

Supplementary file

Hyperparameter details for the DeepLabV3 ResNet101 segmentation model:

Training Configuration Parameters:

1. Dataset Specifications
Dataset format: COCO-style annotation format (dataset = 'xxx_coco_fmt')
Data storage directory: Custom path defined by my_dir variable (data_path = my_dir)
2. Model Architecture
Selected model: DeepLabV3 with ResNet101 backbone (model = 'DeepLabV3 ResNet101')
Pretrained weights: Enabled (pretrained = True)
Auxiliary loss head: Disabled (aux_loss = False)
3. Training Hyperparameters
Learning rate: 0.001 (lr = 0.001)
Optimization method: Stochastic Gradient Descent (optimizer = 'sgd')
Momentum value: 0.9 (momentum = 0.9)
Weight decay (L2 regularization): 1e-4 (weight_decay = 1e-4)
Training epochs: 40 (epochs = 40)
4. Data Loading Configuration
Batch sizes:
Training batch: 4 samples (batch_size = 4)
Validation batch: 1 sample (val_batch_size = 1)
Data loading workers: 4 parallel processes (workers = 4)
Input image dimensions:
Base size: 480 pixels (base_size = 480)
Crop size: 480 pixels (crop_size = 480)
5. System Configuration
Distributed training:
World size: 1 (single process) (world_size = 1)
Distributed URL: Environment variable-based (dist_url = 'env://')
GPU memory allocation: [Not explicitly specified]
6. Operational Parameters
Model saving:
Output directory: Current folder (save_dir = '.')
Epoch-level saving: Disabled (save_per_epoch = False)
Training resumption: No checkpoint path specified (resume = "")
Logging frequency: Every iteration (print_freq = 1)
Test-only mode: Disabled (test_only = False)

7. Loss function: Cross-Entropy Loss (CE)

$$CE = -\frac{1}{N} \sum_{i=1}^N (\text{ground truth label})_i \cdot \log(\text{predicted probability}_i)$$

8. Additional Notes
Downsampling ratio: 1 (original resolution maintained) (downsample_ratio = 1)
Custom attributes: None defined (attr = {})

Hyperparameter details for the machine learning models:

- **Logistic Regression (LR)**
Regularization penalty is disabled (penalty='none')
Maximum training iterations set to 100 (max_iter=100)
- **Support Vector Machine (SVM)**
Linear kernel function is used (kernel='linear')
Probability estimation is enabled (probability=True)
Maximum solver iterations limited to 100 (max_iter=100)
- **Random Forest**
Number of decision trees set to 5 (n_estimators=5)
Maximum tree depth restricted to 3 levels (max_depth=3)
Minimum 4 samples required to split a node (min_samples_split=4)
Random seed fixed at 0 (random_state=0)

- **XGBoost**
 Boosting rounds limited to 6 (`n_estimators=6`)
 Binary logistic regression objective function
 (`objective='binary:logistic'`)
 Tree depth capped at 3 levels (`max_depth=3`)
 Minimum sum of instance weight in leaf nodes
 set to 0.2 (`min_child_weight=0.2`)
 Label encoder usage is disabled
 (`use_label_encoder=False`)
 Evaluation metric uses classification error rate
 (`eval_metric='error'`)
- **LightGBM**
 Boosting iterations limited to 2 (`n_estimators=2`)
 Tree depth constrained to 1 level (`max_depth=1`)
 Minimum sum of instance weight in leaves set
 to 0.5 (`min_child_weight=0.5`)
- **Extra Trees**
 Ensemble contains 60 decision trees
 (`n_estimators=60`)
 Maximum tree depth limited to 5 levels
 (`max_depth=5`)
 Minimum 2 samples required for node splitting
 (`min_samples_split=2`)
 Random seed fixed at 0 (`random_state=0`)
- **Gradient Boosting**
 Boosting stages limited to 10 (`n_estimators=10`)
 Random seed fixed at 0 (`random_state=0`)
- **AdaBoost**
 Number of weak learners set to 10
 (`n_estimators=10`)
 Random seed fixed at 0 (`random_state=0`)
- **Multilayer Perceptron (MLP)**
 Network architecture contains four hidden lay-
 ers with 61, 128, 64, and 32 neurons respec-
 tively (`hidden_layer_sizes = (61,128,64,32)`)
 Training capped at 300 epochs (`max_iter=300`)
 Stochastic gradient descent optimizer is used
 (`solver='sgd'`)
 Random seed fixed at 0 (`random_state=0`)

Training details of the SVM model:

Number of Epochs:

SVM Context: SVMs don't use "epochs" tradi-
 tionally (optimization stops at convergence).

Equivalent control via:

- `max_iter`: Maximum optimizer iterations
 (default: -1, no limit).
- `tol`: Tolerance for early stopping (default:
 1e-3).

Optimizer:

SVM Implementation:

- Algorithm: Uses Sequential Minimal Op-
 timization (SMO) by default (no explicit
 optimizer choice).
- Workarounds: For custom solvers (e.g.,
 SGD-based SVM), libraries like `sklearn`.
`svm.LinearSVC`
- support loss='hinge' with SGD optimizer.

Loss Function:

Default: Hinge Loss (for classification):

$$L = \max(0, 1 - y_i (w^T X_i + b))$$

Options:

- `hinge`: Standard SVM loss (max-margin).
- `squared_hinge`: Squared hinge loss (e.g.,
 in `LinearSVC`).
- `epsilon_insensitive`: Used in regression
 (SVR).

Hyperparameter details for the DTL models:

Hyperparameter Configuration

- Data Pipeline
 Training data path: `train_f`
 Validation data path: `val_f`
 Label file path: `labels_f`
 Data pattern specification: `data_pattern`
 No maximum sample limit for train-
 ing/validation (`max2use=None`,
`val_max2use=None`)
 Batch balancing disabled
 (`batch_balance=False`)
 Normalization method: ImageNet standard-
 ization (`normalize_method='imagenet'`)
- **Training Protocol**
 Batch size: 32 samples (`batch_size=32`)

- Training epochs: 40 (epochs=40)
 Initial learning rate: 0.01 (init_lr=0.01)
 Optimizer: Stochastic Gradient Descent
 (optimizer='sgd')
 No retraining path specified (retrain=None)
- System Configuration
 - GPU device ID 0 utilized (gpus=[0])
 - Parallel processing threads: 0 (disabled) (j=0)
 - Distributed training: Not configured (implied by gpus=[0])
 - **Operational Parameters**
 - Model output directory: Current folder (model_root='.')
 - No timestamp in output filenames (add_date=False)
 - Training starts from iteration 0 (iters_start=0)
 - Logging frequency: Every iteration (iters_verbose=1)
 - Epoch-level model saving disabled (save_per_epoch=False)
 - **Execution Command**
 - Configuration packaged into Args namedtuple

Main classifier function called via
 clf_main(Args(**params))

Training details of the DenseNet201 model:

Number of Epochs: 40

Optimizer:

Stochastic Gradient Descent with Momentum:

- lr=0.1, momentum=0.9 (common for ImageNet training).
- Requires learning rate scheduling (e.g., reduce by 10x at 30/60/90 epochs).

Loss Function:

Binary Cross-Entropy with Logits (BCE With Logits Loss)

$$L = - [y \log (\sigma(x)) + (1-y) \log (1-\sigma(x))]$$

σ = Sigmoid activation

Copyright: the Author(s), 2026. Licensee Mattioli 1885, Fidenza, Italy. This is an open-access article distributed under the terms of the Creative Commons Attribution NonCommercial License (CC BY-NC-4.0).

Disclaimer/Publisher's Note: The statements, opinions and data contained in this article are solely those of the author(s) and contributor(s) and do not necessarily reflect those of their affiliated organizations, the publisher, the editors or the reviewers. The publisher and the editors disclaim any responsibility for injury to people or property resulting from any ideas, methods, instructions or products mentioned in the content. Any product that may be evaluated in this article, or claim made by its manufacturer, is not guaranteed or endorsed by the publisher.