Manual

This manual guides a reproduction of main experiment results. For data production, please refer to the manul.pdf shipped in the OSOCR-data repo.

I. **Setting up environments.**

Step 1A. Install a clean Manjaro Linux (Archlinux should do as well)

Step 1B. Install dependencies using scripts from:
https://github.com/lancercat/make_env

Step2. While waiting, grab the code and data.

1) Grab the data from kaggle (https://www.kaggle.com/vsdf2898kaggle/osocrtraining):

```
CVPR2016.zip,NIPS2014.zip,ssddata_2.zip,ctwcheval.zip,ssddata_1.tar.gz,ssddata.tar.gz
```

2) Unzip & remove the packages.

```
for i in $(ls | grep zip); do unzip $i; done
tar -xvf ssddata_1.tar.gz;
tar -xvf ssddata.tar.gz
mkdir ../packs; mv *.* ../packs;
```

Step 3C. Stop and check if you have all the following folders

Data:

```
[lasercat@lasercat-proj260 osocrdata]$ ls
artdb_seen  ctwcheval   CUTE80    dicts     IC13_1015    lsvtdb_seen      mlttrjp_hori  NIPS2014          rctwtrdb_seen
ctwch       ctwdb_seen  CVPR2016  IC03_867  IIIT5k_3000  mlttrchlat_seen  mlttrkr_hori  pami_ch_fsl_hwdb  SVT
```

Models:

```
[lasercat@lasercat-proj260 prfinal]$ ls
ablative  ctw_extra  hwdb  models  revision_extra
```

II. **Evaluate the Trained models.**

1. Zero-shot Chinese character recognition

CTW dataset (ZSL)

CTW			
# characters in training set			
500	1000	1500	2000
28.03	**49.00**	**58.37**	**64.03**

CTW dataset (OSTR)

	CTW			
#NIC	50	100	200	250
#NOC	450	400	300	250
A(NIC)	79.3	77.1	72.6	69.6
R(NOC)	73.3	54.7	37.7	31.5
P(NOC)	98.9	95.9	92.4	88.6
F(NOC)	84.2	69.7	53.5	46.5

basic_ctwch_CE_alter/testg2.py

```
test accuracy:
Accuracy: 0.280333, AR: 0.280333, CER: 0.719667, WER: 0.719667
97591
500 Done
```

```
Accuracy: 0.583753, AR: 0.583753, CER: 0.416247, WER: 0.416247
97591
1500 Done
```

```
Accuracy: 0.490035, AR: 0.490035, CER: 0.509965, WER: 0.509965
97591
1000 Done
```

```
Accuracy: 0.640387, AR: 0.640387, CER: 0.359613, WER: 0.359613
97591
2000 Done
```

basic_ctwch_CE_alter/testg2-rej.py

```
test rej accuracy:
KACR: 0.793193,URCL:0.733016, UPRE 0.989458, F 0.842147
97591
dictrej50.pt Done
```

```
test rej accuracy:
KACR: 0.771495,URCL:0.547913, UPRE 0.959210, F 0.697440
97591
dictrej100.pt Done
```

```
test rej accuracy:
KACR: 0.726157,URCL:0.377160, UPRE 0.924411, F 0.53573
97591
dictrej200.pt Done
```

```
test rej accuracy:
KACR: 0.696198,URCL:0.315216, UPRE 0.886503, F 0.465068
97591
dictrej250.pt Done
```

HWDB dataset (ZSL)
basic_hwdb_CE_alter/testg2.py

HWDB			
# characters in training set			
500	1000	1500	2000
47.92	**74.02**	**81.11**	85.72

```
Accuracy: 0.479248, AR: 0.479231, CER: 0.520769, WER: 0.520752
59777
500 Done
```

```
Accuracy: 0.740285, AR: 0.740285, CER: 0.259715, WER: 0.259715
59777
1000 Done
```

```
Accuracy: 0.811064, AR: 0.811064, CER: 0.188936, WER: 0.188936
59777
1500 Done
```

```
Accuracy: 0.857253, AR: 0.857253, CER: 0.142747, WER: 0.142747
59777
2000 Done
```

HWDB dataset(OSTR)
basic_hwdb_CE_alter/testg2-rej.py

	HWDB			
#NIC	100	200	400	500
#NOC	900	800	600	500
A(NIC)	93.5	93.9	91.0	90.0
R(NOC)	48.0	24.6	7.9	5.1
P(NOC)	99.7	99.5	97.9	96.7
F(NOC)	64.8	39.5	14.6	9.7

```
KACR: 0.935008,URCL:0.480737, UPRE 0.997340, F 0.648759
59777
dictrej100.pt Done
```

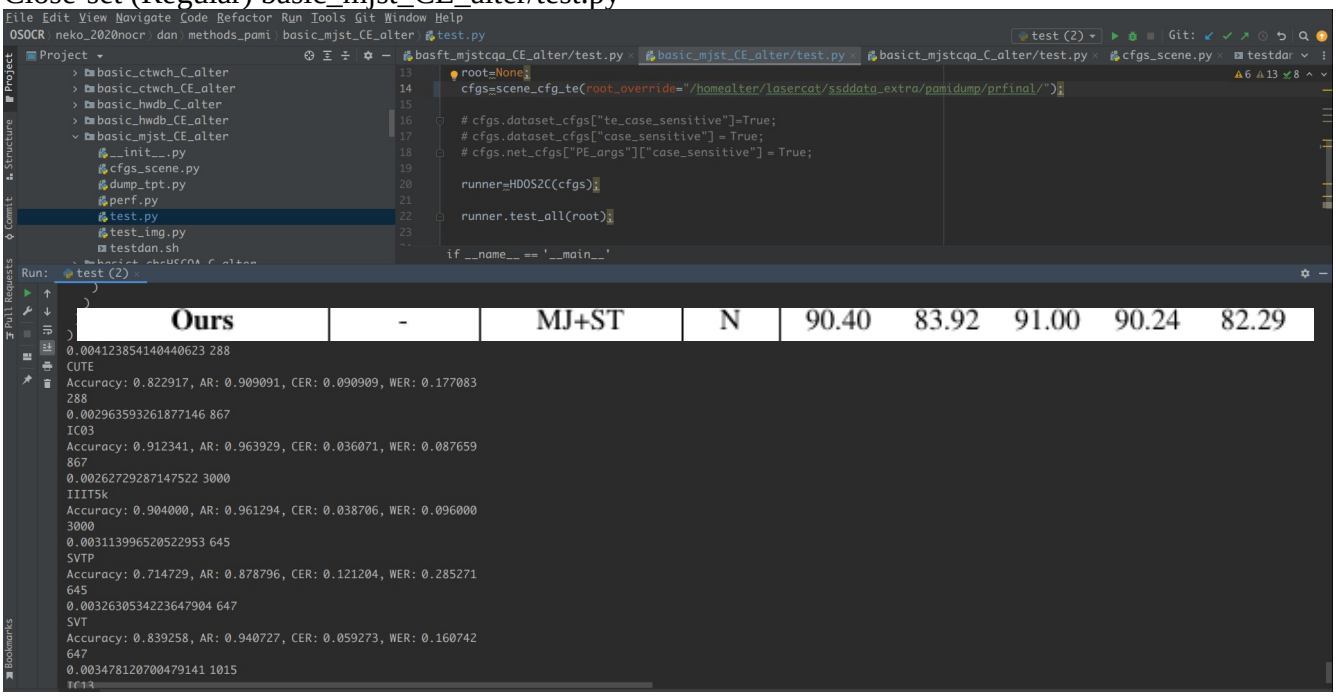
```
KACR: 0.939931,URCL:0.246738, UPRE 0.995697, F 0.395475
59777
dictrej200.pt Done
```

```
KACR: 0.910422,URCL:0.079130, UPRE 0.979634, F 0.146432
59777
dictrej400.pt Done
```

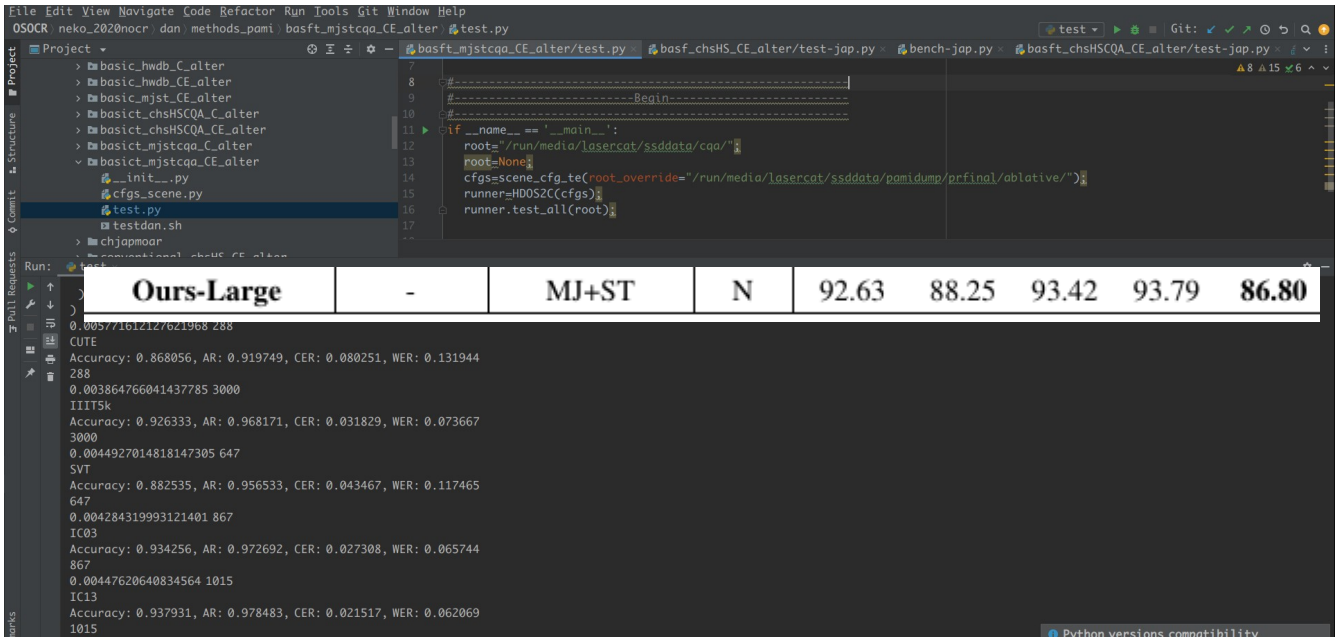
```
test rej accuracy:
KACR: 0.900278,URCL:0.051198, UPRE 0.967130, F 0.097248
59777
dictrej500.pt Done
```

2. Close-set text recognition

Close-set (Regular) basic_mjst_CE_alter/test.py



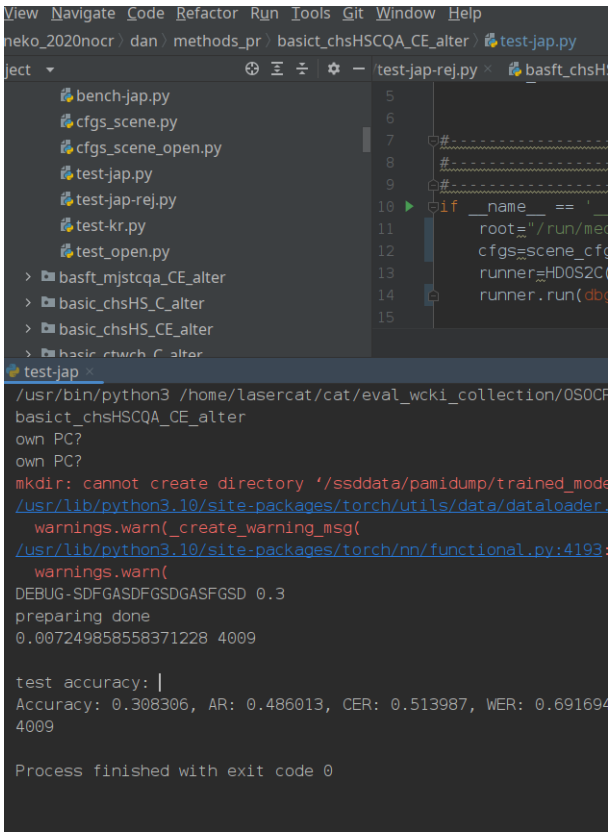
Close-set (Ours-large) basict_mjstcqa_CE_alter/test.py



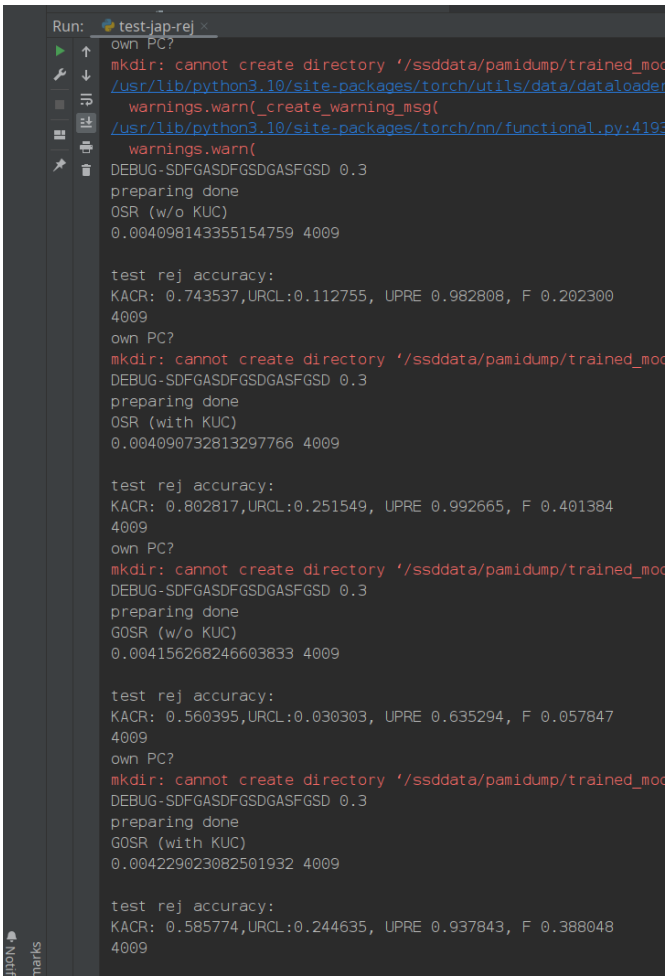
3. Open-set Text Recognition

Name	C^i_{test}	C^o_{test}	$\lvert C^i_{test} \rvert$	LA	R	P	F
GZSL	Unique Kanji, Shared Kanji, Kana, Latin	$\emptyset$	1460	30.83	-	-	-
OSR w/o SOC	Shared Kanji, Latin	Unique Kanji, Kana	849	74.35	11.27	98.28	20.23
OSR with SOC	Shared Kanji	Unique Kanji, Kana, Latin	787	80.28	25.15	99.26	40.13
GOSR	Shared Kanji, Unique Kanji, Latin	Kana	1301	56.03	3.03	63.52	5.78
OSTR	Shared Kanji, Unique Kanji	Kana, Latin	1239	58.57	24.46	93.78	38.80

GZSL
basict_chsHSCQA_CE_alter/test-jap.py



Others:
basict_chsHSCQA_CE_alter/test-jap-rej.py



Details:
chjapmoar/accr_folder.py (make sure you have results from basict_chsHSCQA_CE_alter/test-jap.py)

Name	Sample Requires	Sample Excludes	CA(%)	**LA(%)**
Shared Kanji	Shared Kanji	Unique Kanji, Kana	85.69	73.21
Unique Kanji	Unique Kanji	Kana	76.50	40.87
All Kanji	Unique Kanji or Shared Kanji	Kana	79.94	54.91
Kana	Hiragana or Katakana		25.10	0.72
All			54.03	30.83

```
Connected to pydev debugger (build 221.5591.52)
/run/media/lasercat/20615BC32265B955/prfinal/chs-japxl/
Accuracy: 0.308306, AR: 0.486013, CER: 0.513987, WER: 0.691694
Overall 0.5403113212380896 0.3083063108006984
/run/media/lasercat/20615BC32265B955/prfinal/chs-japxl/
Accuracy: 0.732161, AR: 0.856908, CER: 0.143092, WER: 0.267839
Seen 0.8112844997463888 0.7321613236814891
/run/media/lasercat/20615BC32265B955/prfinal/chs-japxl/
Accuracy: 0.408730, AR: 0.765065, CER: 0.234935, WER: 0.591270
Unique Kanji 0.7549325410039688 0.4087301587301587
/run/media/lasercat/20615BC32265B955/prfinal/chs-japxl/
Accuracy: 0.549169, AR: 0.799458, CER: 0.200542, WER: 0.450831
All Kanji 0.7794014876155192 0.5491692860350247
/run/media/lasercat/20615BC32265B955/prfinal/chs-japxl/
Accuracy: 0.007295, AR: 0.251016, CER: 0.748984, WER: 0.992705
Kana 0.24151569804923498 0.007295173961840628
Ours&54.03/30.83&&81.13/73.22&75.49/40.87&77.94/54.92&24.15/0.73\\

Process finished with exit code 0
```

III. Train your own models.

Overall, training model can be done by running the playdan.sh script in the method path. Here is a guide to train the OSOCR-Large model for open-set text recognition.

1. Setup paths in neko_sdk/root.py

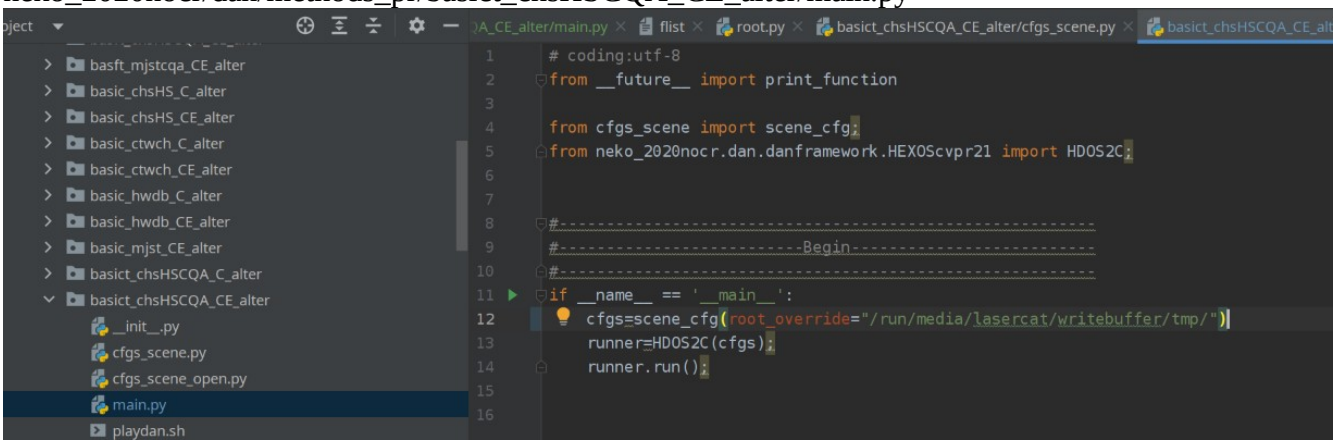
Point the return values of thhe find_data_root function

```
if(username!="prir1005"):
    print("own PC?")
    return os.path.join("/home",username,"ssddata");
```

to the folder containing the data you just dowloaded.

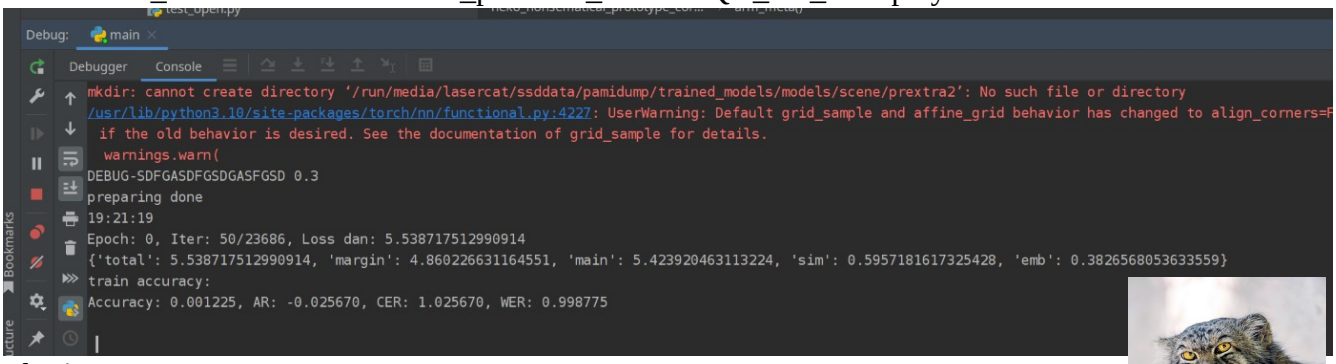
```
[lasercat@lasercat-proj260 osocrdata]$ ls
artdb_seen  ctwcheval   CUTE80    dicts     IC13_1015    lsvtdb_seen      mlttrjp_hori  NIPS2014          rctwtrdb_seen
ctwch       ctwdb_seen  CVPR2016  IC03_867  IIIT5k_3000  mlttrchlat_seen  mlttrkr_hori  pami_ch_fsl_hwdb  SVT
```

2. Set up model saving path with root_override in neko_2020nocr/dan/methods_pr/basict_chsHSCQA_CE_alter/main.py



3. Run the script

or run neko_2020nocr/dan/methods_pr/basict_chsHSCQA_CE_alter/playdan.sh



That's it.