

Assignment no. 1

Scenario: Participating in a ML challenge

Now you will implement your NN using PyTorch and train/tune it to compete in the project challenge. First you will write a function that creates the input and target arrays used in the challenge. Then you will implement/train/tune your NN to process an input image and predict a target array for the project challenge.

Exercise 4 [15 points]

In this exercise you should create a function `ex4(image_array, crop_size, crop_center)` that creates two input arrays and one target array from one input image. For this, your function should crop out (=set values to zero) a part of the image, which will then become the target. Since it could be valuable information for our network to know which part was cropped out and should be restored, we will also prepare an additional input channel that includes information about which pixels are in or outside the cropped-out rectangle.

In detail, your function should take the following keyword arguments:

- `image_array`: A numpy array containing the image data in an arbitrary datatype and shape (X, Y).
- `crop_size`: A tuple containing 2 odd int values. These two values specify the size of the rectangle that should be cropped-out in pixels for the two spatial dimensions X and Y.
- `crop_center`: A tuple containing 2 int values. These two values are the position of the center of the to-be cropped-out rectangle in pixels for the two spatial dimensions X and Y.

Your function should return a tuple (`image_array`, `crop_array`, `target_array`), where the returned `image_array` is a modified version of the original `image_array` that the function gets as argument.

`image_array` should be modified such that the pixels in the cropped-out rectangle are set to 0, while the rest of the pixels remains unchanged. You may edit the original `image_array` in-place or create a copy.

`crop_array` should be a numpy array of same shape and datatype as `image_array`, containing value 0 for pixels located outside the cropped-out rectangle and 1 for pixels located in the cropped-out rectangle.

`target_array` should be a 2D numpy array of the same datatype as `image_array`, containing the values of the original `image_array` in the cropped-out rectangle.

The to-be cropped-out rectangle is specified via the center of the rectangle `crop_center` and the size of the rectangle in pixels `crop_size`. Theoretically, we could rotate the rectangle or choose other forms to crop out but we will not consider these cases here.

Your function should raise a `ValueError` exception if

- `image_array` is not a 2D numpy array (see hints on how to check if an object is a numpy array instance).

- `crop_size` or `crop_center` do not contain exactly 2 objects. (You do not need to check the datatype of the objects, you can assume them to be integers.)
- The values in `crop_size` are even numbers.
- The minimal distance between the to-be cropped-out rectangle and the border of `image_array` is less than 20 pixels.

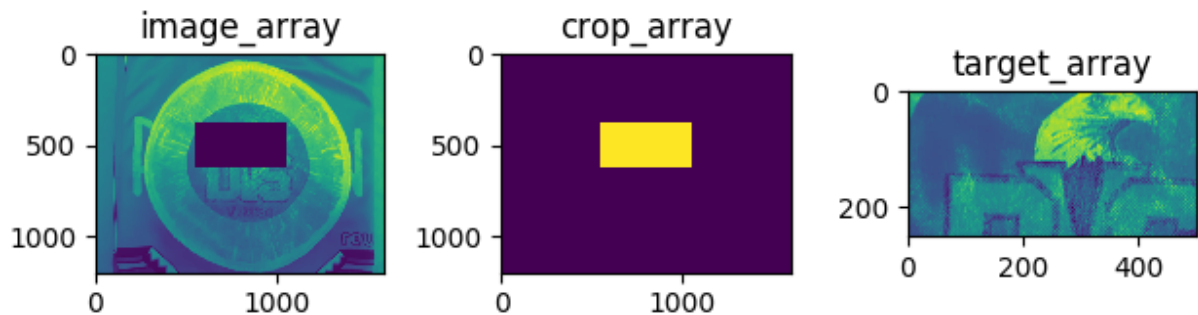


Fig.1: Example for arrays returned by function `ex4()` (plotted using `matplotlib`) with `crop_size=(251, 501)` and `crop_center=(500, 800)`.

Hint: To check whether an object is a certain instance, you can use the `isinstance()` function. Numpy arrays are instances of `np.ndarray`.

Hint: To create `crop_array` with same shape and datatype as `image_array`, you can use the `np.zeros_like()` function. To create the `target_array` you can use slicing to obtain the values from `image_array` (but don't forget to use `np.copy(target_array)` if you modify the `image_array` values in-place afterwards).

Hint: For feeding the input into a NN in exercise 5, you could concatenate the channels of `image_array` and `crop_array`, resulting in an input array of shape `(2, X, Y)` or `(X, Y, 2)`.

Exercise 5 [36 points + 10 bonus points]

Project submission (points determined by model performance). Details tba.

Exercise 6 [10 bonus points]

Bonus exercise. Details tba.

Submission: electronically via Moodle:

`https://moodle.jku.at/`

Deadline: For deadlines see individual Moodle exercises.

Follow the **instructions for submitting homework** stated on the Moodle page!

Copyright statement:

This material, no matter whether in printed or electronic form, may be used for personal and non-commercial educational use only. Any reproduction of this material, no matter whether as a whole or in parts, no matter whether in printed or in electronic form, requires explicit prior acceptance of the authors.