Convolutional Neural Networks for Phase Retrieval in Speckle-based Imaging

Simón González López

Universidad de los Andes | Grupo de Investigación Altas Energías

Contents

Why do we need to use Deep Learning?

MIST [1,2]:

Needs priori information about the sample.

Originally written for monochromatic energy.

Needs at least two [1] or four [2] sample images to retrieve multimodal information.

UMPA [3]:

Iterative minimization of a cost function.

It requires several images (20) to retrieve signals with high quality and spatial resolution.

Why Deep Learning?

Limitations on traditional phase retrieval algorithms: Absorbed dose and priori information is needed.

A model could learn a function to retrieve speckle pattern displacements from one sample image only.

Deep Learning in a Nutshell

- Deep Learning is a subset of Machine Learning.
- Deep Neural Networks learn hierarchical representations of a dataset through multiple levels of abstraction [4].
- Mathematically: Optimization of $\Phi(x;\theta) \rightarrow y$. [5]
- Depending on data availability: Supervised or Unsupervised.

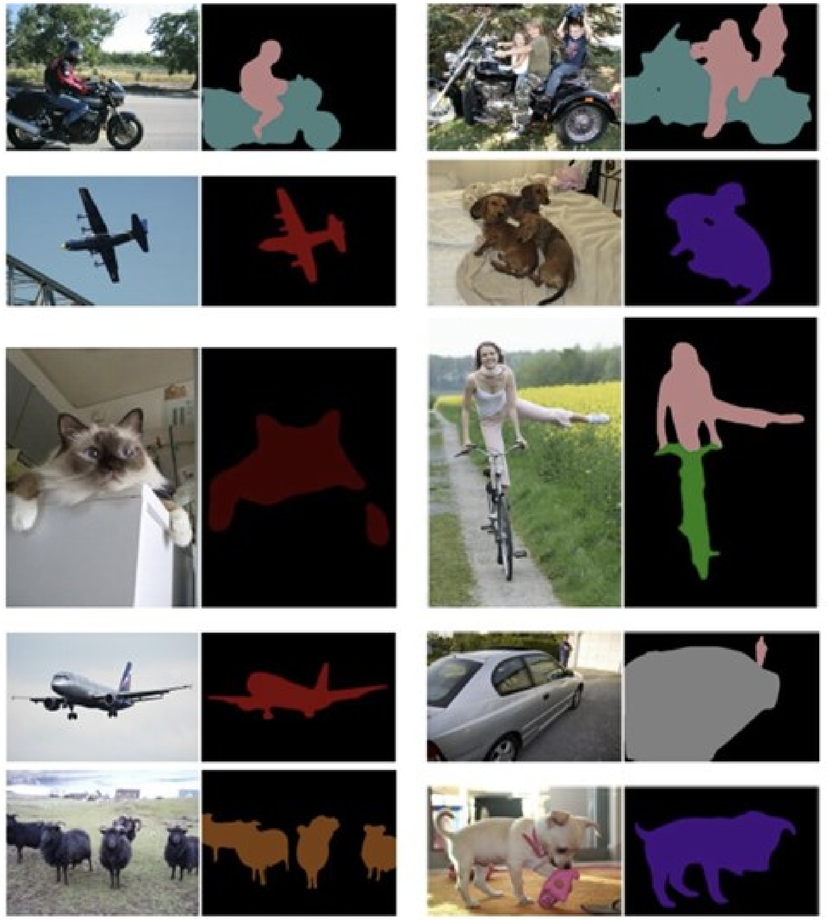
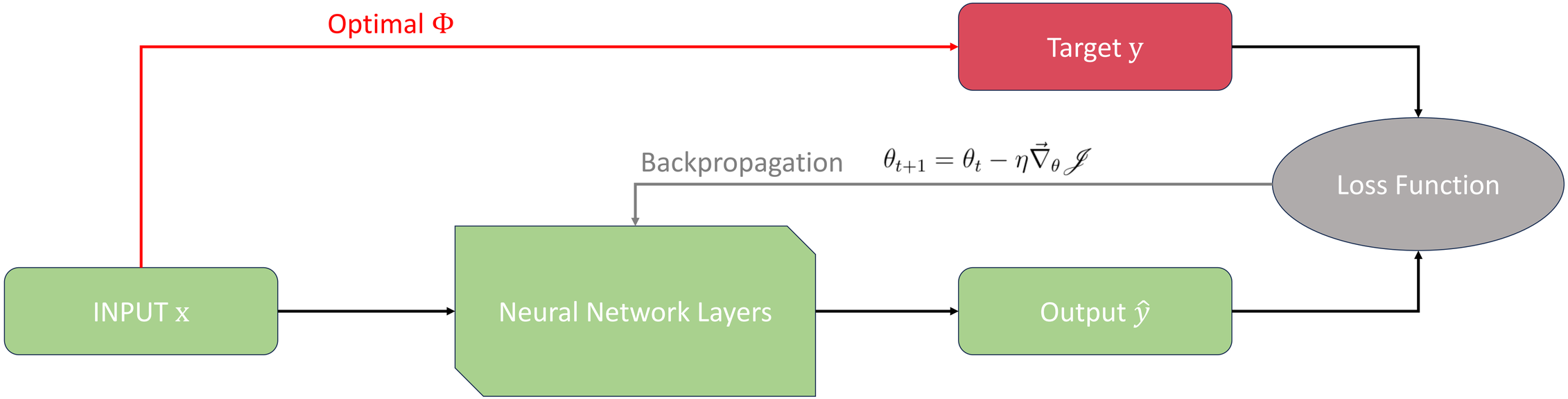
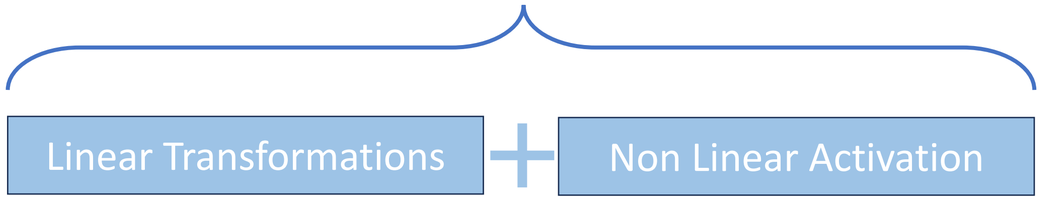


Fig1: Segmentation task performed by Deep Learning models. Taken from [6].

Deep Learning in a Nutshell





$$z = \vec{w}^{\,T}\vec{x} + b$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$\sigma(z) = \max(0, z)$$

$$\sigma(z) = \begin{cases} z & \text{if } z > 0 \\ \alpha z & \text{if } z \leq 0 \end{cases}$$

Convolutional Neural Networks

- Image processing [5].
- Operates on high-dimensional tensors (H x W x C).
- Convolution for extracting hierarchical features [5] $(I * K)(i,j) = \sum\sum I(i-m, j-n)K(m,n)$
- Max Pooling to systematically reduce the dimensionality of input tensor [5].

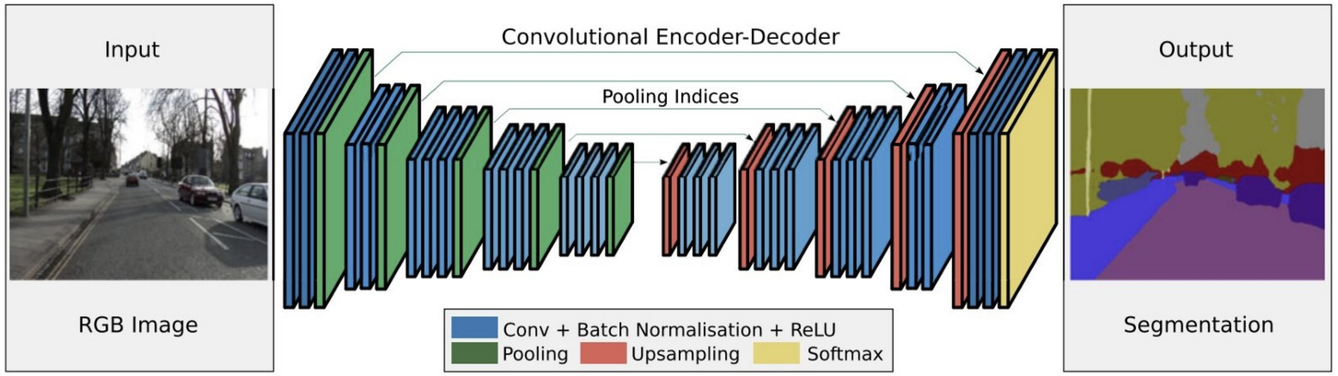


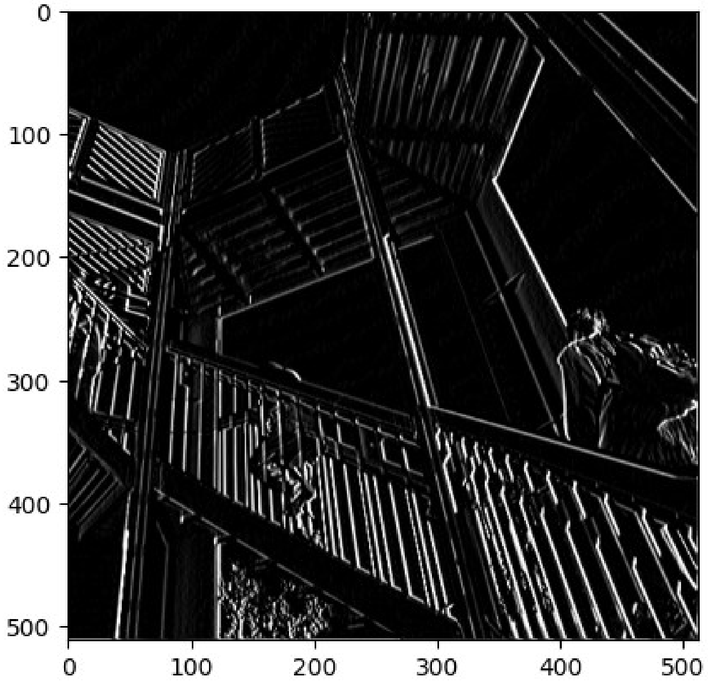
Fig2: Convolutional Neural Network. Taken from [6].

Convolutions: Practical Example

Input Image

$$* \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \longrightarrow$$

Highlight vertical borders

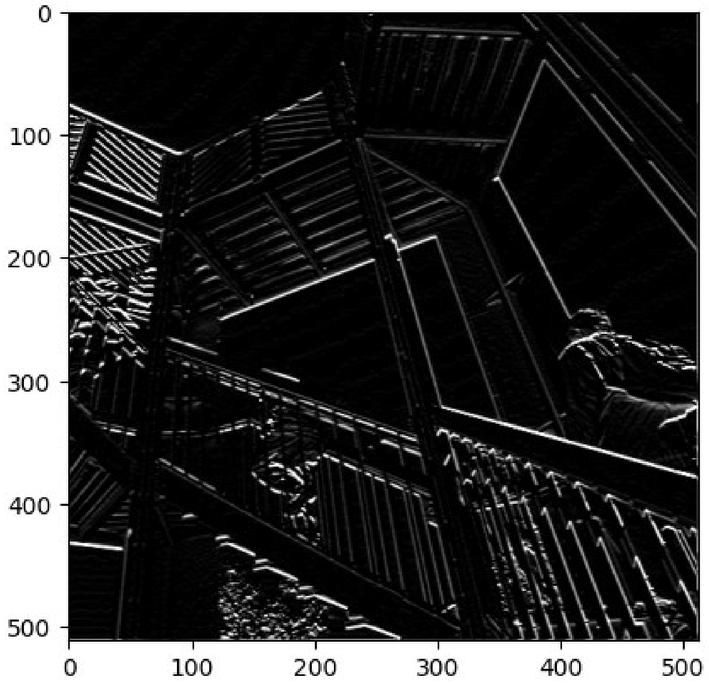


Convolutions: Practical Example

Input Image

$$* \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \longrightarrow$$

Highlight horizontal borders

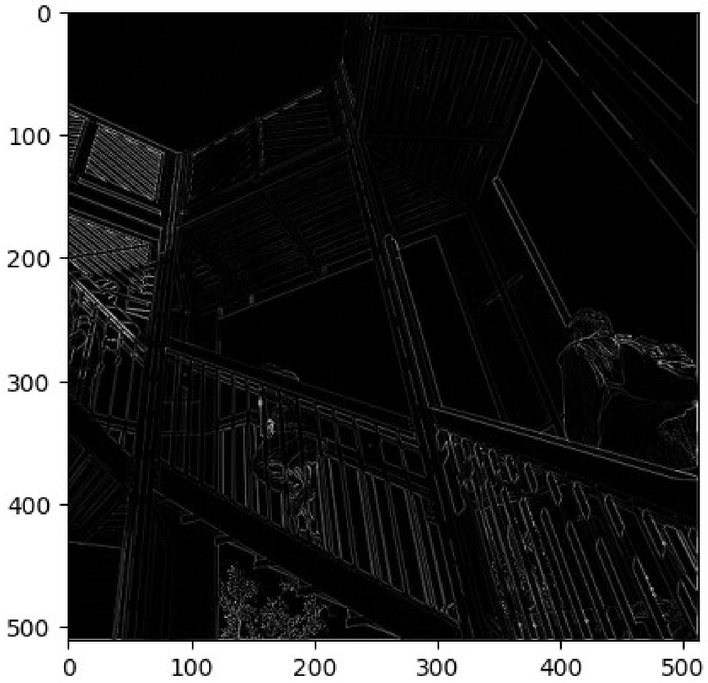


Convolutions: Practical Example

Input Image

$$* \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \longrightarrow$$

Highlight all borders



UNet: Retrieving Speckle Displacements

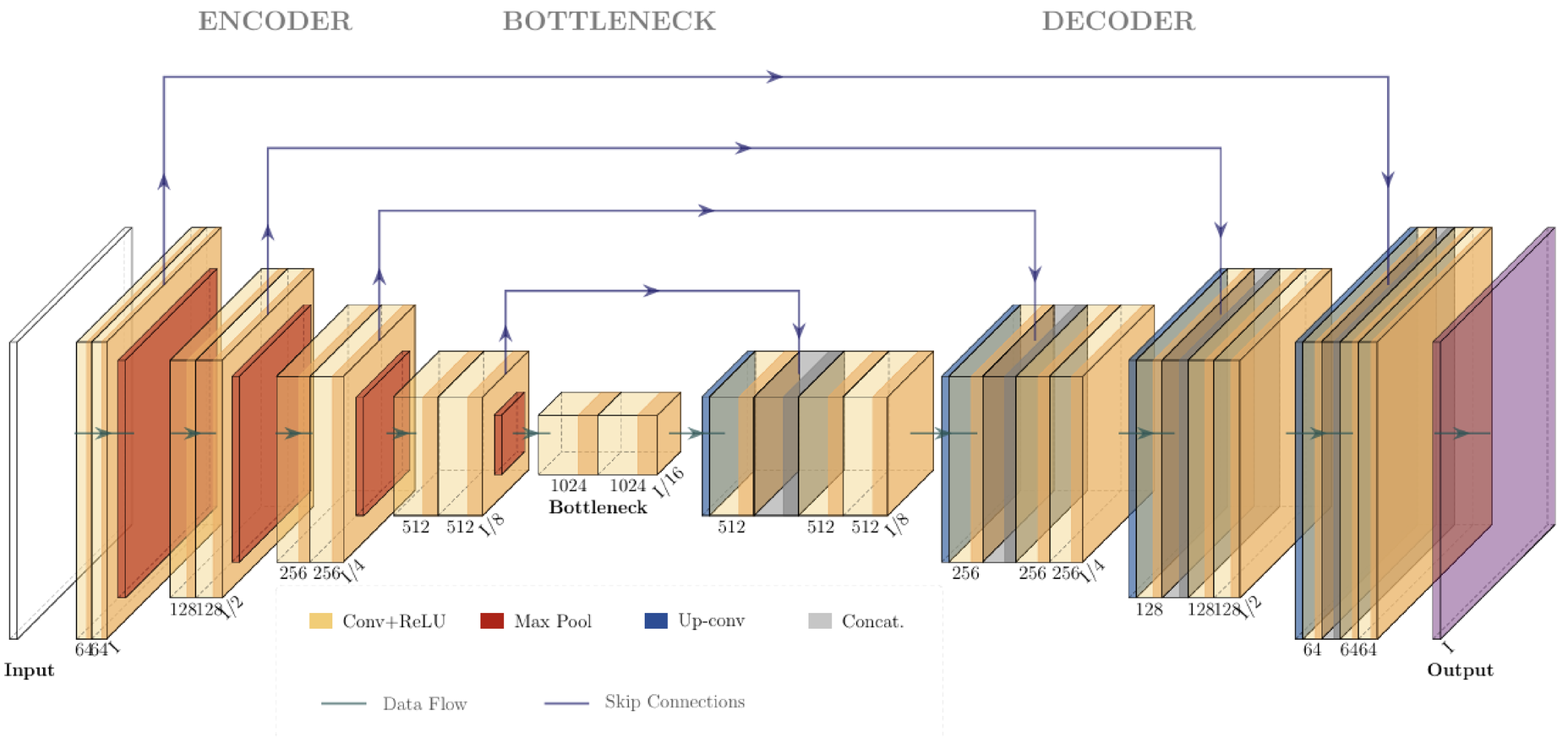
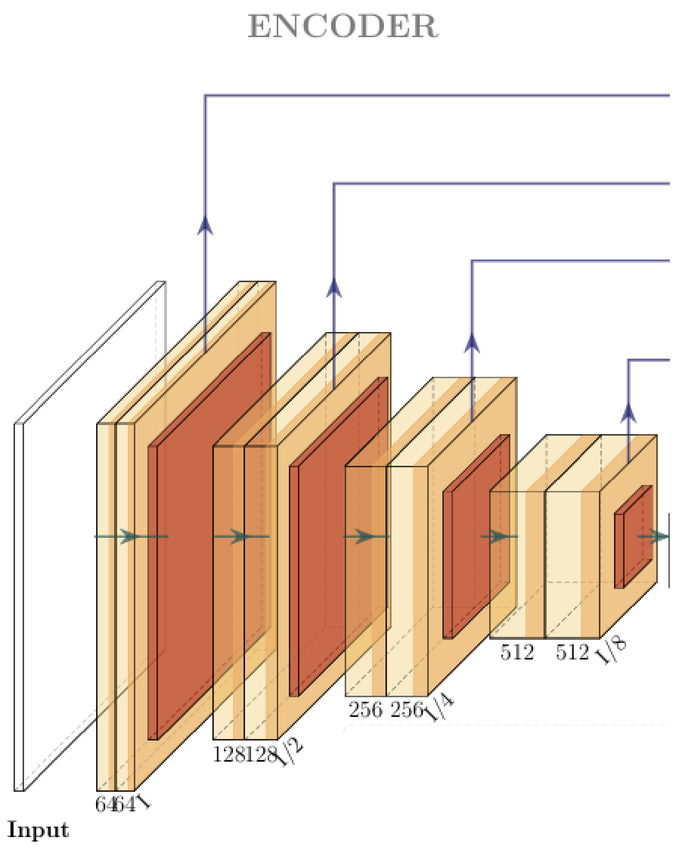


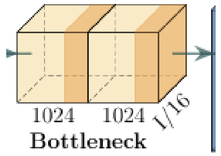
Fig3: UNet Architecture.

UNet: Retrieving Speckle Displacements



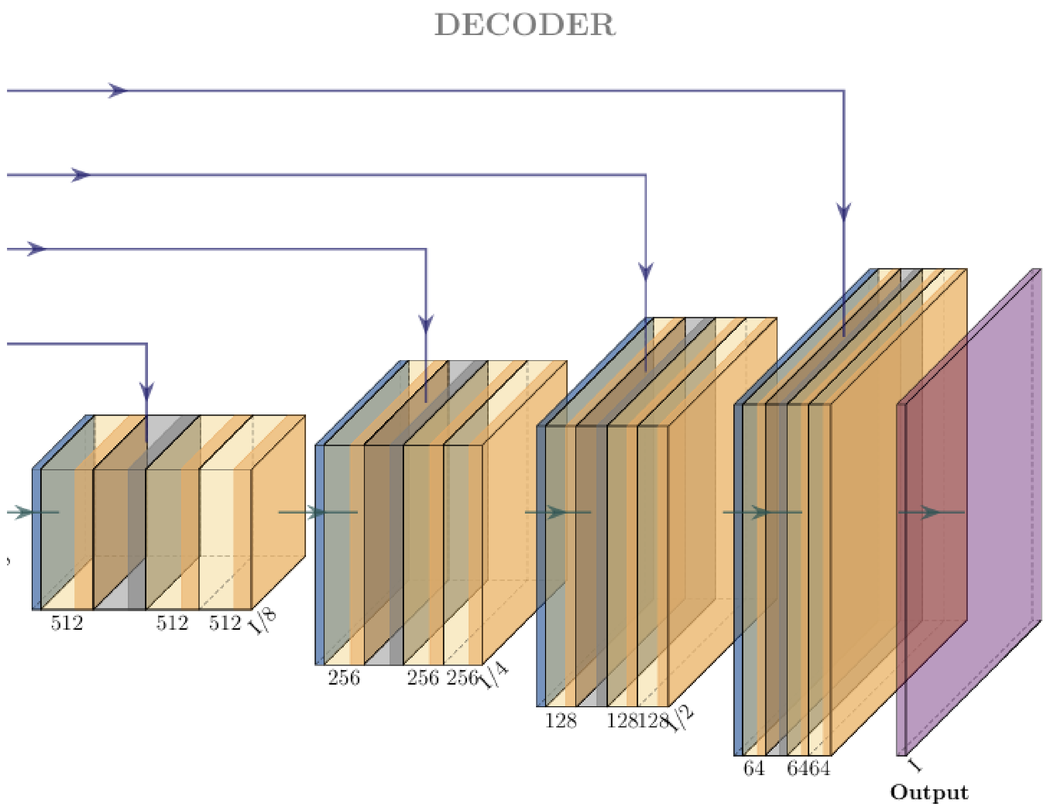

```python
def build_unet(input_shape=(256, 256, 2), num_classes=2):

    inputs = Input(input_shape)

    # Bloque 1 (64 filtros)
    c1 = Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
    c1 = Conv2D(64, (3, 3), activation='relu', padding='same')(c1)
    p1 = MaxPooling2D((2, 2))(c1)

    # Bloque 2 (128 filtros)
    c2 = Conv2D(128, (3, 3), activation='relu', padding='same')(p1)
    c2 = Conv2D(128, (3, 3), activation='relu', padding='same')(c2)
    p2 = MaxPooling2D((2, 2))(c2)

    # Bloque 3 (256 filtros)
    c3 = Conv2D(256, (3, 3), activation='relu', padding='same')(p2)
    c3 = Conv2D(256, (3, 3), activation='relu', padding='same')(c3)
    p3 = MaxPooling2D((2, 2))(c3)

    # Bloque 4 (512 filtros)
    c4 = Conv2D(512, (3, 3), activation='relu', padding='same')(p3)
    c4 = Conv2D(512, (3, 3), activation='relu', padding='same')(c4)
    p4 = MaxPooling2D((2, 2))(c4)
```

UNet: Retrieving Speckle Displacements




```
# Bloque Central (1024 filtros)
c5 = Conv2D(1024, (3, 3), activation='relu', padding='same')(p4)
c5 = Conv2D(1024, (3, 3), activation='relu', padding='same')(c5)
```

UNet: Retrieving Speckle Displacements




```python
# Bloque 4 (Expansión hacia 512 filtros)
u6 = Conv2DTranspose(512, (2, 2), strides=(2, 2), padding='same')(c5)
u6 = concatenate([u6, c4]) # Skip connection con Bloque 4
c6 = Conv2D(512, (3, 3), activation='relu', padding='same')(u6)
c6 = Conv2D(512, (3, 3), activation='relu', padding='same')(c6)

# Bloque 3 (Expansión hacia 256 filtros)
u7 = Conv2DTranspose(256, (2, 2), strides=(2, 2), padding='same')(c6)
u7 = concatenate([u7, c3]) # Skip connection con Bloque 3
c7 = Conv2D(256, (3, 3), activation='relu', padding='same')(u7)
c7 = Conv2D(256, (3, 3), activation='relu', padding='same')(c7)

# Bloque 2 (Expansión hacia 128 filtros)
u8 = Conv2DTranspose(128, (2, 2), strides=(2, 2), padding='same')(c7)
u8 = concatenate([u8, c2]) # Skip connection con Bloque 2
c8 = Conv2D(128, (3, 3), activation='relu', padding='same')(u8)
c8 = Conv2D(128, (3, 3), activation='relu', padding='same')(c8)

# Bloque 1 (Expansión hacia 64 filtros)
u9 = Conv2DTranspose(64, (2, 2), strides=(2, 2), padding='same')(c8)
u9 = concatenate([u9, c1]) # Skip connection con Bloque 1
c9 = Conv2D(64, (3, 3), activation='relu', padding='same')(u9)
c9 = Conv2D(64, (3, 3), activation='relu', padding='same')(c9)

outputs = Conv2D(num_classes, (1, 1), activation='linear')(c9)

model = Model(inputs=[inputs], outputs=[outputs])
return model
```

Training Dataset

$$\vec{d}_{\perp}(x,y) = \frac{z_d}{k}\vec{\nabla}_{\perp}\phi(x,y)$$

$$\phi(x,y,z) = -k\int_{z=0}^{z=z_0}\delta_{\omega}(x,y,z)dz$$

$$\phi(x,y,z) = -k\delta_{\omega}T(x,y)$$

- 600 images.
- 150 images per geometry.
- Three materials: Al_2O_3, PMMA, Nylon.
- 28keV, 4 sandpapers SiC 125um, source-detector distance 2m.

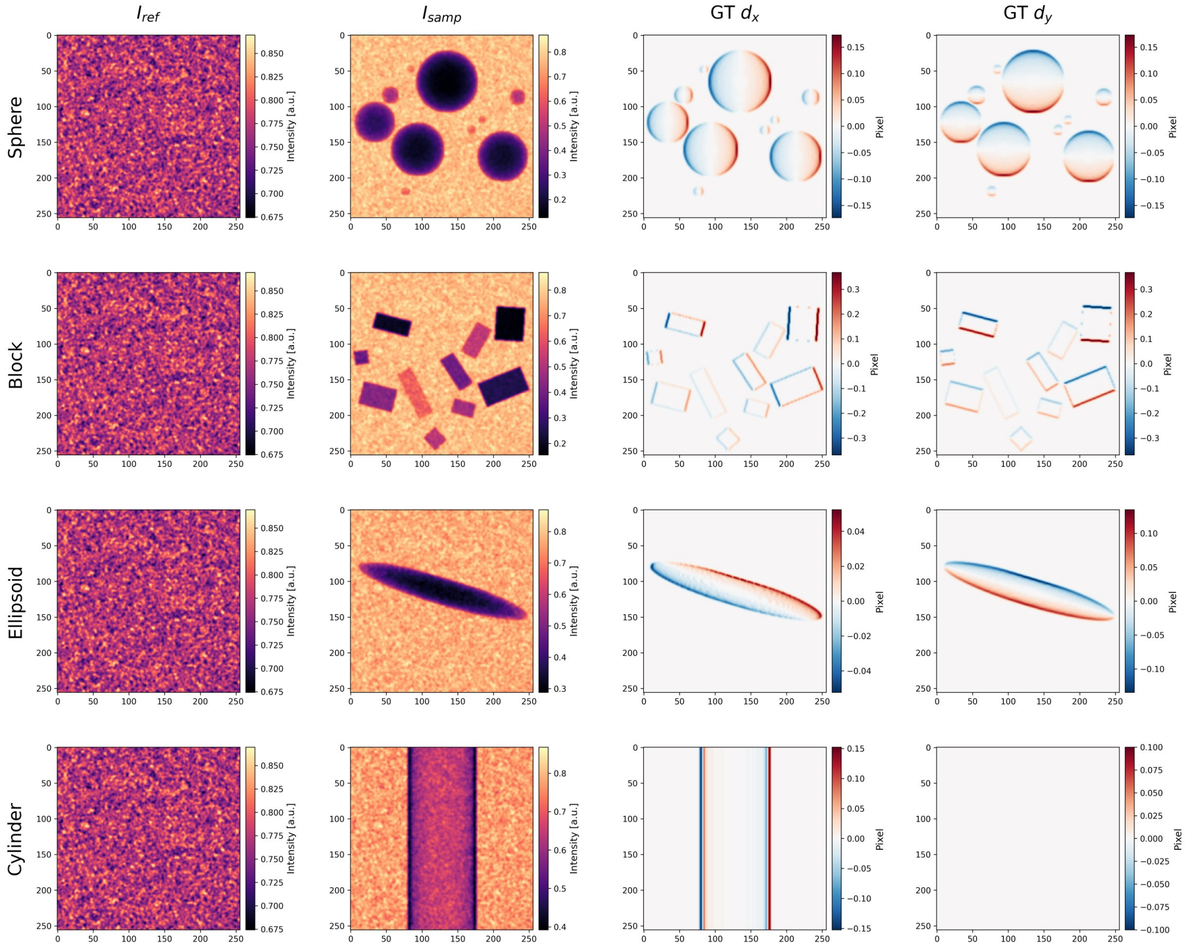


Fig4: Training dataset.

Validation Dataset

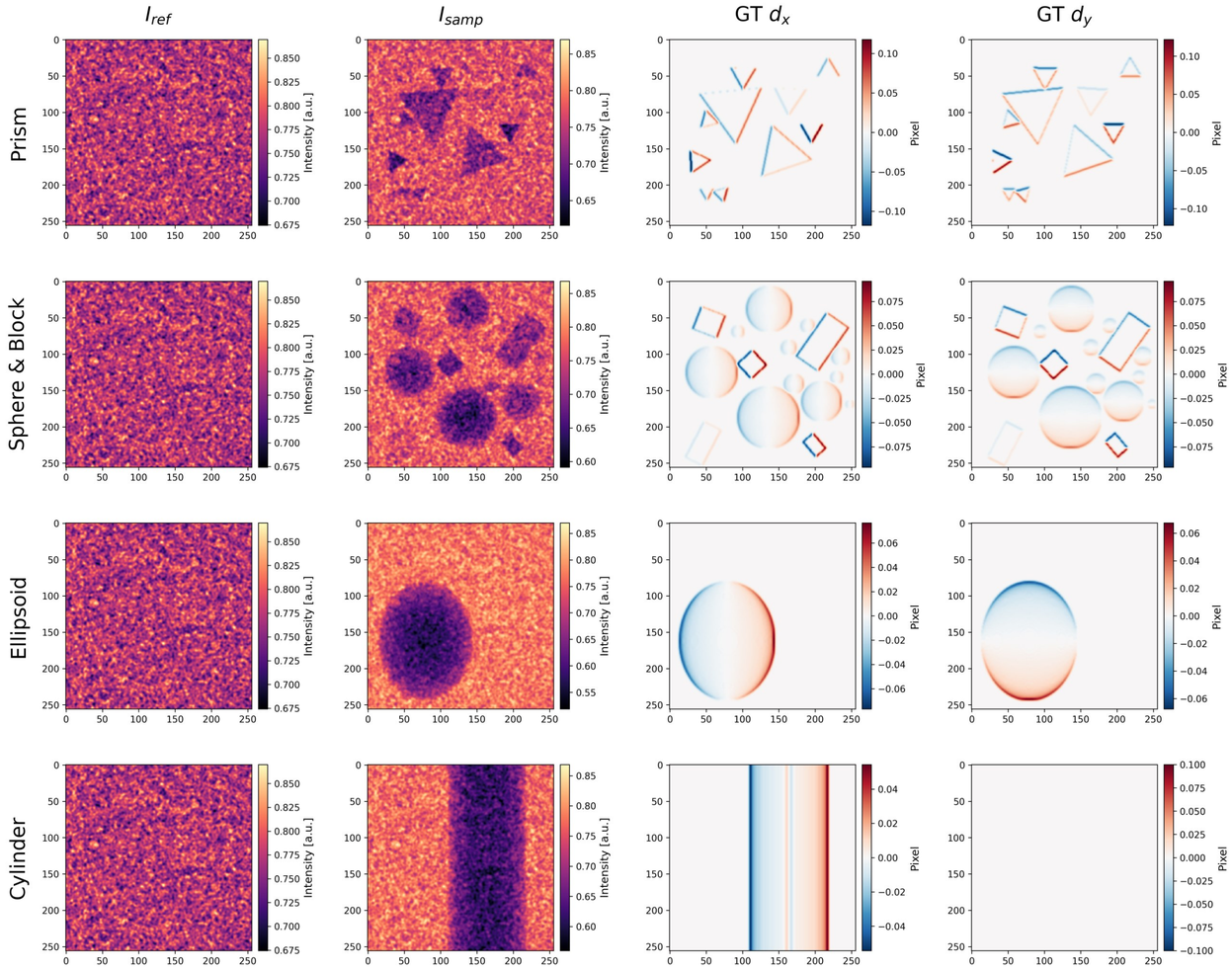


Fig5: Validation dataset.

Evaluation Dataset

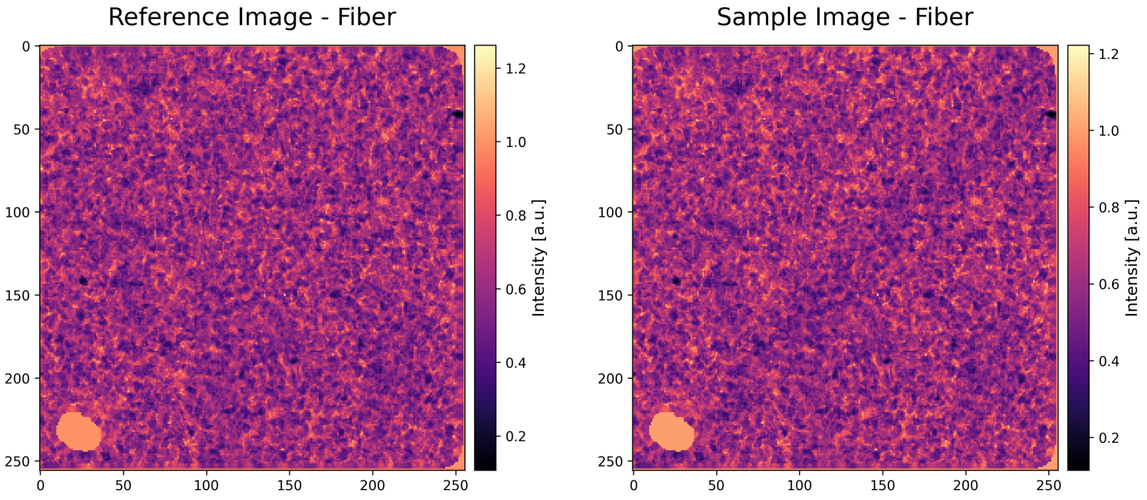


Fig6: Evaluation: Nylon fiber.

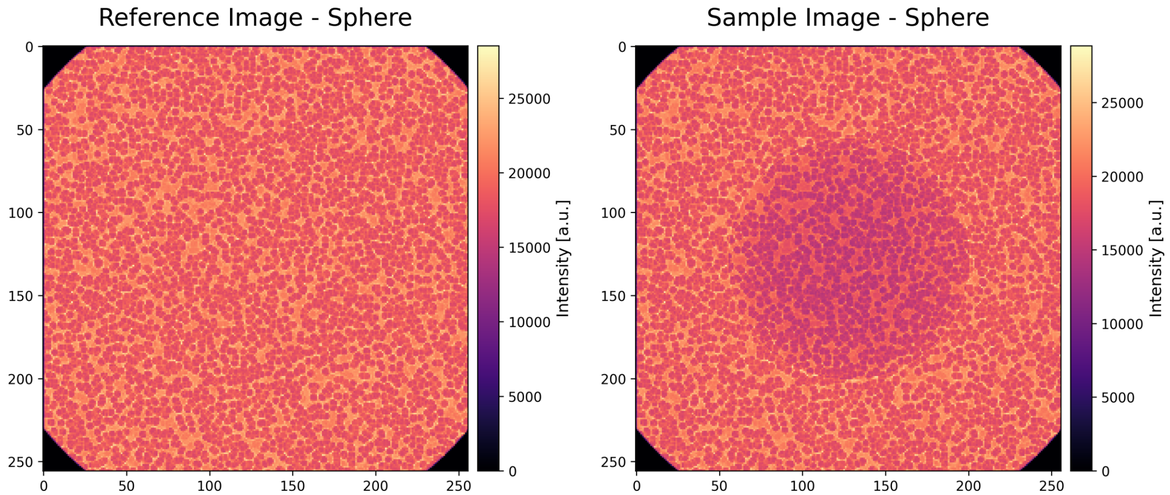


Fig7: Evaluation: Nylon sphere.

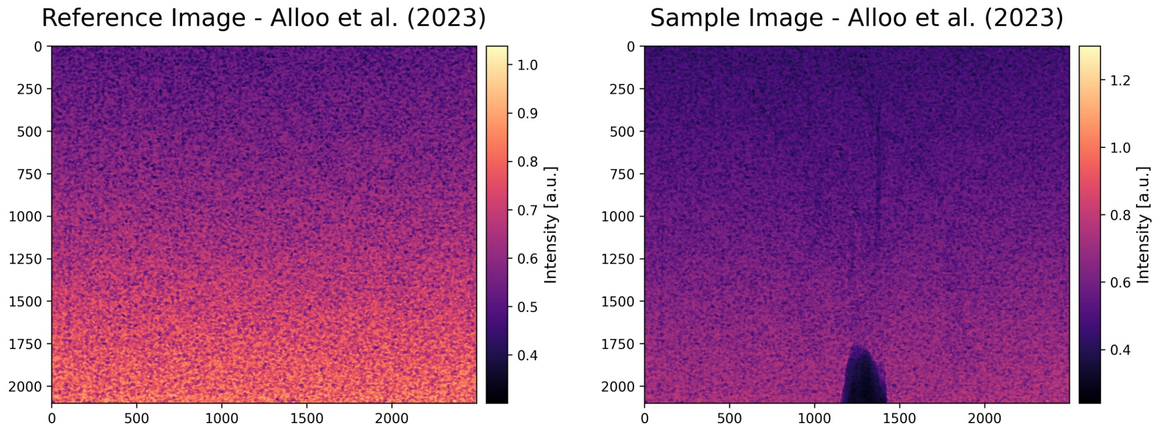


Fig8: Evaluation: Wattle flower [2].

Link to Google Drive

- Link: https://drive.google.com/drive/folders/1TFuEhGH6aF2DhmRQAdSwJm6O6c6zZ8FN?usp=sharing

References

1. S. J. Alloo, D. M. Paganin, K. S. Morgan, M. J. Kitchen, A. W. Stevenson, S. C. Mayo, H. T. Li, B. M. Kennedy, A. Maksimenko, J. C. Bowden, et al., “Dark-field tomography of an attenuating object using intrinsic x-ray speckle tracking,” Journal of Medical Imaging, vol. 9, no. 3, pp. 031502–031502, 2022.
2. S. J. Alloo, K. S. Morgan, D. M. Paganin, and K. M. Pavlov, “Multimodal intrinsic speckle-tracking (mist) to extract images of rapidly-varying diffuse x-ray dark-field,” Scientific Reports, vol. 13, no. 1, p. 5424, 2023.
3. I. Zanette, T. Zhou, A. Burvall, U. Lundström, D. H. Larsson, M. Zdora, P. Thibault, F. Pfeiffer, and H. M. Hertz, “Speckle-based x-ray phase-contrast and dark-field imaging with a laboratory source,” Physical review letters, vol. 112, no. 25, p. 253903, 2014.
4. Y.LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436 444, 2015.
5. I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning, vol. 1. MIT press Cambridge, 2016
6. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence, 44(7), 3523-3542. 2021.