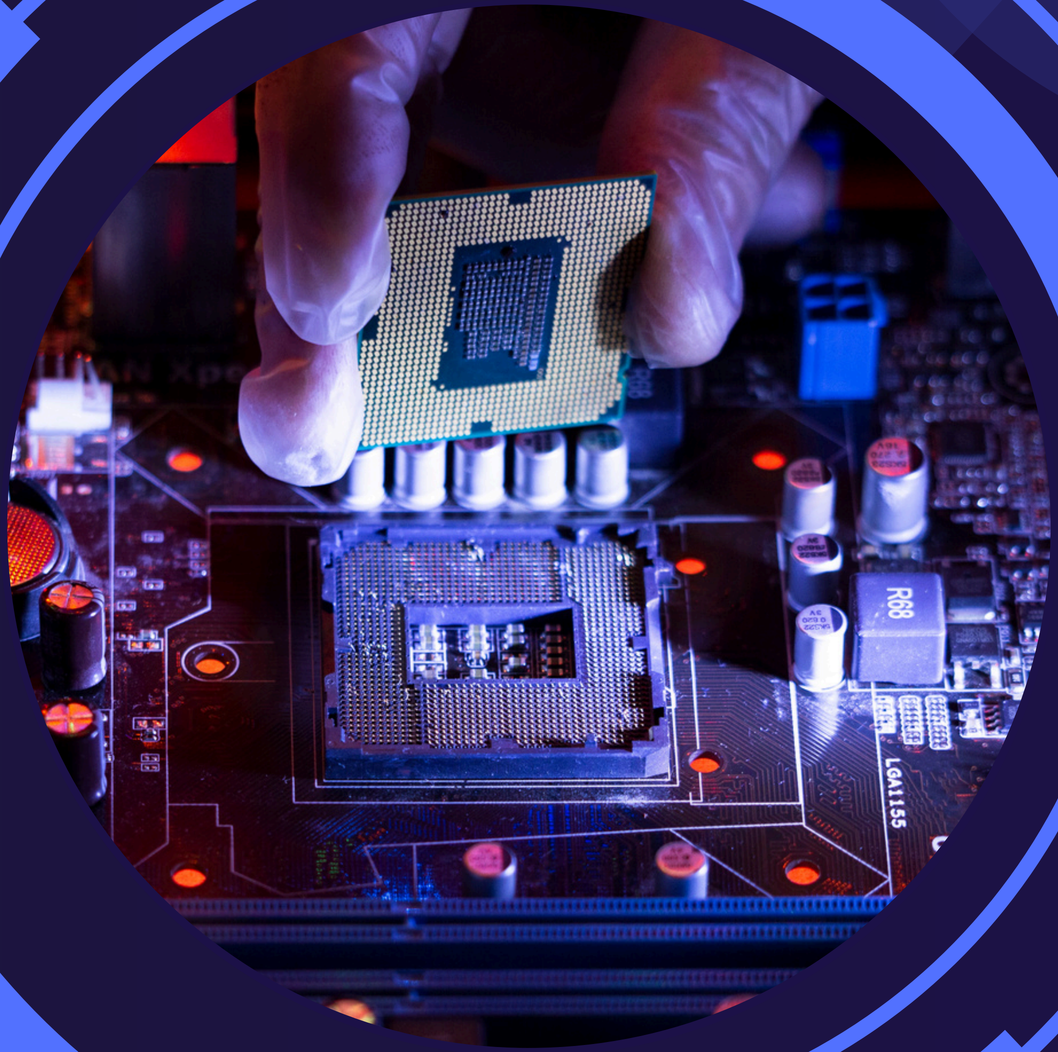
CSE475

Rice Grain Detection using ML

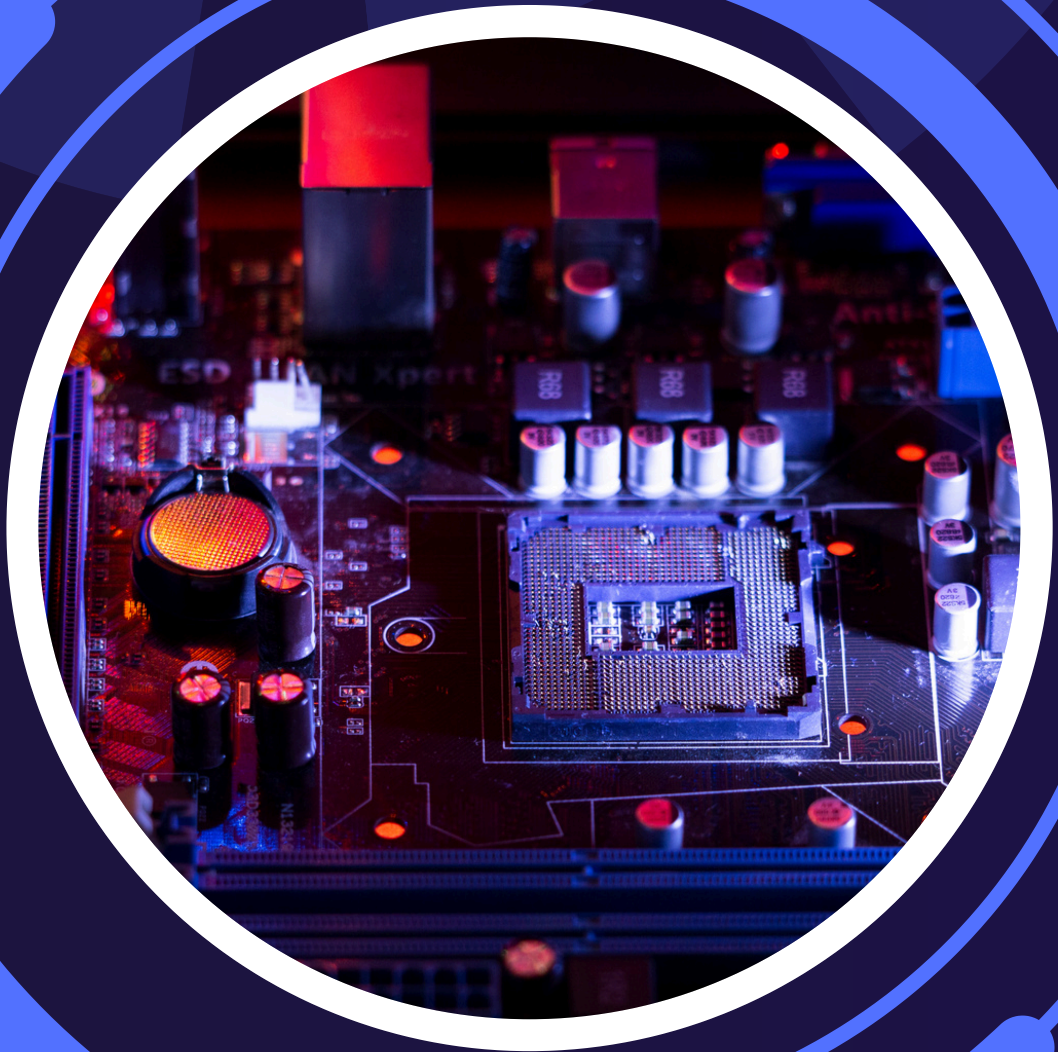


Submitted to:
Dr. Raihan Ul Islam
Associate Professor
Department of Computer Science & Engineering

Details About Our Dataset

The name of An Extensive Image Dataset for Classifying Rice Varieties in Bangladesh

The Original Image file Dataset being used in this project. It has 38 classes , and each class has exact 500 images. So Data is balanced. We dont to need apply data preprocessing since every class is balanced



Problem statement

Task: Multiclass image classification of rice cultivars/varieties.
Dataset: riceds-original/Original with folder-per-class layout.
Classes: 38 total (e.g., BRRI67, BRRI74, Binadhan10 … Binadhan26, BD30…), stratified 80/20 train/test split (15,200 / 3,800 images).
Metric focus: Top-1 accuracy on the held-out 20% test set, plus detailed per-class precision/recall/F1, confusion matrices, ROC/AUC.
Result (best epoch): ~93.76% test accuracy with EMA + TTA.

Previous work related information

Name of the Dataset	Models Used	Accuracy
RiceSEG Dataset (self supervised based on this dataset)	U-Net, U-Net++, FPN, PSPNet, LinkNet, MA-Net, DeepLabV3, DeepLabV3+, Transformer-based segmentation models (e.g., SegFormer	77.06%
Synthetic seed image dataset (self supervised based on another dataset)	U-Net, U-Net++, FPN, PSPNet, LinkNet, MA-Net, DeepLabV3, DeepLabV3+, Transformer-based segmentation models (e.g., SegFormer	77% (MoCo + 5% labels), ~90%

Our work results

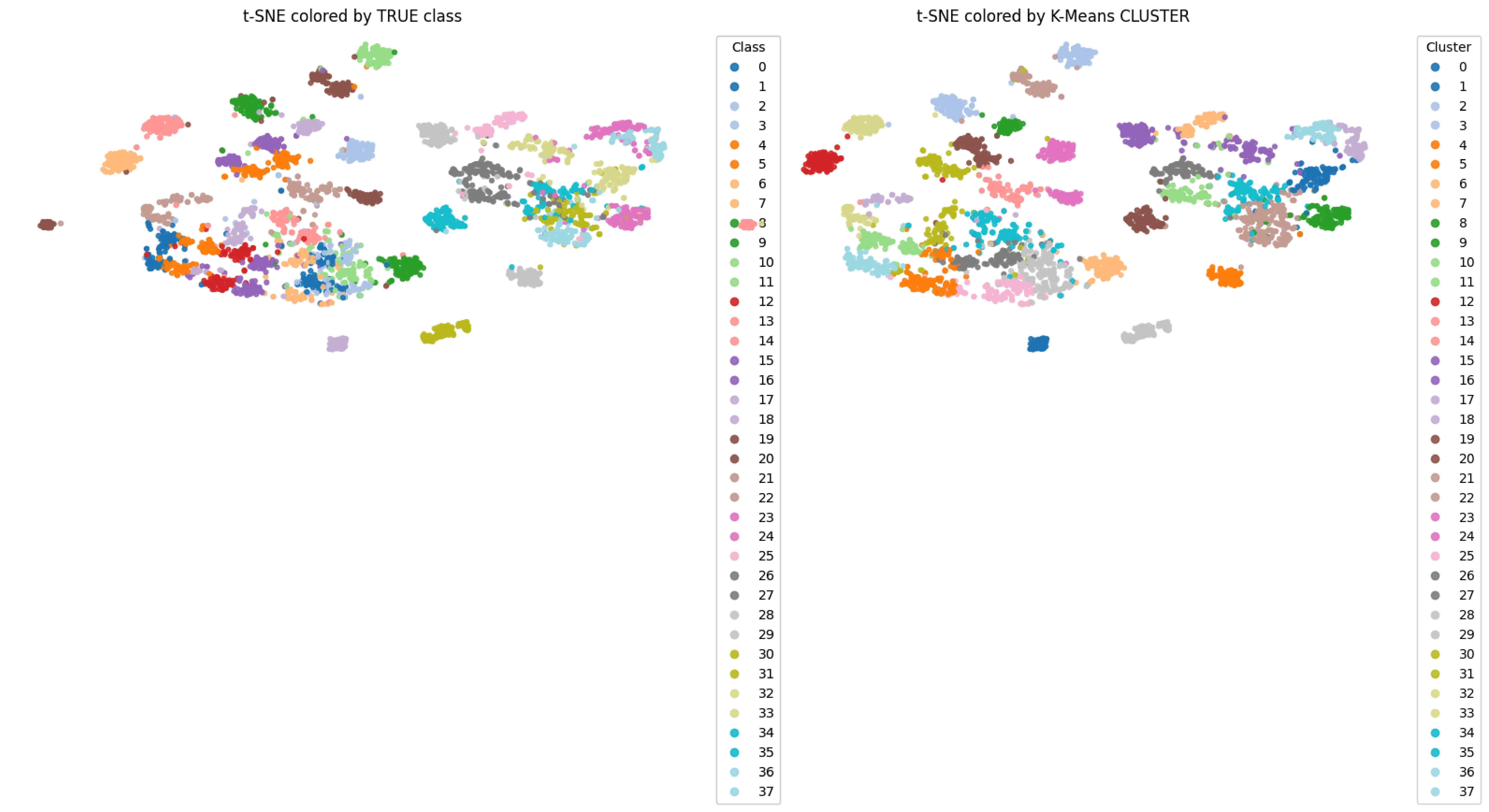
Models Used	Train Test Valdation Ratio	Accuracy
SimCLR	80-20-0	93.76%
SimCLR	80-10-10	93.63%
SimCLR	70-20-10	93.03%
Byol	80-20-0	93.45%
Byol	80-10-10	Code Ready Still Running
Byol	70-20-10	Code Ready, Still Running
MoCo	80-20-0	Code Ready,Still Running
MoCo	80-10-10	Code Ready, Still Running
MoCo	70-20-10	Code Ready,Still Running

Prediction | True: Binadhan21

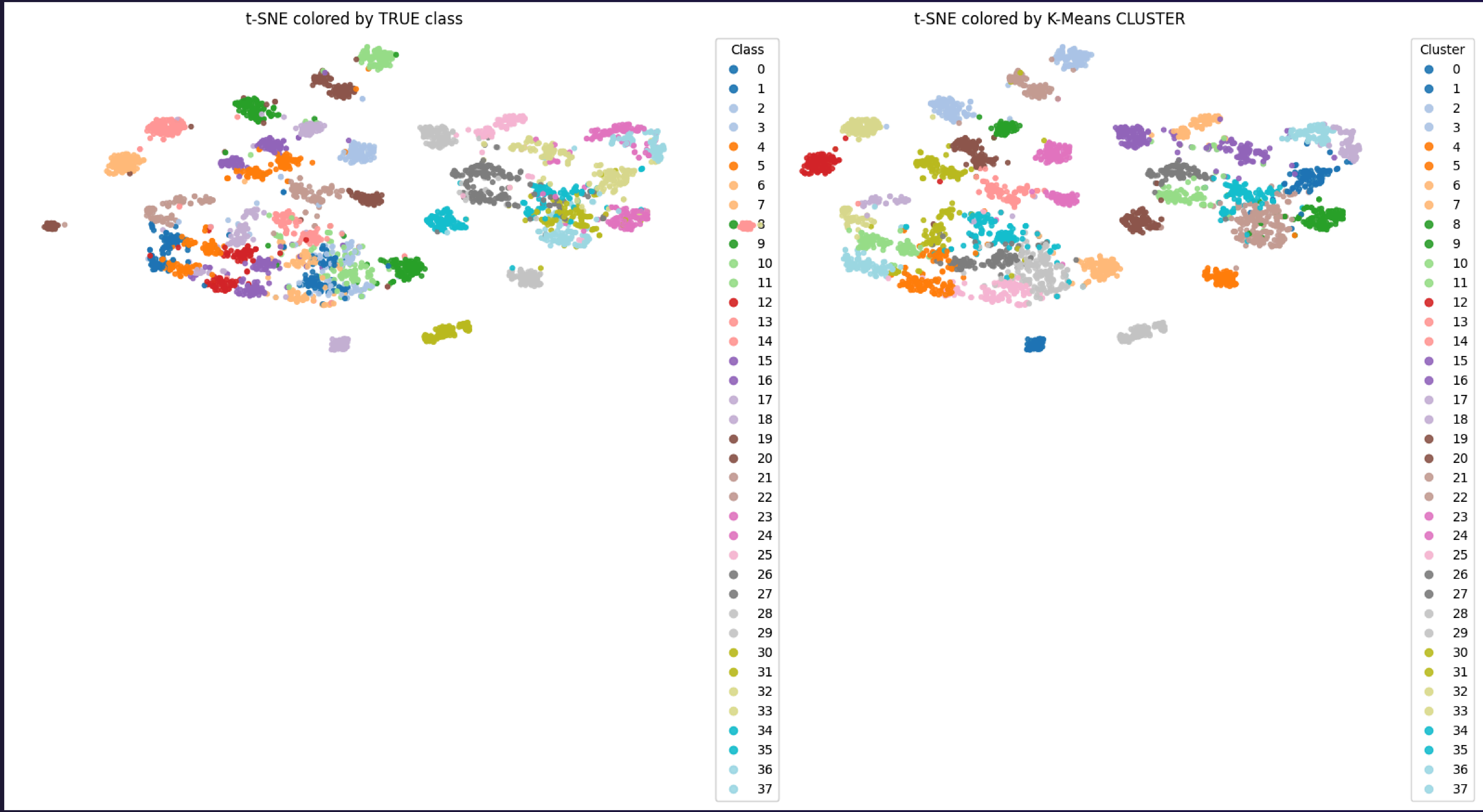
```
Top-k probabilities:
          Binadhan21: 0.8692
          Binadhan24: 0.0648
          Binadhan14: 0.0427
          Binadhan25: 0.0089
           Binadhan7: 0.0075
```

Image 1: Single Prediction

- **Shows a rice grain image classified as Binadhan21.**



t-SNE colored by TRUE class
Class
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
t-SNE colored by K-Means CLUSTER
Cluster
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37


t-SNE colored by TRUE class
Class
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
t-SNE colored by K-Means CLUSTER
Cluster
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37

Left Plot: t-SNE Colored by TRUE Class

- Each dot = one rice grain image represented in the learned feature space.
- Each color = a ground truth rice variety label (e.g., Binadhan21, Binadhan24, etc.).
- Observations:
    - Some classes form tight, well-separated clusters, showing that the model can extract highly discriminative features for those rice types.
    - A few classes have overlapping regions, meaning their features are harder to distinguish (likely due to visual similarity of grains).
    - The variety of colors across the space indicates that the dataset contains 38 unique classes (0–37), each with its own distribution.

Right Plot: t-SNE Colored by K-Means Clusters

Same embeddings, but now colored by K-Means clustering results instead of ground truth.

Each color here = an unsupervised cluster assignment.

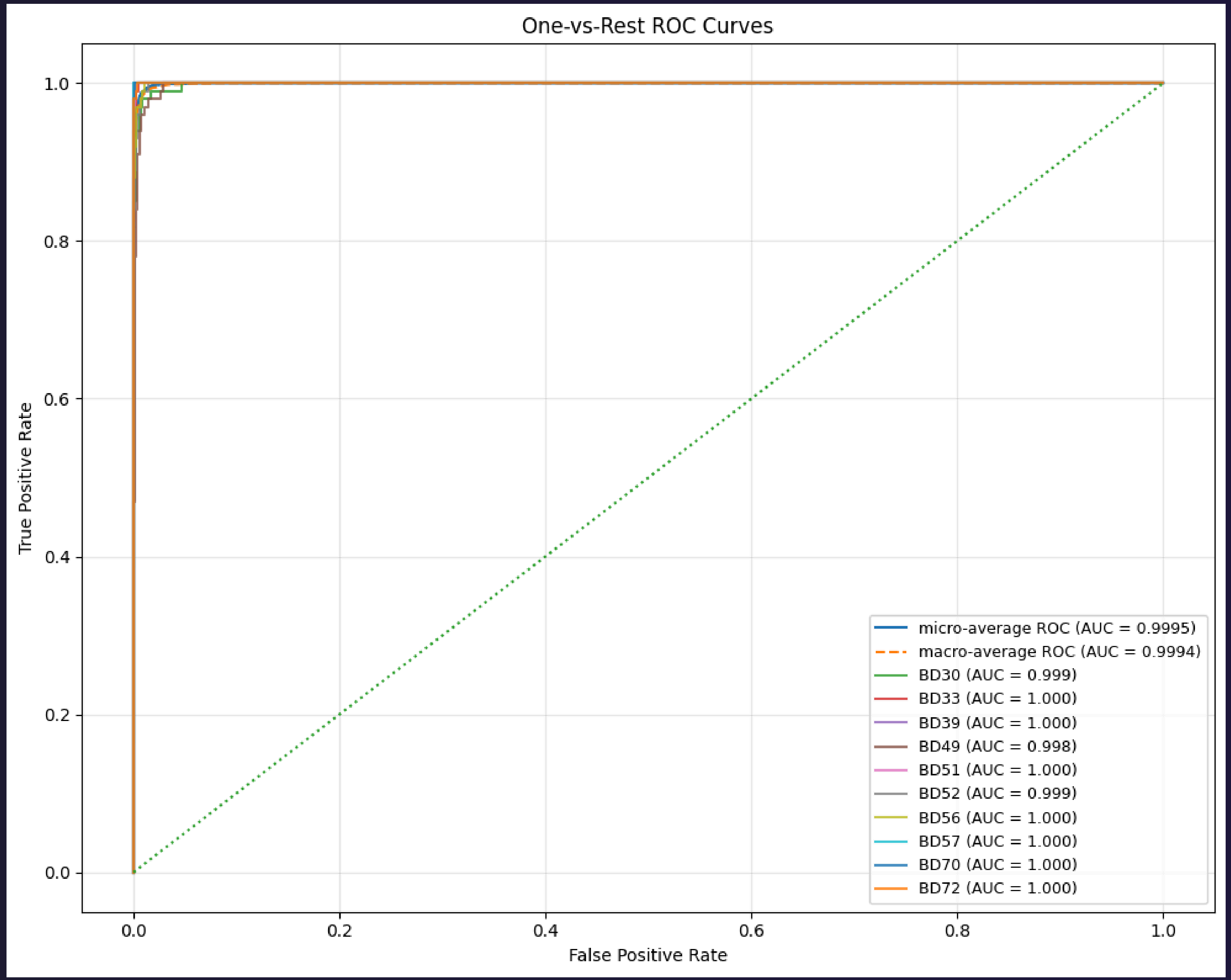
Observations:

Many clusters align fairly well with the true classes, confirming that the feature space captures meaningful separation.

Some clusters split what is actually one true class (over-segmentation). This happens when K-Means creates more clusters than necessary.

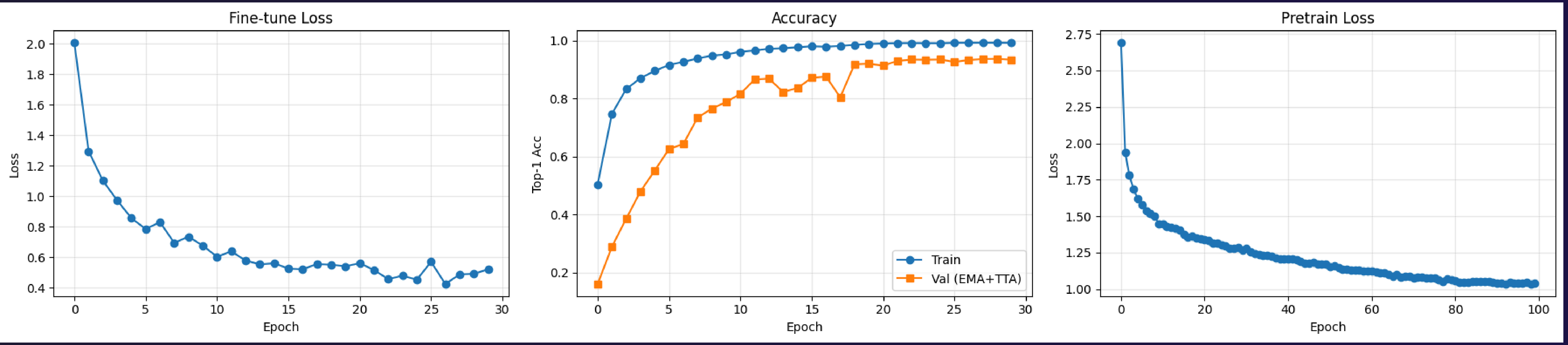
In other cases, multiple true classes fall into the same cluster (under-segmentation), showing that the visual features of those classes are too similar.

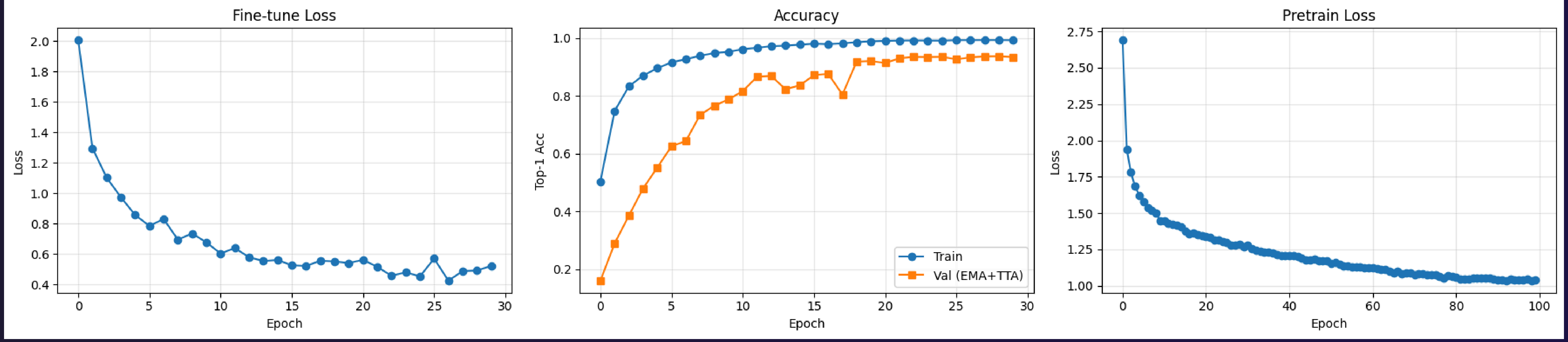
The spread of clusters shows that K-Means is capturing local neighborhood structures in the embedding space, but not perfectly aligned with actual class labels.



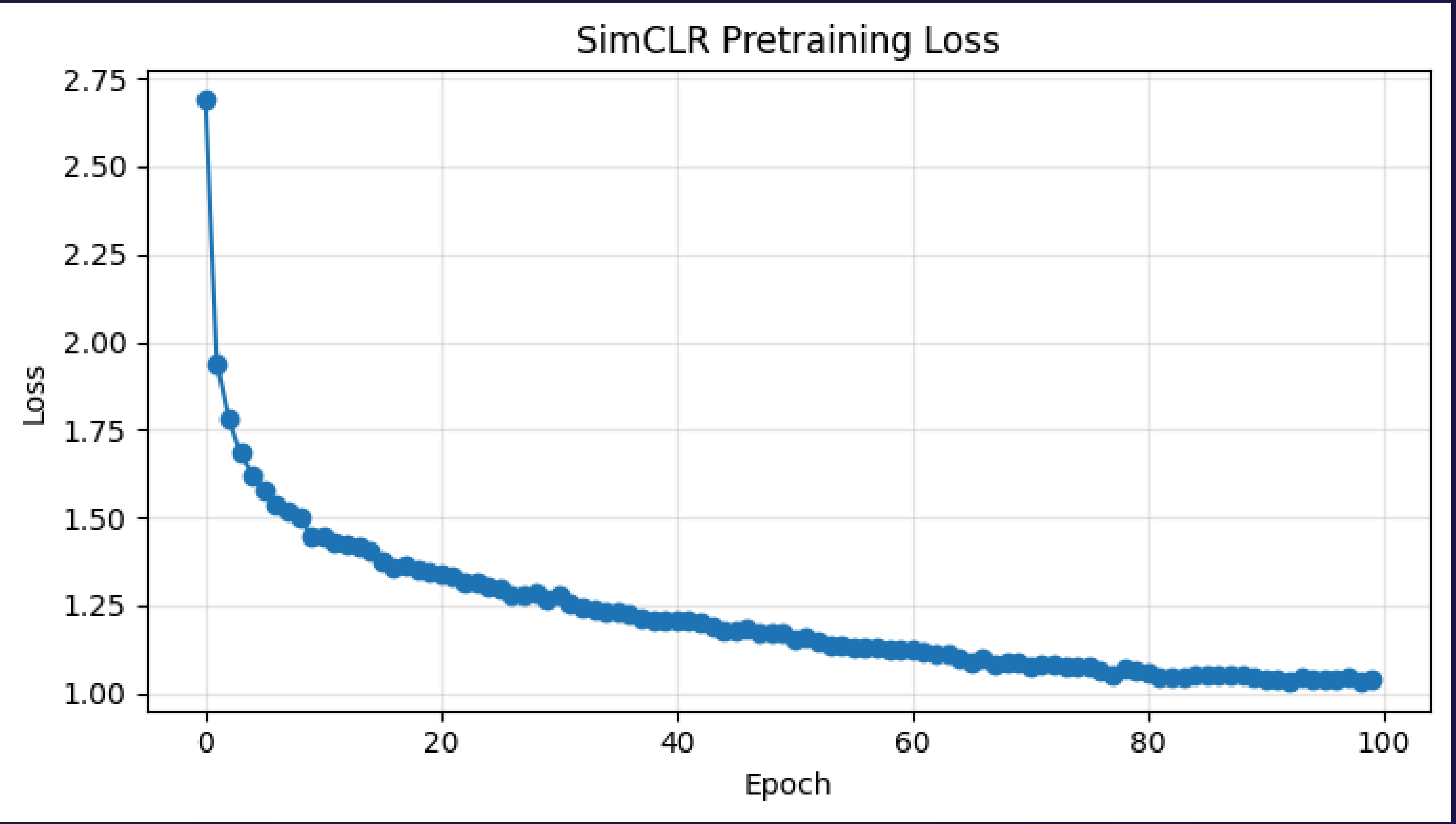
ROC Curves

- Multi-class One-vs-Rest ROC curves.
- Micro-average AUC = 0.9995, Macro-average AUC = 0.9994.
- Most individual classes (BD30, BD33, BD39, BD51, etc.) have AUC close to 1.0, meaning excellent classification performance.


Fine-tune Loss
Loss
Epoch
Accuracy
Top-1 Acc
Epoch
Train
Val (EMA+TTA)
Pretrain Loss
Loss
Epoch


Fine-tune Loss
Loss
Epoch
Accuracy
Top-1 Acc
Epoch
Train
Val (EMA+TTA)
Pretrain Loss
Loss
Epoch

- Fine-tune Loss (left): Loss decreases steadily, showing effective learning.
- Accuracy (middle): Training accuracy approaches ~99%, validation accuracy stabilizes around ~90–95%.
- Pretrain Loss (right): Loss decreases consistently during pretraining, indicating good feature learning.


SimCLR Pretraining Loss
Loss
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0
20
40
60
80
100
Epoch

SimCLR Pretraining Loss

Loss curve for SimCLR contrastive pretraining. Starts around 2.7 and drops close to 1.0 by epoch 100.

Shows the representation learning quality improves significantly with training.

Backbone

Architecture: ResNet-50 from torchvision.models.

Initialization: ImageNet weights (ResNet50_Weights.IMAGENET1K_V2) when USE_IMAGENET_WEIGHTS=True.

Feature output: Global pooled 2048-D vector (the model is used up to avgpool; the FC layer is dropped).

Evaluation & diagnostics

- Per-epoch eval: Accuracy on the held-out test split with EMA + TTA.
- Final report: classification_report, overall/macro/weighted metrics; confusion matrix (counts + normalized).
- Probabilities: Softmax probabilities collected for ROC/AUC.
- ROC/AUC: One-vs-rest per class + micro and macro AUC.
- Feature quality checks:
  - K-NN on normalized encoder features (optional, gated by ENHANCED).
  - Clustering: K-Means on L2-normalized features → ARI, NMI, Silhouette.
  - t-SNE visualization (true labels vs clusters).
  - Grad-CAM: Per-class overlays (both confidently correct and incorrect examples) from layer4 (last ResNet block) to inspect attention.

Key toggles (for controlled experiments)

ENHANCED: master flag (enables warmup, class-balanced sampler, BN tracking in EMA, KNN eval).
USE_LIGHTLY_SIMCLR: choose Lightly’s SimCLR stack vs a native SimCLR implementation.
USE_IMAGENET_WEIGHTS: turn on/off ImageNet init for the encoder.
USE_EMA, EMA_DECAY: EMA of weights.
USE_MIXUP, MIXUP_ALPHA: MixUp regularization.
USE_TTA: test-time horizontal flip averaging.

Why this stack fits the problem

Limited domain-specific labels per class benefit from SSL pretraining (SimCLR) to learn crop-invariant, color/texture-aware features from the same data distribution.

No-freeze fine-tuning with a lower LR on the backbone preserves useful SSL/Imagenet features while adapting to cultivar traits.

MixUp + EMA + TTA improve generalization and calibration—important for 38 subtle classes.

Rich diagnostics (ROC/AUC, Grad-CAM, clustering) help validate that the model learns biologically plausible cues rather than spurious artifacts.

Upstream (Self-Supervised Pretraining)

Purpose: learn generic, label-free visual features on your rice images.
Data views: Two-view SimCLR pipeline using lightly.transforms.SimCLRTransform (224×224, blur) → two crops per image.
Model: ResNet-50 encoder (ImageNet init on), plus a projection head (SimCLRProjectionHead 2048→512→128) used only for the contrastive loss.
Loss: NT-Xent (InfoNCE) with temperature = 0.2.
Optim & schedule: AdamW (lr 3e-4, wd 1e-6) with cosine schedule (per-iteration stepping; optional warmup off by default).
Precision: Mixed precision (AMP) + GradScaler.
Duration: 100 epochs (you note 200–400 would be better if possible).
Artifact: Save encoder weights to simclr_encoder.pth.
(Important: the projection head is for training only and is not used later.)
Mental model: upstream teaches the backbone to be invariant to crops, color jitter, blur, etc., producing robust 2048-D features.

Downstream (Supervised Fine-Tuning + Evaluation)

Purpose: adapt those features to your 38-class rice variety classification task and evaluate.

Head: Linear classifier 2048→38.

Loss: Soft cross-entropy (works with MixUp targets + optional label smoothing).

Optim & LRs: AdamW with two LR groups

– Backbone 1e-4 (small)

– Head 1e-3 (bigger)

– Weight decay 1e-4

– Cosine schedule over 30 epochs (optional warmup if enabled)

Regularization:

– MixUp alpha=0.2 on images and soft labels

– Grad clipping at 5.0

– (Optional) class-balanced sampler

Stability / Generalization:
– EMA of weights (decay 0.999); swap in EMA weights for eval, then restore
– TTA: average logits from original + horizontal-flip
Precision: Mixed precision (AMP) during fine-tune.
Evaluation & analysis:
– Per-epoch val accuracy (with EMA+TTA)
– Final classification report, confusion matrices (counts & normalized)
– ROC/AUC (one-vs-rest, micro, macro)
– Feature probes: optional k-NN, K-Means (ARI/NMI/Silhouette), t-SNE plots
– Grad-CAM overlays (correct & incorrect exemplars) for interpretability
Outcome (your run): best Top-1 = 93.76% on the 20% held-out test split.