

Some Data Augmentation Libraries:

<https://github.com/aleju/imgaug>

Tuning Parameters:

* Dataset format (one well-defined fix format by default, reasonable extension point for custom support) -> torchvision.datasets.ImageFolder (folder/s inside folder), possible solution: specified dataset format(in np.array?)
* number of representatives selected
* Representative selection algorithm (i.e. our method, none, random, etc.)
* Augmentation class whitelist/blacklist
* Placeholder
* Hyperparam for OPTICS/DBSCAN/KMM/Agglomerative/SinGAN (and their default setting)

**CLI API:**

python singleimageaugmentation.py

-h, --help Displays help

-r, --num-representatives Number of representatives to select from each class for training (default=5)

-i, --input Input dataset location

-o, --output\_dir Output directory location

-d, --diversity-analysis-algorithm Diversity analysis (i.e. None [default], SSIM, some other comparison)\*

-a, --single-image-augmentation-algorithm Single image GAN to use for augmentation (default=SinGAN)

-p, --model-parameters Input file to override default Single image GAN parameters (default=None)\*\*

-b, --class-blacklist Input file containing a blacklist for classes to ignore in augmentation (new-line delimited)

-w, --class-whitelist Input file containing a whitelist for classes to include in augmentation (new-line delimited),

-g, --gpu Enable/disable GPU (NOTE: Maybe should auto-detect)

\* Diversity analysis is an informational output

\*\* Model parameters should be a well-defined, structured JSON file. This allows tuning of single image GAN implementation, DBSCAN, etc. all in a single input.

**Rough idea:**

{

“<TunableSystemName>”: {

“<Parameter>”: “<value>”

}

}

**Example:**

{

“SinGAN”: {

“num\_layers”: 2

}

}

**Action Items:**

Uzay - (i) Add geometric transformations.

Create the JSON file.