Yolo V7 for industry object detection application

Theerawat 9 Sep 2022

Inspiration:

Yolo V7 is the AI Vision opensource software which delivers the best performance both accuracy and speed yet easy and make most sense ever of development environment which key trigger of the deployment. Yolo V7 is also available from single broad computer like Jetson nano onward which suitable for low cost project such AI at the Edge application.

Topic:

- Introduction: What is YOLO ? Who is Inventor ?
- Installation: Requirement
- Pretrain model Inference demo
- Custom model development demo

Introduction: What is YOLO ? Who is Inventor ?

- https://pjreddie.com/darknet/
- https://www.youtube.com/watch?v=Cgxsv1riJhI
- https://pjreddie.com/resume/
- https://github.com/WongKinYiu/yolov7

Installation: Requirement

- CPU ARM or x86
- GPU nVIDIA (option for Custom Dataset Model Training)
- OS 64 Bit
- Python 3.8 Environment such as Anaconda
- Source code https://github.com/theerawatramchuen/yolov7

Procedure:

- Create Python 3.8 with anaconda
  - ➢ conda create -n yolov7 python=3.8

```
(base) PS E:\> conda create -n yolov7 python=3.8
```

  - ➢ conda activate yolov7

```
(base) PS E:\> conda activate yolov7
(yolov7) PS E:\> |
```

Procedure:

- Clone source code from github

  ➢ git clone https://github.com/theerawatramchuen/yolov7

  ```
  (yolo7) PS E:\> git clone https://github.com/theerawatramchuen/yolov7
  Cloning into 'yolov7'...
  remote: Enumerating objects: 214, done.
  remote: Counting objects: 100% (110/110), done.

  remote: Total 214 (delta 10), reused 100 (delta 7), pack-reused 104
  Receiving objects: 100% (214/214), 58.52 MiB | 1.13 MiB/s, done.
  Resolving deltas: 100% (39/39), done.
  (yolo7) PS E:\>
  ```

  ➢ cd yolov7

  ```
  (yolo7) PS E:\> cd yolov7
  (yolo7) PS E:\yolov7>
  ```

Procedure:

- Update Installation Software

  ➢ python -m pip install --upgrade pip

```
(yolo7) PS E:\yolov7> python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\41162395\anaconda3\envs\yolo7\lib\site-packages (22.1.2)
Collecting pip
  Using cached pip-22.2.2-py3-none-any.whl (2.0 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.1.2
    Uninstalling pip-22.1.2:
      Successfully uninstalled pip-22.1.2
Successfully installed pip-22.2.2
(yolo7) PS E:\yolov7>
```

Procedure:

- Install yolov7 to our machine
  - ➢ pip install -r requirements.txt

```
Installing collected packages: wcwidth, tensorboard-plugin-wit, pytz, pyasn1, pure-eval, pickleshare, executing, backcal
l, zipp, urllib3, typing-extensions, traitlets, tensorboard-data-server, six, rsa, PyYAML, pyparsing, pygments, pyasn1-m
odules, psutil, protobuf, prompt-toolkit, Pillow, parso, oauthlib, numpy, MarkupSafe, kiwisolver, idna, fonttools, decor
ator, cycler, colorama, charset-normalizer, cachetools, absl-py, werkzeug, tqdm, torch, scipy, requests, python-dateutil
, packaging, opencv-python, matplotlib-inline, jedi, importlib-metadata, grpcio, google-auth, asttokens, torchvision, th
op, stack-data, requests-oauthlib, pandas, matplotlib, markdown, seaborn, ipython, google-auth-oauthlib, tensorboard
Successfully installed MarkupSafe-2.1.1 Pillow-9.2.0 PyYAML-6.0 absl-py-1.2.0 asttokens-2.0.8 backcall-0.2.0 cachetools-
5.2.0 charset-normalizer-2.1.1 colorama-0.4.5 cycler-0.11.0 decorator-5.1.1 executing-1.0.0 fonttools-4.37.1 google-auth
-2.11.0 google-auth-oauthlib-0.4.6 grpcio-1.47.0 idna-3.3 importlib-metadata-4.12.0 ipython-8.4.0 jedi-0.18.1 kiwisolver
-1.4.4 markdown-3.4.1 matplotlib-3.5.3 matplotlib-inline-0.1.6 numpy-1.23.2 oauthlib-3.2.0 opencv-python-4.6.0.66 packag
ing-21.3 pandas-1.4.4 parso-0.8.3 pickleshare-0.7.5 prompt-toolkit-3.0.30 protobuf-3.19.4 psutil-5.9.1 pure-eval-0.2.2 p
yasn1-0.4.8 pyasn1-modules-0.2.8 pygments-2.13.0 pyparsing-3.0.9 python-dateutil-2.8.2 pytz-2022.2.1 requests-2.28.1 req
uests-oauthlib-1.3.1 rsa-4.9 scipy-1.9.1 seaborn-0.11.2 six-1.16.0 stack-data-0.5.0 tensorboard-2.10.0 tensorboard-data-
server-0.6.1 tensorboard-plugin-wit-1.8.1 thop-0.1.1.post2207130030 torch-1.12.1 torchvision-0.13.1 tqdm-4.64.0 traitlet
s-5.3.0 typing-extensions-4.3.0 urllib3-1.26.12 wcwidth-0.2.5 werkzeug-2.2.2 zipp-3.8.1
(yolo7) PS E:\yolov7> |
```

- Download pretrain model to {yolov7} folder (option)
  - ➢ https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt

Procedure:

- Verify Installation

  ➢ python detect.py --weights yolov7-tiny.pt --source inference/images/horses.jpg --img 640


```
The image with the result is saved in: runs\detect\exp4\horses.jpg
Done. (0.293s)
(yolo7) PS E:\yolov7> |
```

Output images

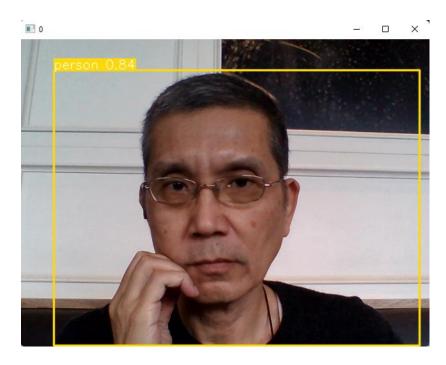**Input images**

```
Anaconda Powershell Prompt (Anaconda3)                                    —    □    ✕

(base) PS C:\Users\ramch> f:
(base) PS F:\> cd yolov7
(base) PS F:\yolov7> conda activate yolov7
(yolov7) PS F:\yolov7> python detect.py --weights yolov7-tiny.pt --source inference/images/horses.jpg --img 640
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.25, device='', exist_ok=False, img_size=640, iou
_thres=0.45, name='exp', no_trace=False, nosave=False, project='runs/detect', save_conf=False, save_txt=False, source='i
nference/images/horses.jpg', update=False, view_img=False, weights=['yolov7-tiny.pt'])
YOLOR  v0.1-105-g064c71e torch 1.12.1 CUDA:0 (NVIDIA GeForce GTX 1070, 8191.6875MB)

Downloading https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7-tiny.pt to yolov7-tiny.pt...
100%|████████████████████████████████████████████████| 12.1M/12.1M [00:03<00:00, 3.20MB/s]

Fusing layers...
Model Summary: 200 layers, 6219709 parameters, 229245 gradients
 Convert model to Traced-model...
 traced_script_module saved!
 model is traced!

C:\Users\ramch\.conda\envs\yolov7\lib\site-packages\torch\functional.py:478: UserWarning: torch.meshgrid: in an upcoming
 release, it will be required to pass the indexing argument. (Triggered internally at  C:\cb\pytorch_1000000000000\work\
aten\src\ATen\native\TensorShape.cpp:2895.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
5 horses, Done. (12.0ms) Inference, (46.0ms) NMS
 The image with the result is saved in: runs\detect\exp2\horses.jpg
Done. (0.154s)
(yolov7) PS F:\yolov7>
(yolov7) PS F:\yolov7>
```

Pretrain model Inference demo

Input USB web-cam ID 0

- Live inference with web-cam id 0

  ➢ python detect.py --weights yolov7-tiny.pt --source 0 --img 640
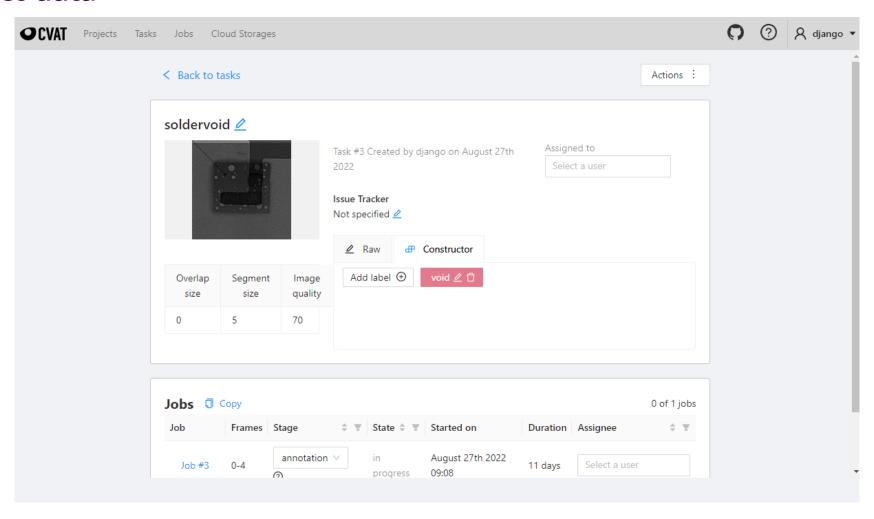

Live Output on Screen

Custom model development demo

1. Collect data
2. Label data
3. Split data (train & test)
4. Prepare config files
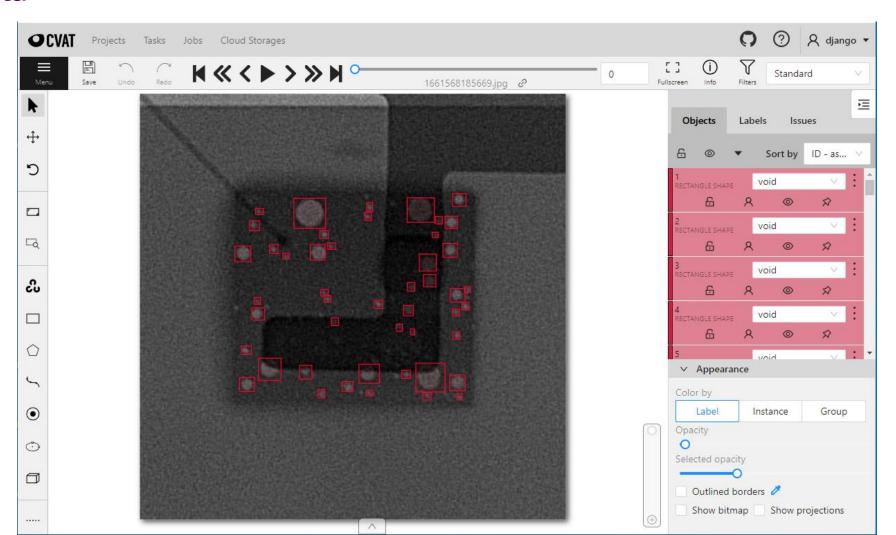5. Start training (Need GPU nVIDIA)
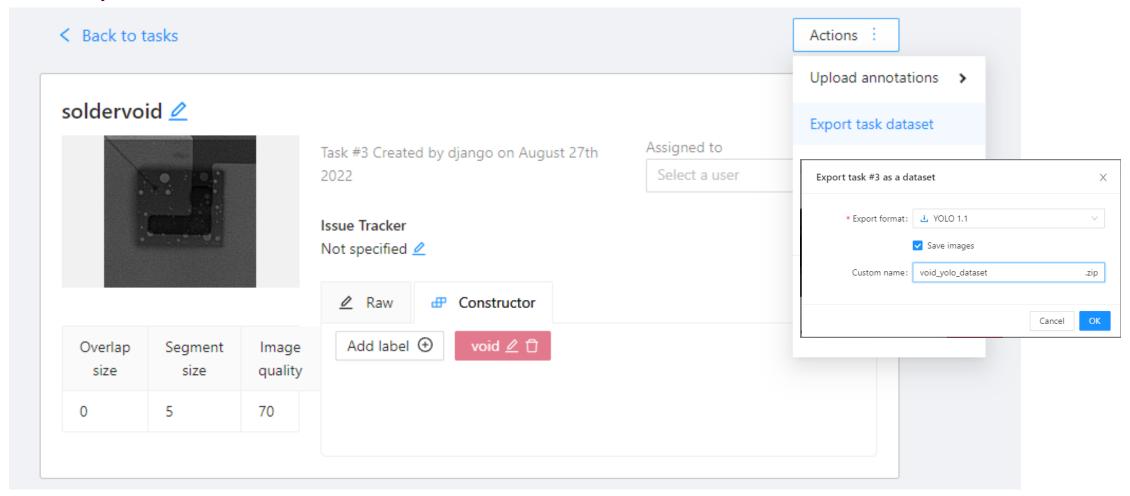6. Use custom model

Custom model development demo
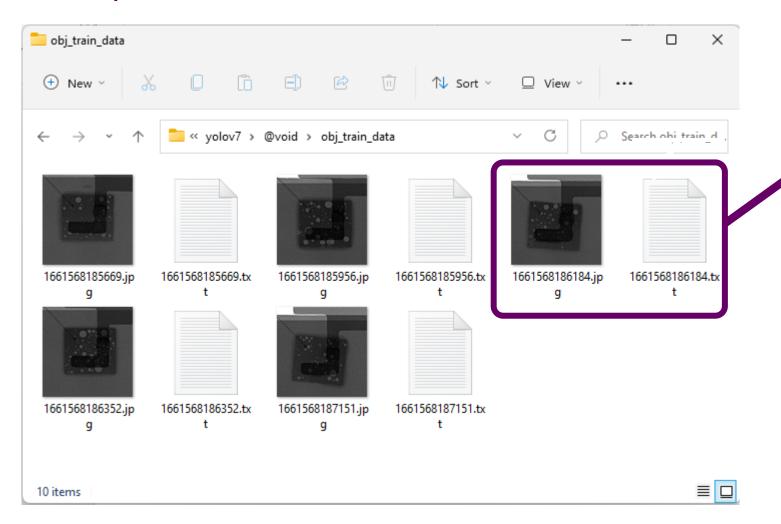1. Collect data

Custom model development demo
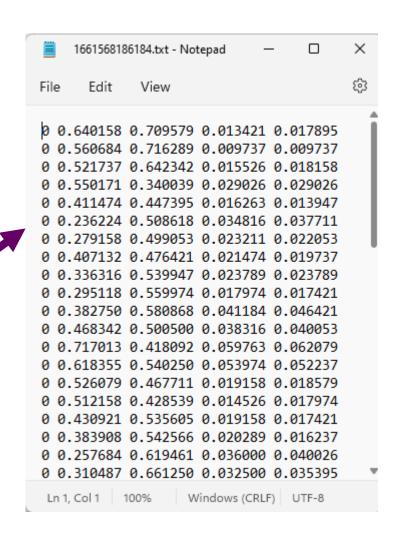## 2. Label data

Custom model development demo
2.1 Export from CVAT in Yolo format

Custom model development demo
2.1 Export from CVAT in Yolo format
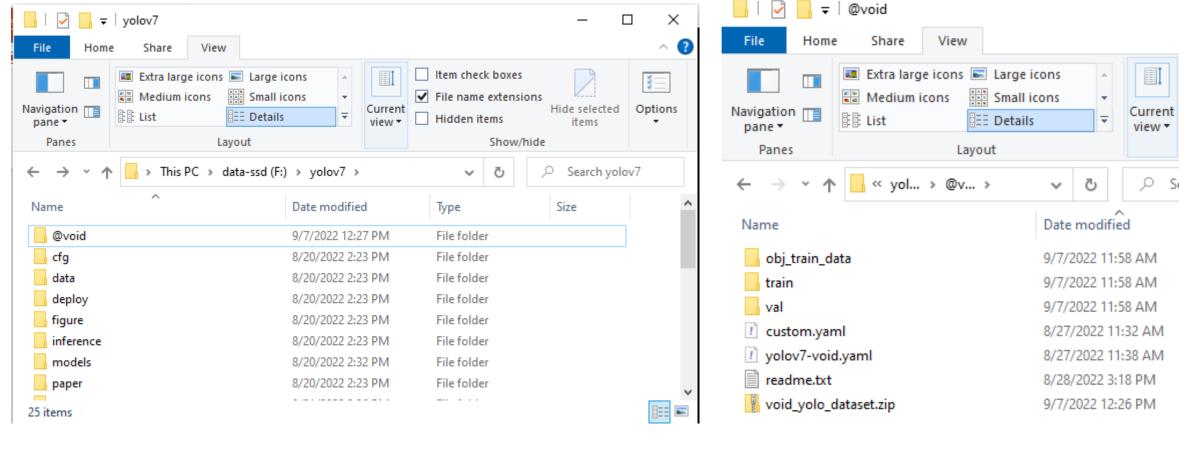
Custom model development demo
3. Split data (train & validate)

Custom model development demo
4.1 Create config files   "custom.yaml"

train = directory of train images with labeling files
val = directory of validation images with labeling files
nc = Number of classes
names = Project name

Custom model development demo
4.2 Edit model "yolov7-void.yaml" file for require number of class *{line #2}*

Custom model development demo
5. Start training (Need GPU nVIDIA)

# Script for model development## Training the model
$ cd yolov7
$ python train.py --weights yolov7.pt --data "@void/custom.yaml" --workers 2 --batch-size 2 --img 416 --cfg @void/yolov7-void.yaml --name void --hyp data/hyp.scratch.p5.yaml --epochs 1000

```
Anaconda Powershell Prompt (Anaconda3)                              —    □    ✕

(yolov7) PS F:\yolov7> python train.py --weights yolov7.pt --data "@void/custom.yaml" --workers 2 --batch-size 2 --img 4
16 --cfg @void/yolov7-void.yaml --name void --hyp data/hyp.scratch.p5.yaml --epochs 1000_
```

Yolo V7 for industry object detection application


```
Anaconda Powershell Prompt (Anaconda3)                                    —   □   ×

   Epoch   gpu_mem        box        obj        cls      total     labels   img_size
 995/999     1.52G    0.09606    0.02424          0     0.1203        181        416: 100%|█| 4/4 [00:00<00:00,  6.60it/s]
             Class     Images     Labels                P          R       mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
               all          3        207              0.121      0.155     0.0492     0.00715

   Epoch   gpu_mem        box        obj        cls      total     labels   img_size
 996/999     1.52G    0.09091    0.01577          0     0.1067          8        416: 100%|█| 4/4 [00:00<00:00,  6.71it/s]
             Class     Images     Labels                P          R       mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
               all          3        207              0.121      0.155     0.0492     0.00715

   Epoch   gpu_mem        box        obj        cls      total     labels   img_size
 997/999     1.52G    0.09981    0.01383          0     0.1136         85        416: 100%|█| 4/4 [00:00<00:00,  6.70it/s]
             Class     Images     Labels                P          R       mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
               all          3        207              0.121      0.155     0.0492     0.00715

   Epoch   gpu_mem        box        obj        cls      total     labels   img_size
 998/999     1.52G    0.09424    0.03525          0     0.1295         22        416: 100%|█| 4/4 [00:00<00:00,  5.30it/s]
             Class     Images     Labels                P          R       mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
               all          3        207              0.121      0.155     0.0492     0.00715

   Epoch   gpu_mem        box        obj        cls      total     labels   img_size
 999/999     1.52G    0.09088    0.01133          0     0.1022         11        416: 100%|█| 4/4 [00:00<00:00,  6.99it/s]
             Class     Images     Labels                P          R       mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
               all          3        207              0.121      0.155     0.0492     0.00715
1000 epochs completed in 0.530 hours.

Optimizer stripped from runs\train\void\weights\last.pt, 74.8MB
Optimizer stripped from runs\train\void\weights\best.pt, 74.8MB
(yolov7) PS F:\yolov7>
```
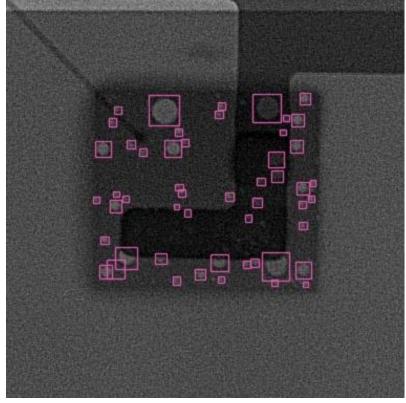
Custom model development demo
6. Use the custom model

Inference with new model
$ cd yolov7
$ python detect.py --weights runs/train/void/weights/best.pt --conf-thres 0.7 --source @void/obj_train_data

```
Anaconda Powershell Prompt (Anaconda3)                                                    —    □    ✕
(yolov7) PS F:\yolov7> python detect.py --weights runs/train/void/weights/best.pt --conf-thres 0.7 --source @void/obj_train_data
```

```
(yolov7) PS F:\yolov7> python detect.py --weights runs/train/void/weights/best.pt --conf-thres 0.7 --source @void/obj_train_data
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.7, device='', exist_ok=False, img_size=640, iou_thres=0.45,
name='exp', no_trace=False, nosave=False, project='runs/detect', save_conf=False, save_txt=False, source='@void/obj_train_data', upd
ate=False, view_img=False, weights=['runs/train/void/weights/best.pt'])
YOLOR  v0.1-105-g064c71e torch 1.12.1 CUDA:0 (NVIDIA GeForce GTX 1070, 8191.6875MB)

Fusing layers...
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
IDetect.fuse
C:\Users\ramch\.conda\envs\yolov7\lib\site-packages\torch\functional.py:478: UserWarning: torch.meshgrid: in an upcoming release, it
 will be required to pass the indexing argument. (Triggered internally at  C:\cb\pytorch_1000000000000\work\aten\src\ATen\native\Ten
sorShape.cpp:2895.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
Model Summary: 314 layers, 36481772 parameters, 6194944 gradients, 103.2 GFLOPS
 Convert model to Traced-model...
 traced_script_module saved!
 model is traced!

Done. (50.0ms) Inference, (1.0ms) NMS
 The image with the result is saved in: runs\detect\exp\1661568185669.jpg
Done. (46.0ms) Inference, (0.0ms) NMS
 The image with the result is saved in: runs\detect\exp\1661568185956.jpg
Done. (44.0ms) Inference, (0.0ms) NMS
 The image with the result is saved in: runs\detect\exp\1661568186184.jpg
Done. (44.0ms) Inference, (0.0ms) NMS
 The image with the result is saved in: runs\detect\exp\1661568186352.jpg
Done. (42.0ms) Inference, (0.0ms) NMS
 The image with the result is saved in: runs\detect\exp\1661568187151.jpg
Done. (0.276s)
(yolov7) PS F:\yolov7>
```

1661568185669.jpg 1661568185956.jpg 1661568186184.jpg 1661568186352.jpg 1661568187151.jpg

Appendix

Custom AI model Training input compared to Human knowledge development

Custom AI model Training inputs	Human knowledge development	Comment
weight files *{filename.pt}*	Background knowledge Primary school, Bachelor , Master etc.	Weight file must be corresponding with cfg file.
train images *{folder of train}*	Text Book , Information or other material to gain specific knowledge	
valid images *{folder of valid}*	Examination to validate knowledge score	
worker, batch-size, image *{numeric}*	Time to finish the studying of the course	Time to spend for learning per day. More time spend per day is shorter learning period of time
cfg *{filename.yaml}*	Brain cells volume, Age	Yolov7 has 37M cells

Custom AI model Training input compared to Human knowledge development

Custom AI model Training inputs	Human knowledge development	Comment
name *{ai model name}*	Name of specific knowledge or Subject name	
hyp *{hyper parameters}*	Tutor's technic to get learning curve real quick.	tutor's technic can not help improve your learning every case.
epoch	Exercise	More exercise gain more accuracy