

- ▶ Creare uno script in python, il file «*synthetic_builder.py*», che permetta di creare una raccolta di dati sintetici. Un dataset sintetico.

Esercitazione

- ▶ Lo script dovrà contenere una classe dedicata per svolgere questo compito, chiamata «SyntheticBuilder».

Esercitazione

- ▶ Durante la sua inizializzazione, la classe riceverà in input il percorso ad un file di configurazione: *config.json*.

Esercitazione

- ▶ Il file, in formato *json*, conterrà tutte le informazioni necessarie per generare i dati sintetici. Di seguito un esempio di partenza:

Esercitazione

```
{
  "folders": {
    "out": "synth-builder/"
  },
  "params": {
    "pattern": ["*.bmp", "*.png"],
    "save_on_disk": true,
    "as_npz": false,
    "as_png": true,
    "number_of_synthetic_objects": 10000,
    "validation_percentage": 0.0,
    "test_percentage": 0.0
  },
  "synth-image": {
    "builder-type": "colored_shapes",
    "output_size": [256, 256, 3]
  },
  "colored_shapes": {
    "use_square": true
    "use_triangle": true
    "use_circle": true
    "use_pentagone": true
    "do_rotation": true
    "do_scaling": true
    "do_translation": true
    "do_flip": true
  }
}
```

Esercitazione

- ▶ In fase di inizializzazione, la classe leggerà il file json scorrendo tutte le sezioni necessarie e settando le proprie, necessarie, variabili membro.

Esercitazione

- ▶ La lettura di un file json è possibile tramite il package *json*.

(Rif: [json](#), [Read JSON file using Python](#))

Esercitazione

- ▶ Dovranno essere creati 3 gruppi di dati sintetici, divisi nelle categorie *training*, *test* e *validazione*. Le 3 categorie saranno a tutti gli effetti 3 sotto-cartelle all'interno della cartella di output.

Esercitazione

- ▶ Con un apposito metodo «build», il generatore sintetico dovrà:
 - ▶ Eseguire un for loop per x volte, dove x è il numero di oggetti di training da creare.
 - ▶ Eseguire un for loop per y volte, dove y è il numero di oggetti di testing da creare.
 - ▶ Eseguire un for loop per z volte, dove z è il numero di oggetti di validazione da creare.

Esercitazione

- ▶ Ad ogni passo dei cicli, si dovrà:
 - ▶ Creare l'immagine sintetica lanciando il metodo «create_synthetic_object».
 - ▶ Applicare preprocessing all'immagine creata lanciando il metodo «apply_preprocessing».

Esercitazione

- ▶ Le immagini create dovranno essere salvate su disco all'interno delle cartelle di appartenenza: training, test e validazione.
- ▶ Oppure salvate all'interno di tre file separati in formato *.npz*:
 - ▶ *train.npz*
 - ▶ *test.npz*
 - ▶ *validation.npz*

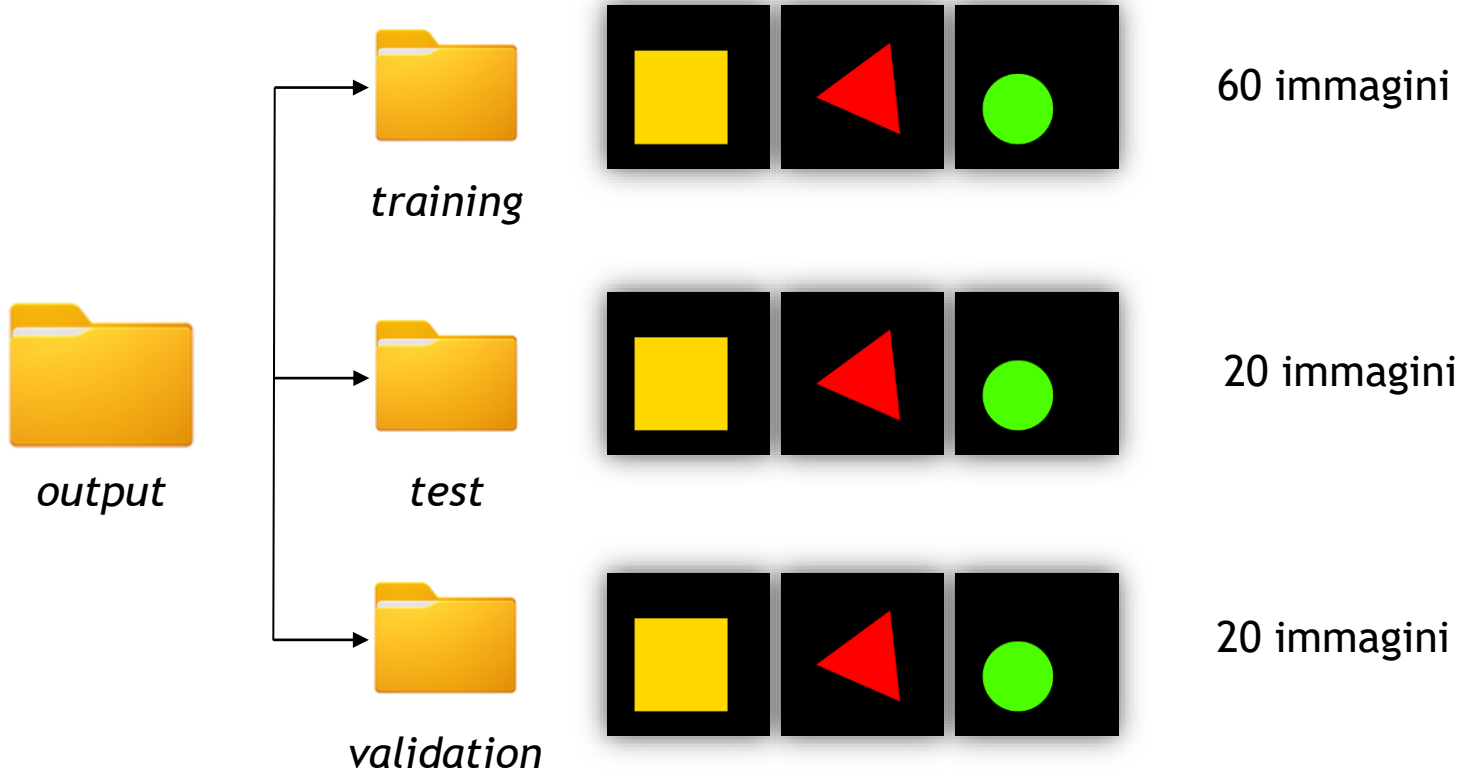
Npz è un formato file fornito da *numpy* e permette di raccogliere in un'unica entità una raccolta di array multidimensionali.

- ▶ [numpy.savez](#)
- ▶ [numpy.load](#)

Esercitazione

- ▶ In conclusione, creare dataset sintetico.
- ▶ Per crearlo, utilizzare la classe SyntheticBuilder.
- ▶ Creare 100 immagini sintetiche divise in:
 - ▶ 60 immagini di training.
 - ▶ 20 immagini di test.
 - ▶ 20 immagini di validazione.
- ▶ Ogni immagine dovrà contenere un solo oggetto, eventualmente preprocessato, di una di tre classi:
 - ▶ Un cerchio verde.
 - ▶ Un triangolo rosso.
 - ▶ Un quadrato giallo.
- ▶ Gli oggetti saranno disegnati su sfondo nero.

Esercitazione



Esercitazione