UNIVERSITÀ
DEGLI STUDI
FIRENZE

Parallel Programming
Lab assignment

Lucia Giorgi

06/03/25

UNIVERSITÀ DEGLI STUDI FIRENZE

Index

UNIVERSITÀ DEGLI STUDI FIRENZE

Image augmentation

Perform Image augmentation using the library Albumentations

- N input images
- M augmentation for each image

UNIVERSITÀ DEGLI STUDI FIRENZE

Image augmentation

UNIVERSITÀ DEGLI STUDI FIRENZE

Implementation

Inputs:

- input folder (with N images)
- output folder
- number of augmentations (M)
- number of parallel processes (parallel versions only)
- output result file

Random augmentations are performed by randomly choosing and combining transformations

UNIVERSITÀ DEGLI STUDI FIRENZE

Implementation

Two loops:

- Images in the input folder (N)
- Augmentations (M for each image)

$\Rightarrow$ Two possible parallelizations

Python multiprocessing

Parallelization

Parallelization on the augmentations

```
def augment_images(input_dir, output_dir, num_augmented, num_processes, result_file):
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
    start_time = time.time()
    with multiprocessing.Pool(processes=num_processes) as  pool:
        for filename in os.listdir(input_dir):
            input_path = os.path.join(input_dir, filename)
            image = cv2.imread(input_path)
            image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
            pool.starmap(apply_transformation,
                [(image, output_dir, filename, i) for i in range(num_augmented)])
    end_time = time.time()
    write_output_to_file(end_time, num_processes, num_augmented, result_file, start_time)
```

Parallelization

Parallelization on the images

```
def process_image(filename, input_dir, output_dir, num_augmented):
    input_path = os.path.join(input_dir, filename)
    image = cv2.imread(input_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    for i in range(num_augmented):
        apply_transformation(image, output_dir, filename, i)

def augment_images(input_dir, output_dir, num_augmented, num_processes, result_file):
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
    start_time = time.time()
     with multiprocessing.Pool(processes=num_processes) as  pool:
        pool.starmap(process_image,
            [(filename, input_dir, output_dir, num_augmented) for filename in os.listdir(input_dir)])
    end_time = time.time()
```

UNIVERSITÀ DEGLI STUDI FIRENZE

Experiments (1)

- N = 40
- M = 40
- Image size: 4000×3000

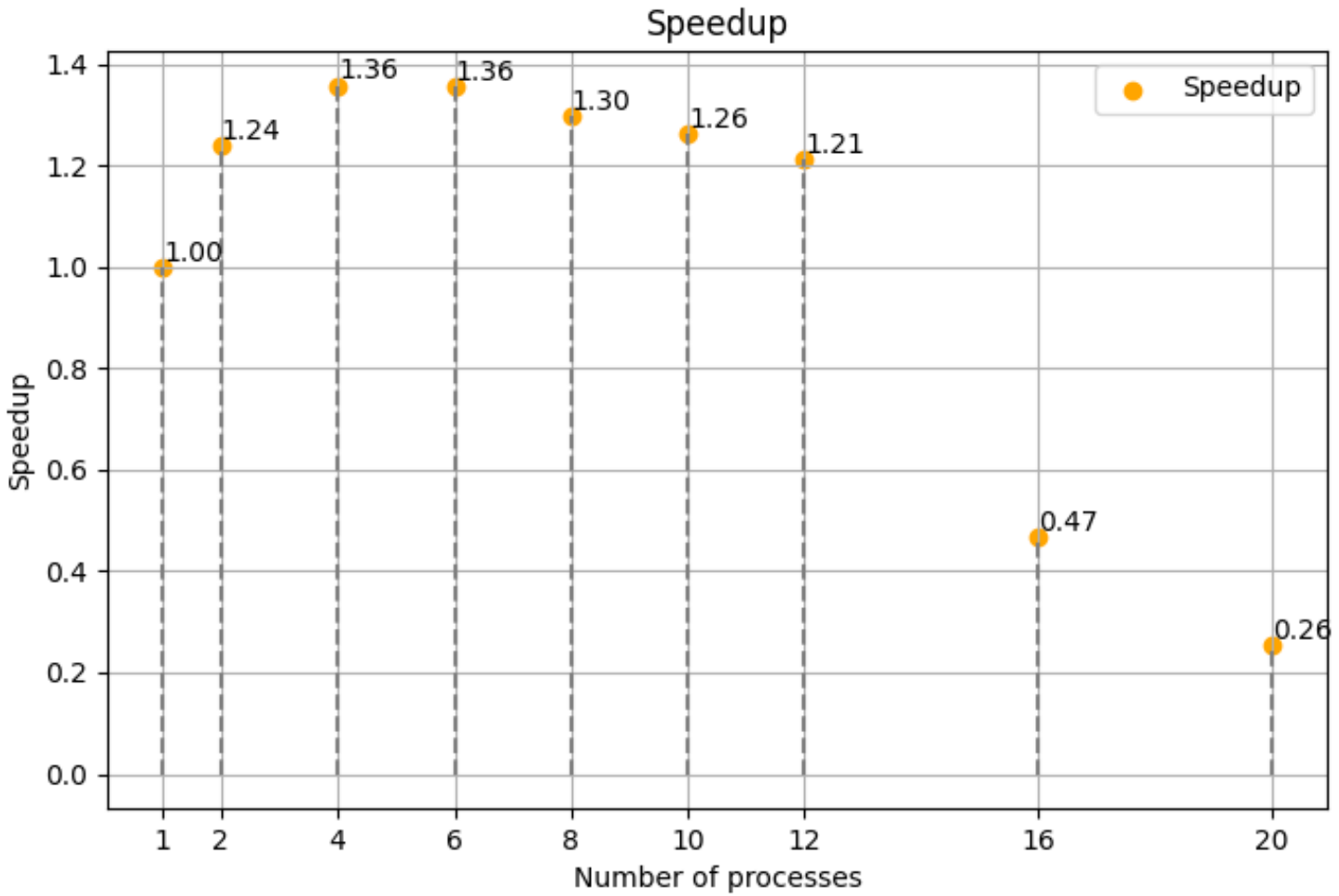
Sequential + Parallel versions

- 2 to 20 processes

UNIVERSITÀ DEGLI STUDI FIRENZE

Results (1)

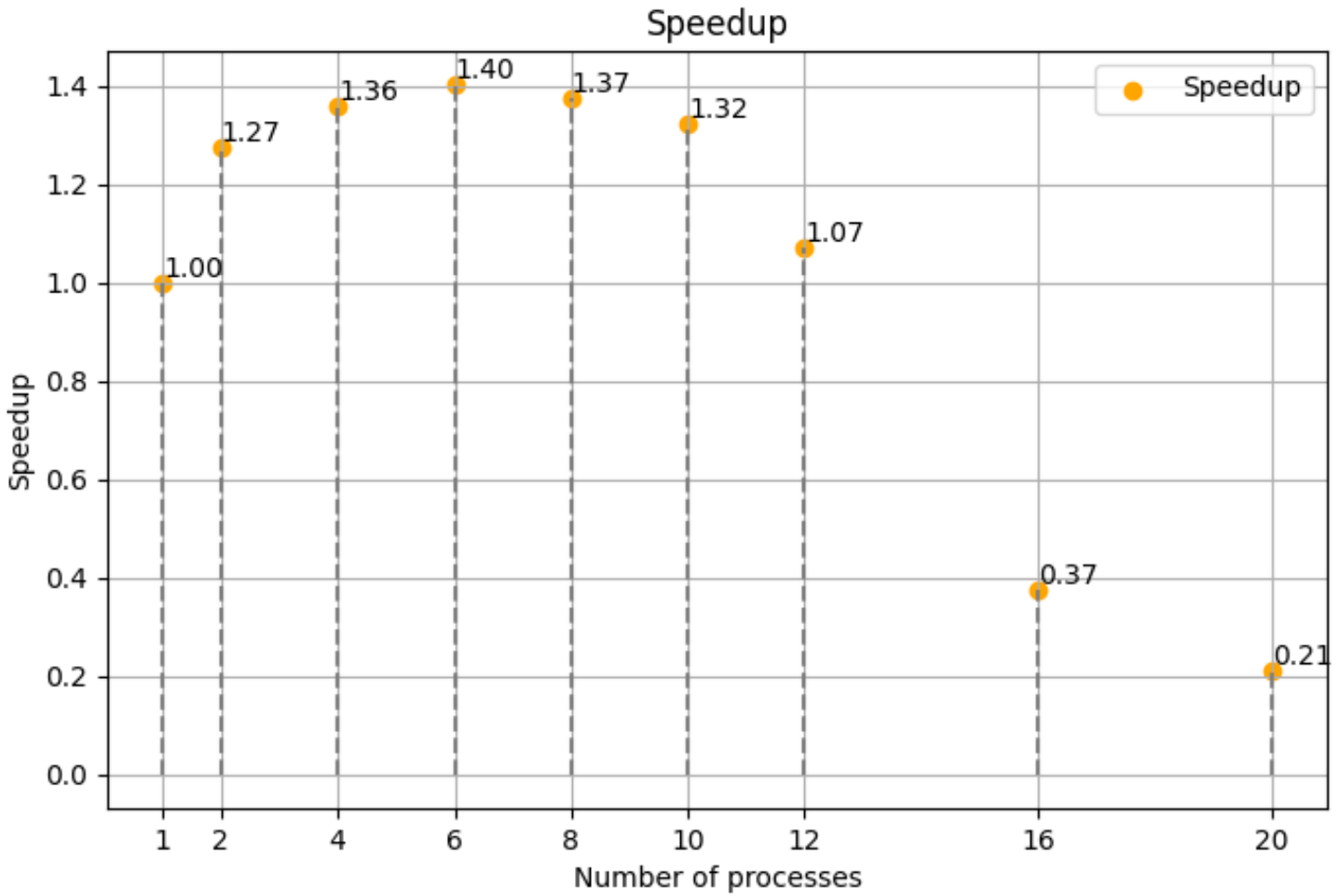
Parallelization on the augmentations



UNIVERSITÀ DEGLI STUDI FIRENZE

Results (1)

Parallelization on the images



UNIVERSITÀ DEGLI STUDI FIRENZE

Experiments (2)

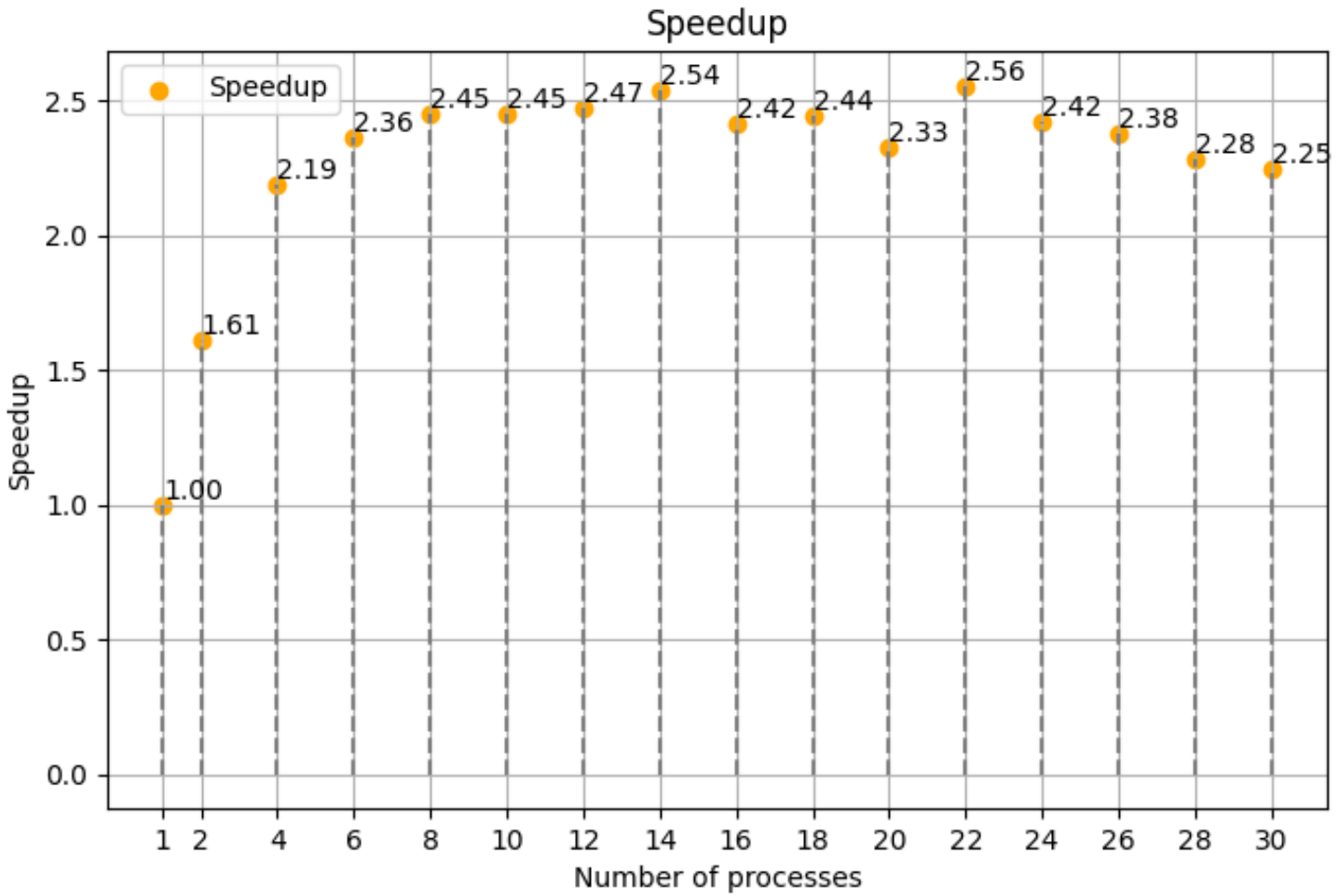
- N = 40
- M = 40
- Image size: 400×300

Sequential + Parallel versions

- 2 to 30 processes

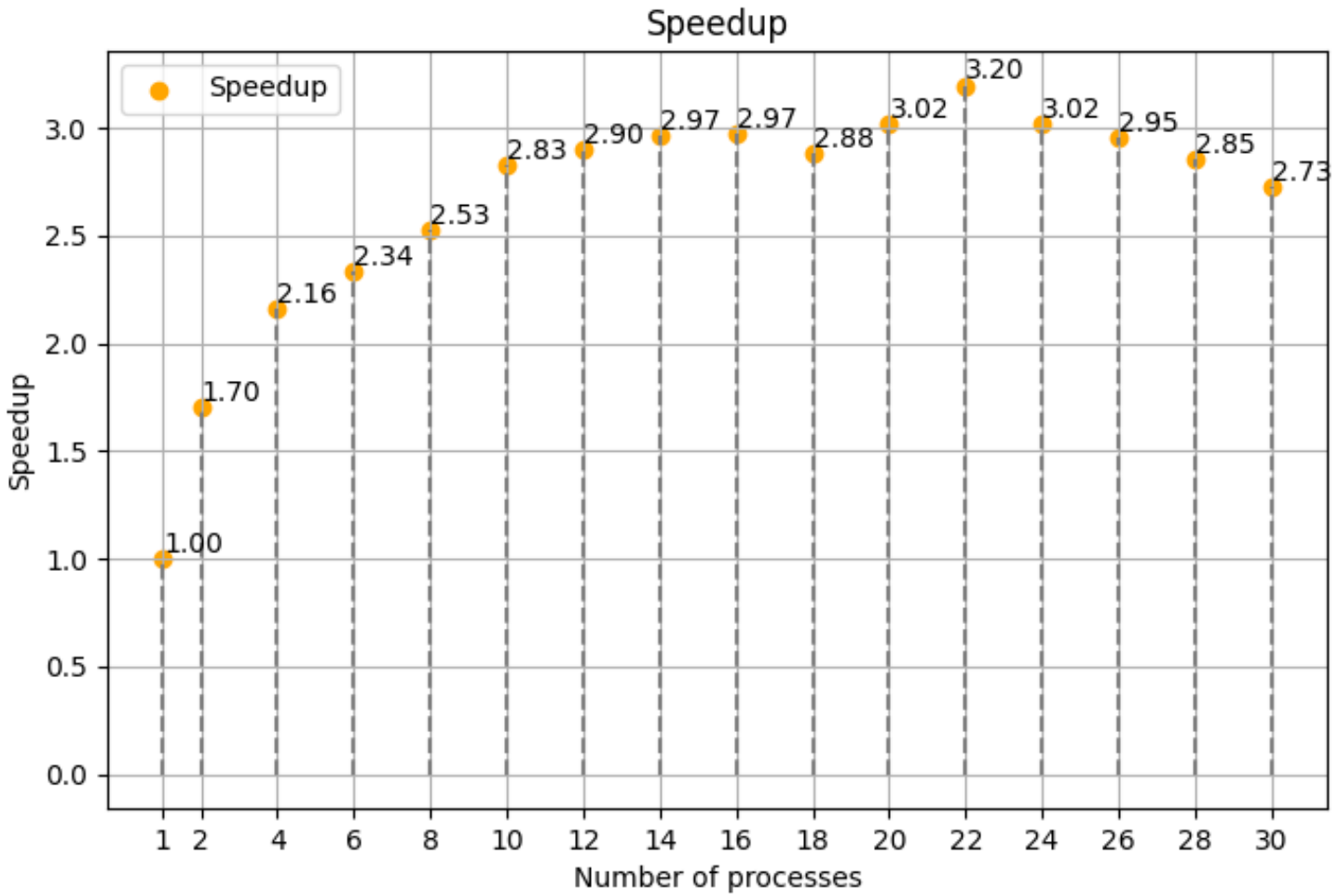
Results (2)

Parallelization on the augmentations



Results (2)

Parallelization on the images



Examples

Original:

Examples

Augmentations: