

Instructions - Group 6

University College London (UCL), WC1E 6BT, London, United-Kingdom

Experimental Setup Instructions

Note: All experiments were performed on UCL Computer Science machines.

1. Assuming the zip file is extracted, navigate to the working directory `src/`.
2. Activate the PyTorch conda environment following the instructions provided in the accompanying PDF.
3. Install the following additional packages using `pip install packagename`
 - **matplotlib** (Version used 3.8.4)
 - **timm** (Version used 0.9.16)
 - **wandb** (Version used 0.16.6)
4. Download and extract the COCO dataset.
5. Download and extract the PET dataset.
6. Search for the comment: `INSERT PATH HERE` and replace paths.
7. (Optional) Read the comments at the beginning of the `main.py` file for additional information.

Logs, Visualisations, and Models:

To facilitate a deeper understanding of the project, we provide access to sample log files, visualisations, and models.

Each file is identified by a timestamp, serving as a unique identifier. Log files, models, and images bearing the same timestamp originate from a single experiment run.

We have curated a selection of log files, images, and models, representing three distinct experimental sessions. These resources are conveniently located within the `samples/` directory.

Sample Experiment Runs

Additionally, we have abstracted some of the inner workings of the framework such that the marker/reviewer may verify that the results are authentic. We have added "demo" usage (i.e. demonstrations) which run in various unique configurations, which do NOT require wandb integration, these demos produce logs, and run for only 1 epoch.

```
# Demo Usage:
# python main.py --demo 1 | noise = vanilla, mask_sampling = random
# python main.py --demo 2 | noise = vanilla, mask_sampling = block
# python main.py --demo 3 | noise = gaussian, mask_sampling = grid
# python main.py --demo 4 | BASELINE FULLY SUPERVISED
```

Using the Experiment Framework (FULL):

To leverage the experiment framework effectively, it is imperative to ensure all prerequisites are met, including the integration of Weights and Biases. If using bash/zsh, you can export your `WANDB_API_KEY` as follows:

```
export WANDB_API_KEY="your_key_here"
```

For detailed instructions, please refer to the documentation provided by Weights and Biases.

Below are some examples showcasing the usage of the framework, which represent a subset of available experiments:

```
# For baseline
python main.py --baseline 1 --epochs 30

# Typical usage 1
python main.py --noise gaussian --mask_sampling random \
--fine_tune_size 2 --epochs 30

# Typical usage 2
python main.py --noise vanilla --mask_sampling block \
--block_mask_ratio 0.5 --fine_tune_size 1 --epochs 30
```

Additionally, the experiment framework offers several arguments to support reproducibility and customization. Below is an exhaustive list:

Basic Parameters:

- `-root_dir`: Root directory for the project. Defaults to the current directory.
- `-wandb_project`: Project name for Weights & Biases logging. Set to an empty string to disable logging. Default: "comp0197-ADL-CW2-MAE".
- `-seed`: Random seed for reproducibility. Default: 42.
- `-num_workers`: Number of worker threads for loading data. Default: 4.

Dataset Parameters:

- `-batch_size`: Number of samples in each batch of data. Default: 128.
- `-image_size`: Size (height and width) of the input images. Default: 224 pixels.
- `-patch_size`: Size of the patches sampled from each image. Default: 16.
- `-data_path`: Path to the dataset. Default: "./data".
- `-output_dir`: Directory to save the output results. Leave empty to disable saving. Default: "./output_dir".
- `-mask_ratio`: Fraction of image patches to be masked during training. Default: 0.75 (75

Optimizer Parameters:

- `-weight_decay`: Weight decay (L2 penalty) for regularization. Default: 0.0001.

- **-lr**: Learning rate for the optimizer. Default: 0.005.

Advanced Settings:

- **-noise**: Type of noise to apply. Options: "vanilla", "gaussian", "salt". Default: "vanilla".
- **-baseline**: Flag to run the baseline model. 0 for no, 1 for yes. Default: 0.
- **-dataset**: Dataset to be used. Currently, only 'coco' is supported. Default: "coco".
- **-mask_sampling**: Strategy for sampling masks. Options: "random", "grid", "block", "semantic". Default: "random".
- **-fine_tune_size**: Proportion of the dataset to use for fine-tuning. Options: 1, 0.5, 0.1. Default: 1.
- **-out_dir**: Alternative output directory. Default: "" (empty).
- **-block_mask_ratio**: Masking ratio for block mask sampling. Default: 0.5.
- **-debug**: Enable debug mode. 0 for no, 1 for yes. Default: 0.

Training Parameters:

- **-epochs**: Number of training epochs. Default: 10.

Encoder and Decoder Parameters:

- **-enc_projection_dim**: Dimension of the encoder's projection space. Default: 128.
- **-dec_projection_dim**: Dimension of the decoder's projection space. Default: 64.
- **-enc_num_heads**: Number of attention heads in the encoder. Default: 4.
- **-dec_num_heads**: Number of attention heads in the decoder. Default: 4.
- **-enc_layers**: Number of layers in the encoder. Default: 6.
- **-dec_layers**: Number of layers in the decoder. Default: 2.

For a comprehensive understanding of these parameters, please refer to the `get_args_parser()` function in `parse_and_log.py`.