

# MOOSE: Multi-Orientation Sharing Experts for Open-set Scene Text Recognition

This repository is the author's implementation of the paper MOOSE: Multi-Orientation Sharing Experts for Open-set Scene Text Recognition.

The repo implements a Multi-Orientation Sharing Experts framework that allows you to handle seen and unseen scene text written in various orientations.

## Getting Started

Implementation has been tested on a Lenovo P360 Ultra with an RTX4060Ti GPU connected through a Thunderbolt dock (TH3P4G2).

Software: Manjaro Linux, Python 3.11

## Environment

Some dependencies will need to be compiled, so you need to follow the guide below.

[https://github.com/lancercat/make\\_envNG](https://github.com/lancercat/make_envNG)

## Dataset

the training samples include English and Chinese word crops taken from ART [28], RCTW [29], CTW [30], LSVT [31], and the Latin and Chinese sections of the MLT [32] dataset. Note that the samples that include characters other than the 3755 Tier1 Chinese characters, 26 English letters, and the 10 digits (3791 classes in total) are excluded from the training set to avoid label leaking. The testing set includes the Japanese subsets of the MLT dataset.

You can download them from the following links (you need to download both datasets)

- <https://www.kaggle.com/datasets/object300/mose-extra/>
- <https://www.kaggle.com/vsdf2898kaggle/osocrtraining>

## Models

All models are released to the following repo

- All Models(45G): <https://www.kaggle.com/datasets/object300/moose-models-release>

## Logs

Our training logs are released with a visualizer.

- Logs: <https://www.kaggle.com/datasets/object300/log-moose-release>
- Log Visualizer: <https://github.com/lancercat/minijinja>

Note the log visualizer includes ssh related behavior,

as it is a part of a home-grown server management library,

used for autonomous monitoring, issuing commands to, and collecting results from a server fleet.

## Results & Manual

The Results and a detailed manual is included in the manul.pdf file.

(The filename is an intended pun, manuals, aka Pallas cats, are the oldest cats)

## Code

The methods are configured and can be launched from the `neko_2023_NGNW/project_moose/[methodname]` directory.

For training, launch `neko_2023_NGNW/project_moose/[methodname]/train_osr_hv_cat_wandb.py`

For testing, launch `neko_2023_NGNW/project_moose/[methodname]/eval_osr_hv.py`

For more details, please consult the manual.

## Naming

This section lists the correspondences between names in the paper and codenames in the repo

- Horizontal
  - 1 horizontal
  - hori\_sharebbn\_s\_05\_ld\_long

- Horizontal-MoE
  - 2 horizontal
  - hori\_sharebbn\_62\_ld\_long
- Single-Horizontal
  - 1 horizontal 1 vertical
  - sharebbn\_05\_ld\_long
- Rotated
  - 2 horizontal, but rotates vertical samples and pipe them to horizontal experts
  - sharebbn\_62RS\_ld\_long
- Share All
  - 2 horizontal 1 vertical
  - shareall\_62\_ld\_long
- Share None
  - 2 horizontal 1 vertical
  - sharenone\_62\_ld\_long
- M0oSE
  - 2 horizontal 1 vertical
  - sharebbn\_62\_ld\_long
- M0oSE-XL
  - 2 horizontal 1 vertical
  - yukon\_sharebbn\_62\_ld\_long\_XL

## Contact

- Chang lasercat@gmx.us, [chang.liu@ltu.se](mailto:chang.liu@ltu.se)
- Simon Corbill é [simon.corbille@associated.ltu.se](mailto:simon.corbille@associated.ltu.se)
- Elisa H. Barney Smith [elisa.barney@ltu.se](mailto:elisa.barney@ltu.se)