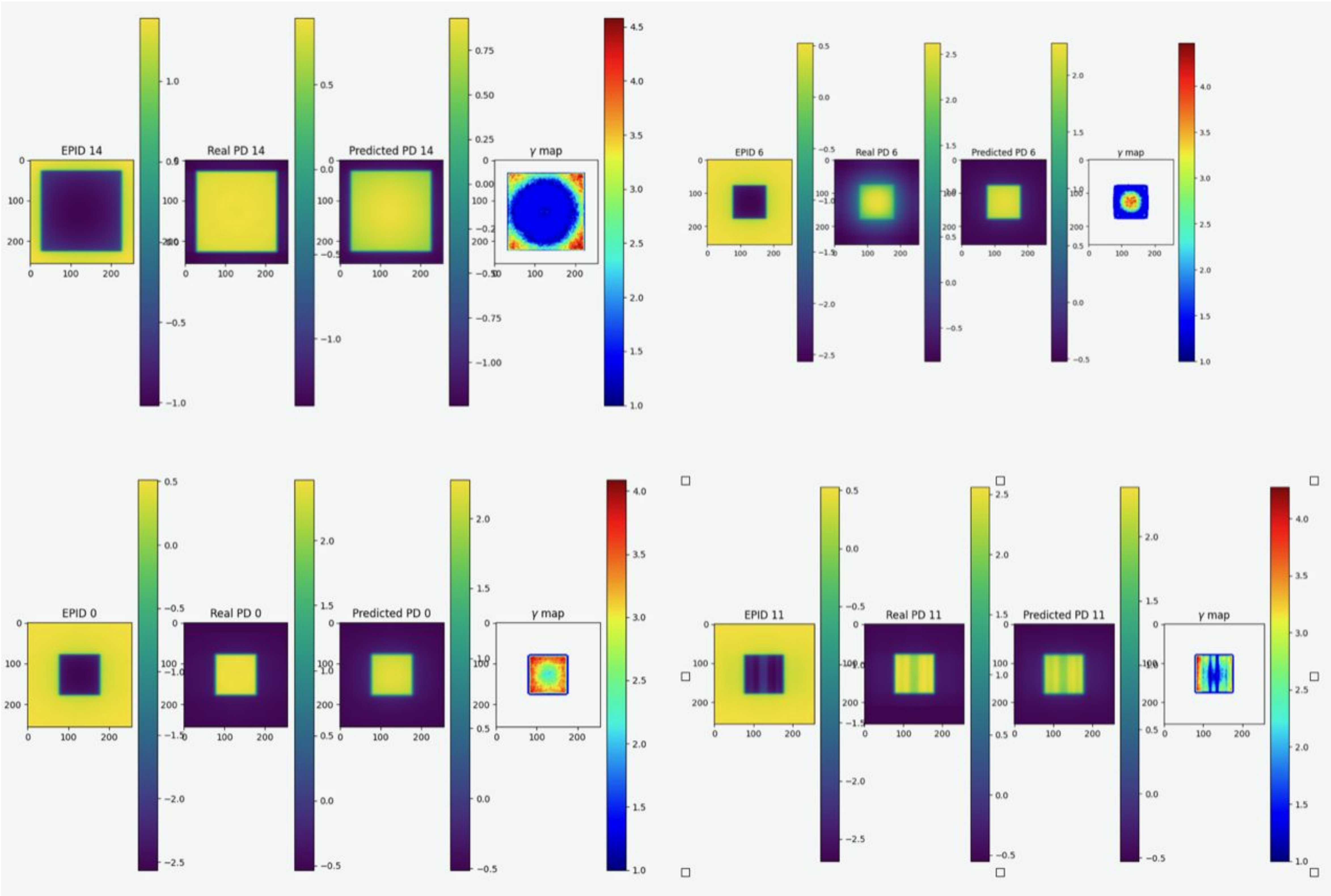
Contacts: lorenzo.marini@phd.unipi.it

Backup slides

Artifacts in the TPS images



Predictions of the model



Predictions of the model

TPS **DL model** **EPID** **TPS** **DL model**