

# Datenaugmentierung

---

## data\_augmentation.py

Argumente:

- *train\_dir*: Ordner der zu augmentierende Bilder enthält (auch rekursiv)
- *n\_max* Anzahl an Bilder die je Klasse erzeugt werden sollen
- *color* Anzahl der *Augmenter* aus der Liste *color\_augmenters* für jedes bild angewendet werden sollen
- *geometric* Anzahl der *Augmenter* aus der Liste *geometric\_augmenters* für jedes bild angewendet werden sollen
- *all* (optional) wie viele Augmentierungen je bilde angewendet werden sollen (ohne spezifizierung)

Beispielhafte Ausführung:

```
python3 data_augmentation.py -t train/ -n 3000 -c 1 -g 1
```

Der *train* Ordner muss die Klassen separat in Unterordnern enthalten. Die Label Files können im gleichem Ordner wie die Bilder oder in einem weiterem Unterordner sein.

Bsp:

```
└─ train
   └─ Deer
      ├── Label
      │   ├── img1.txt
      │   └── img2.txt
      ├── img1.jpg
      └── img2.jpg
   └─ Fox
      └─ ...
```

oder:

```
└─ train
   ├── Deer
   │   ├── img1.txt
   │   ├── img1.jpg
   │   ├── img2.txt
   │   └── img2.jpg
   └── Fox
       └── ...
```

Augmentierungen stammen aus [imgaug](#) und sind in [data\\_augmentation.py](#) in den Listen *color\_augmenters* und *geometric\_augmenters* definiert.

## visualize\_augmentations.py

Zum Anzeigen der Augmentierten Bilder mit eingezeichneter Bounding Box

dafür:

```
python3 visualize_augmentations.py path/to/folder
```

dabei wird anhand gleicher namen von Bild und Label-File die Box zugeordnet und eingezeichnet.

## test\_augmenters.ipynb

Jupyter Notebook zum testen verschiedener Augmentern aus [imgaug](#) auf Bilder in images/ Ordner.

- [test\\_augmenters.ipynb](#)