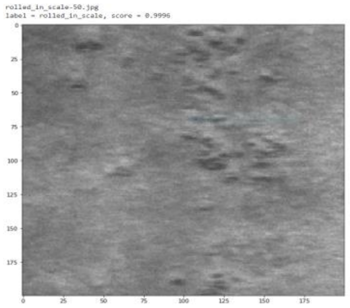
Image-Classification-CSPResNeXt-50-CPP-Jupyter

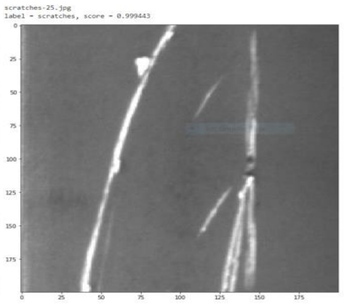
The CSPNet algorithm optimizes the deep learning network structure. In addition to the world's first accuracy, it can also maintain ultra-high inference speed on edge computing embedded systems.

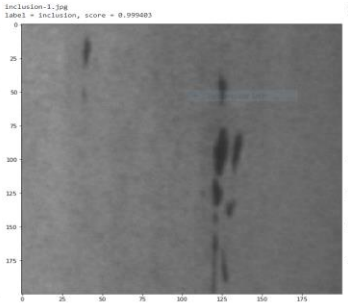
Version 20230223

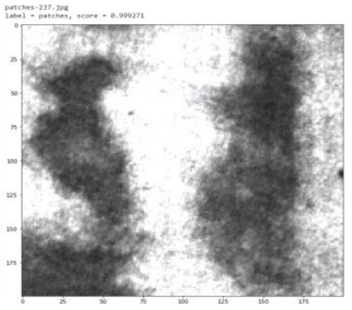
Applications

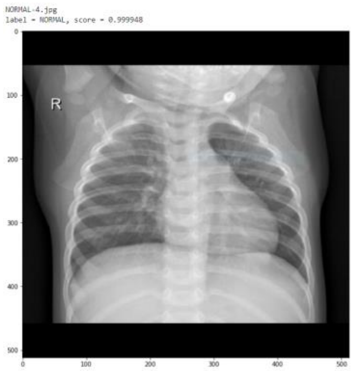
- Factory defect detection
- Traffic flow analysis
- People flow analysis
- Safety monitoring
- Medical image analysis
- Biological image analysis
- Industrial safety image analysis
- Mask image analysis
- Others…

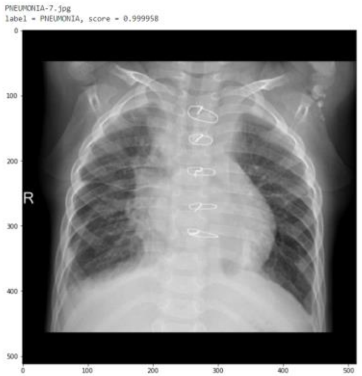




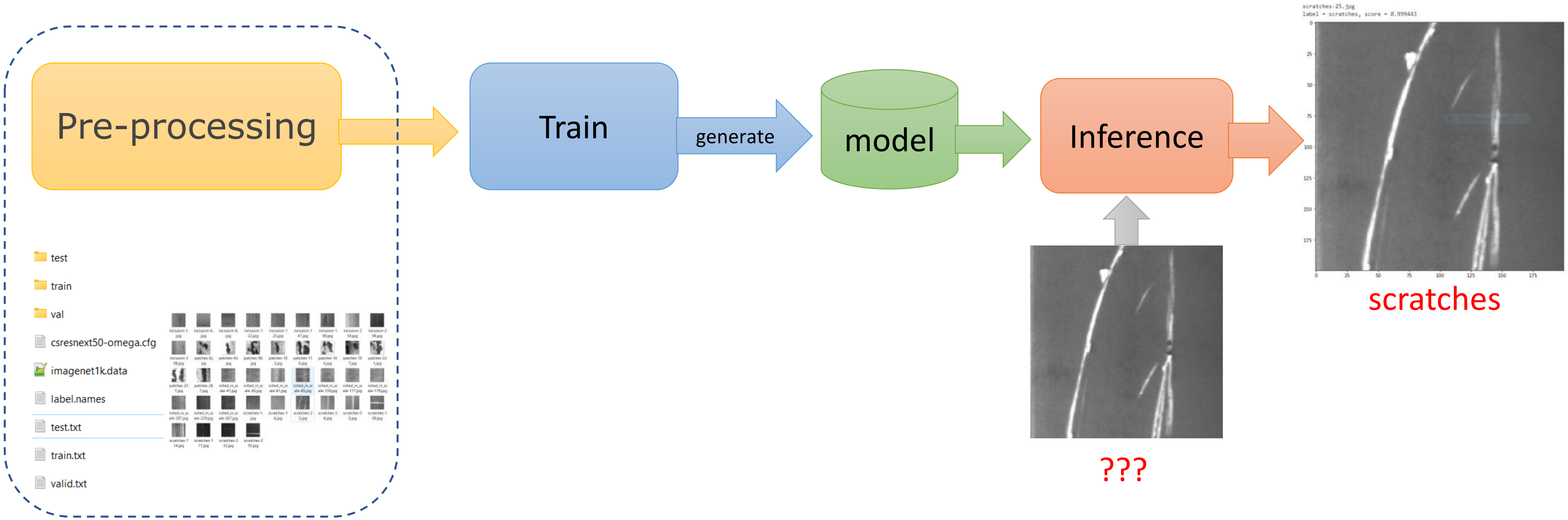








Flow Chart



test
train
val
csresnext50-omega.cfg
imagenet1k.data
label.names
test.txt
train.txt
valid.txt

```
scratches-25.jpg
label = scratches, score = 0.999443
```

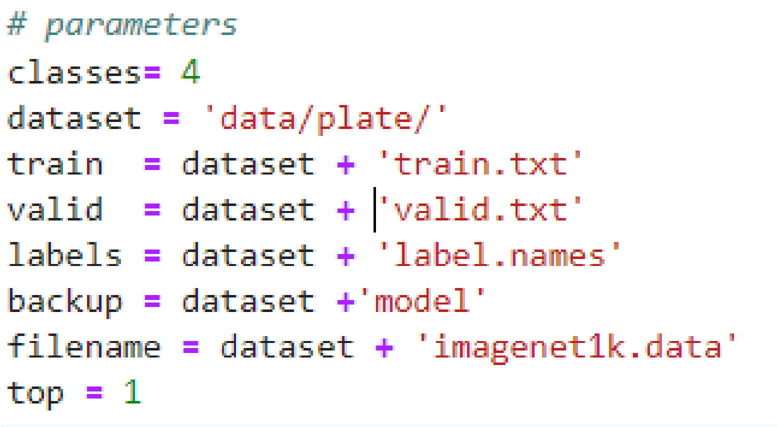
Data Preparation

- If you want to train your image, please copy a default data folder and change it to your own data name.

- Then find the train, val, and test folders, delete the old images in the folders and replace them with images that you need to train, infer, and test.
  - Note:
    - The file name of the image needs to comply with the requirements: The file name of the image must be prefixed with [category name-], for example: inclusion in inclusion-1.jpg is the category name of the image.
    - The image is recommended as a square. If it is not a square, you can use zoom or crop, etc.
    - The image extension must be .jpg or .png or .jpeg.

1_prepare_config_file.ipynb

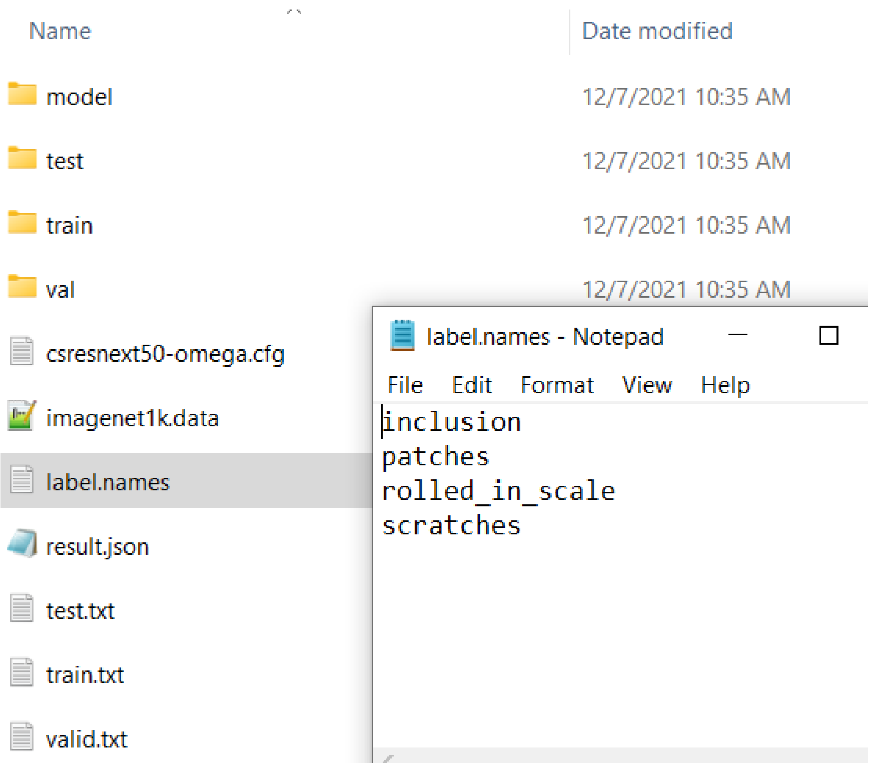
- classes : Set the number of categories.
  - For example: the training image has 5 categories => classes = 5.
- dataset : Set the path. Please change the data/plate to the data/folder name.
  - For example, in the previous step to establish their own sample data and name is pcb, this step needs to be set.



```
# parameters
classes= 4
dataset = 'data/plate/'
train  = dataset + 'train.txt'
valid  = dataset + 'valid.txt'
labels = dataset + 'label.names'
backup = dataset +'model'
filename = dataset + 'imagenet1k.data'
top = 1
```

Set the Category Name

- Open the label.names, and set the training category name.

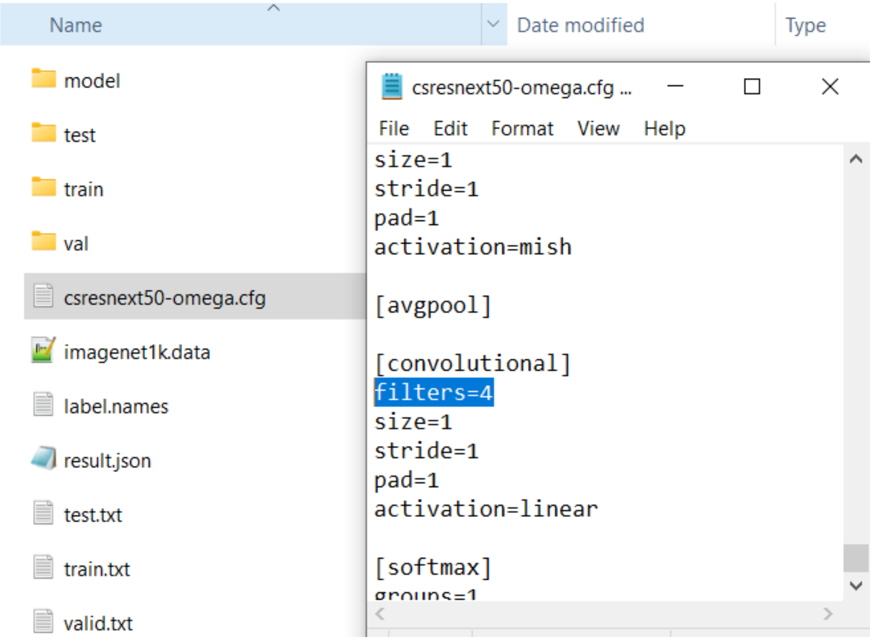


Name	Date modified
model	12/7/2021 10:35 AM
test	12/7/2021 10:35 AM
train	12/7/2021 10:35 AM
val	12/7/2021 10:35 AM
csresnext50-omega.cfg	
imagenet1k.data	
label.names	
result.json	
test.txt	
train.txt	
valid.txt	

```
inclusion
patches
rolled_in_scale
scratches
```

.cfg Configuration File

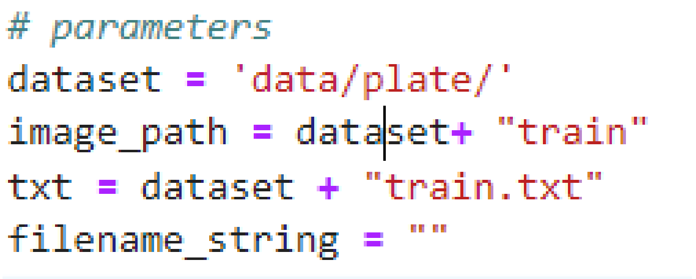
- Edit CSPNet.cfg and find [avgpool]. Below this you can find filters=4. The number 4 is the number of categories of your data.
- For example: the training images have 5 categories => filters=5.



```
size=1
stride=1
pad=1
activation=mish

[avgpool]

[convolutional]
filters=4
size=1
stride=1
pad=1
activation=linear

[softmax]
groups=1
```

2_prepare_train_txt.ipynb

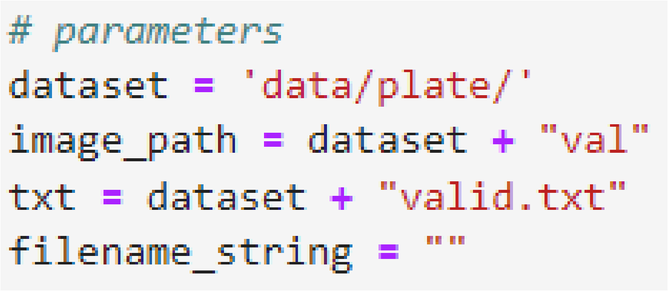
- Set dataset and Run [2. prepare train txt] to automatically generate a list of images for training.



```
# parameters
dataset = 'data/plate/'
image_path = dataset+ "train"
txt = dataset + "train.txt"
filename_string = ""
```

3_prepare_val_txt.ipynb

- Set dataset and Run [3. prepare val txt] to automatically generate a list of images for verification.

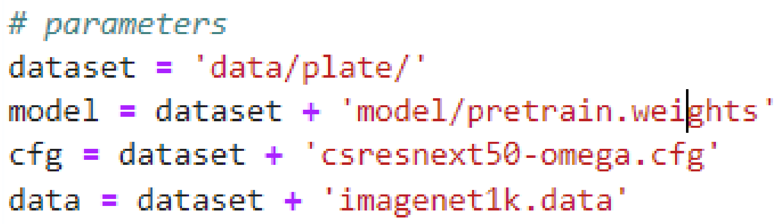


```
# parameters
dataset = 'data/plate/'
image_path = dataset + "val"
txt = dataset + "valid.txt"
filename_string = ""
```

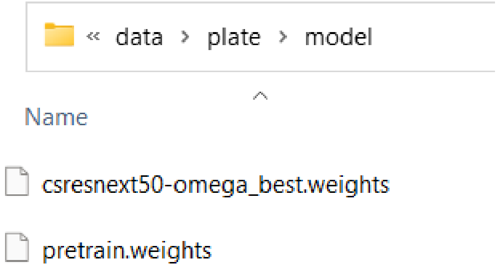
4_train_GPU.ipynb

- Set dataset and if the device supports GPU accelerated computing, please select 6_train_GPU, otherwise, please select 6_train_CPU.

- If you are not training for the first time and you have already trained a model, to continue training, please copy model/csresnext50-omega_best.weights and name it pretrain.weights to overwrite the original pretrain.weights.



```
# parameters
dataset = 'data/plate/'
model = dataset + 'model/pretrain.weights'
cfg = dataset + 'csresnext50-omega.cfg'
data = dataset + 'imagenet1k.data'
```



« data › plate › model

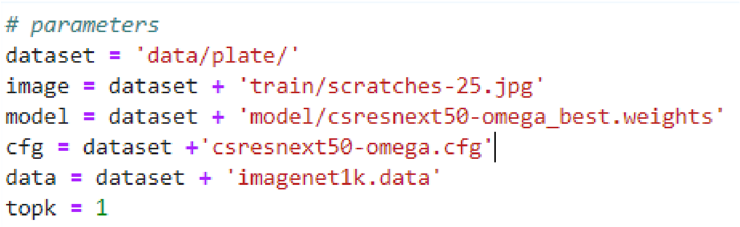
Name

csresnext50-omega_best.weights

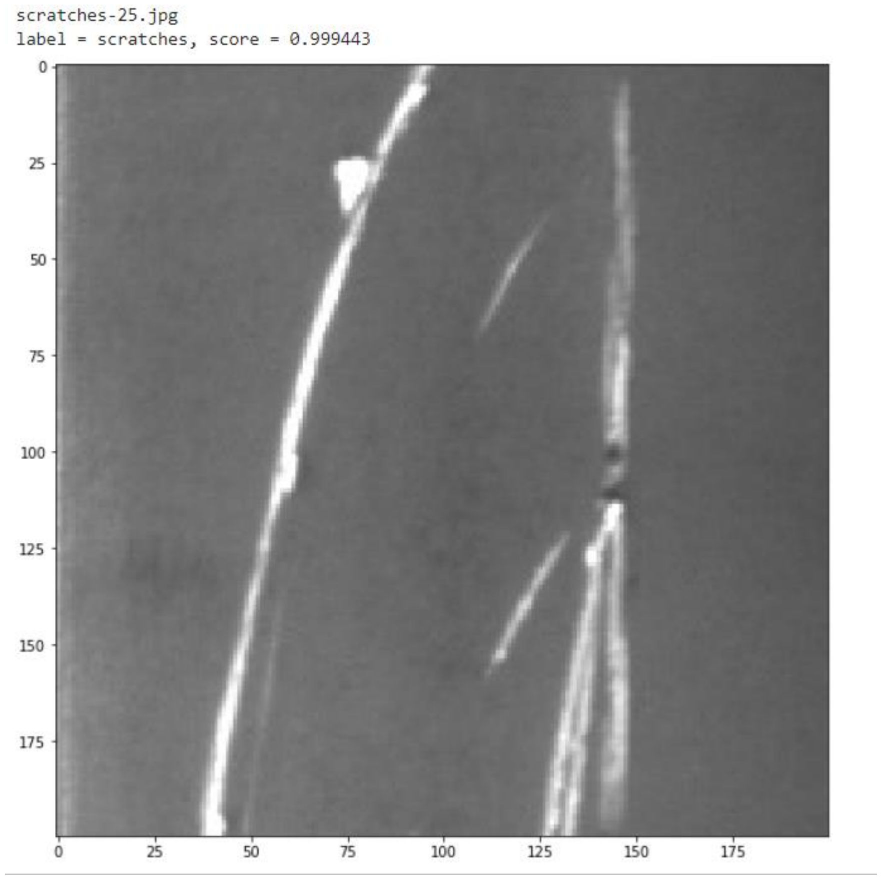
pretrain.weights

5_inference_GPU.ipynb

- Inferring a single image.
  - dataset : the path of dataset.
  - model : Inferred model.
  - Image : Inferred image.



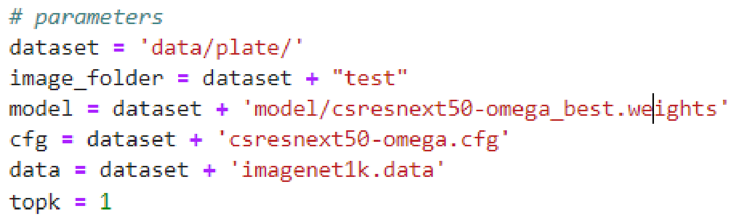
```
# parameters
dataset = 'data/plate/'
image = dataset + 'train/scratches-25.jpg'
model = dataset + 'model/csresnext50-omega_best.weights'
cfg = dataset +'csresnext50-omega.cfg'
data = dataset + 'imagenet1k.data'
topk = 1
```



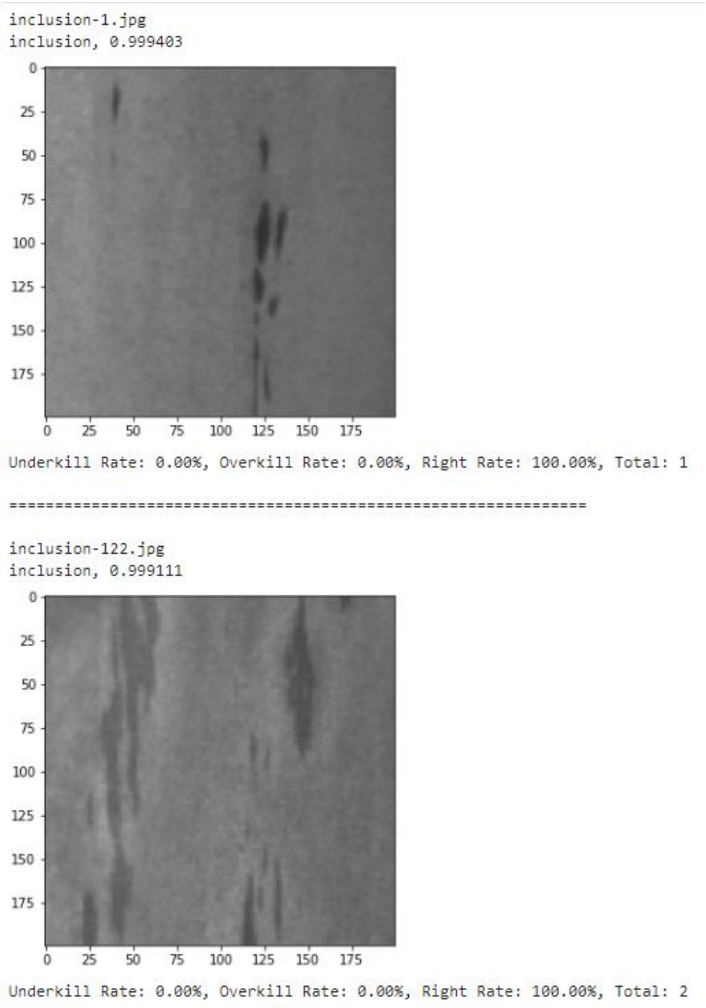
```
scratches-25.jpg
label = scratches, score = 0.999443
```

6_inference_folder_1_GPU.ipynb

- Infer all images in the folder.
  - dataset : the path of dataset.
  - image_folder : Inference folder (jpg, png).
  - model_path : Inferred model.



```
# parameters
dataset = 'data/plate/'
image_folder = dataset + "test"
model = dataset + 'model/csresnext50-omega_best.weights'
cfg = dataset + 'csresnext50-omega.cfg'
data = dataset + 'imagenet1k.data'
topk = 1
```



```
inclusion-1.jpg
inclusion, 0.999403
```

```
Underkill Rate: 0.00%, Overkill Rate: 0.00%, Right Rate: 100.00%, Total: 1

================================================================

inclusion-122.jpg
inclusion, 0.999111
```

```
Underkill Rate: 0.00%, Overkill Rate: 0.00%, Right Rate: 100.00%, Total: 2
```

Reference

- Please refer to the readme.txt in the SDK folder.
- LEADERG AppForAI : https://www.leaderg.com/appforai-windows
-