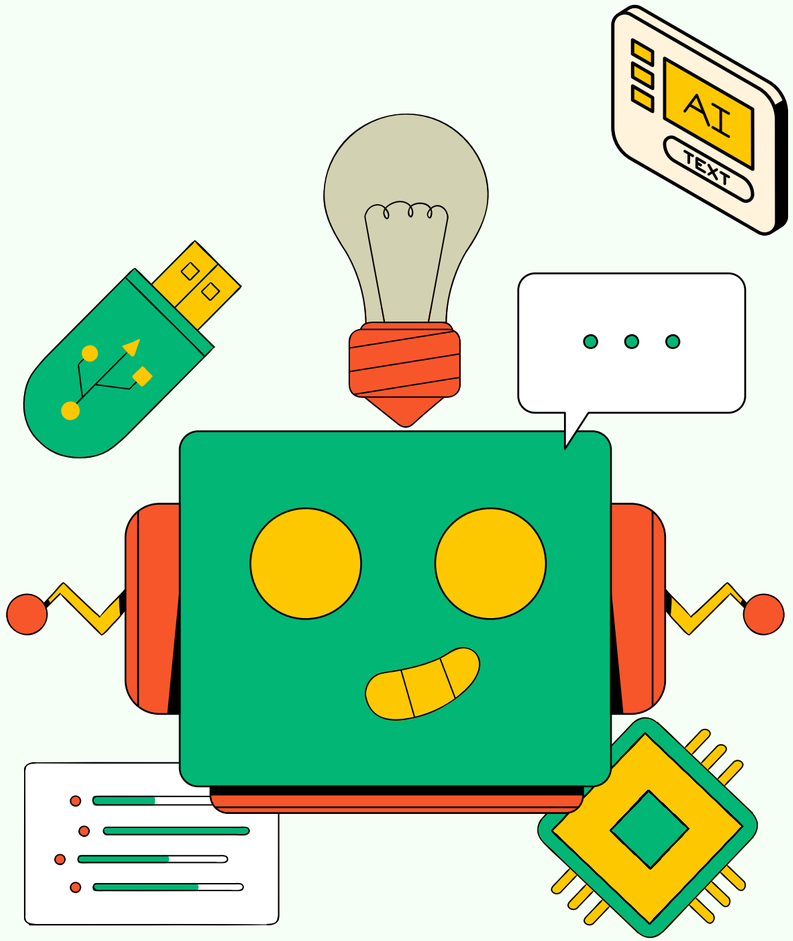
COS30082 APPLIED MACHINE LEARNING

WE LEARN FOR THE FUTURE



CROSS-DOMAIN PLANT SPECIES IDENTIFICATION

MACHINE LEARNING PROJECT

PRESENTED BY: GROUP 12

Deron Foo Yijia

Esther Chai Hui Min

Jayne Wong Hieng Siew

Lai Jun Hong

William Wan Chin Lee

PRESENTATION OUTLINE

- Introduction
- Problem Statement
- Approach 1: Mix-Stream CNN
- Approach 2: DinoV2 as Feature Extractor
- Approach 3: Cut + DINOv2
- Approach 3: Hybrid Approach
- Evaluation and Comparison
- Best Model Selection
- Future Considerations

INTRODUCTION

- Deep learning has achieved strong performance in plant species identification.
- However, many plant species, especially tropical ones, lack sufficient field photos, making training difficult.
- Herbarium images (pressed, dried specimens) are far more available and well-curated.
- Recent research explores using herbarium images to help identify species in real-world field photos, forming a cross-domain identification task.

PROBLEM STATEMENT

- Field images and herbarium images come from different visual domains due to differences in lighting, background, structure, and preservation.
- Models trained on herbarium images alone often fail to generalize to field images.
- Data-deficient species worsen the challenge, as there are few or no field samples for certain classes.
- The task is to build a model that learns correspondence between both domains so it can correctly identify species in field images, even with limited field training data.

SETUP & DATA PREPROCESSING

- Import Libraries and Dependencies
- Data Loading
- Create global class mapping
- Data Split
- Data Augmentation
  - Normalization
  - Resize
  - Center Crop
  - Horizontal Flip
  - Rotation

APPROACH 1: MIX-STREAM CNN

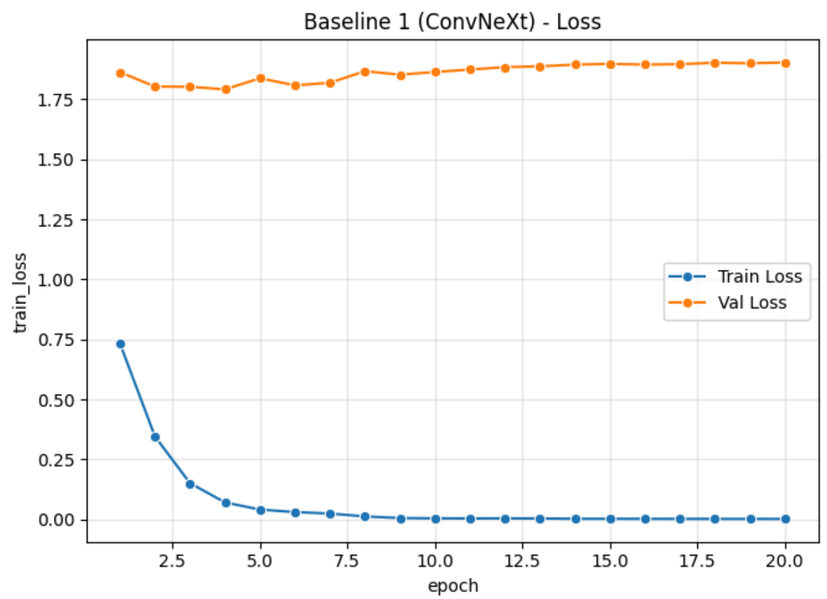
Model Architecture

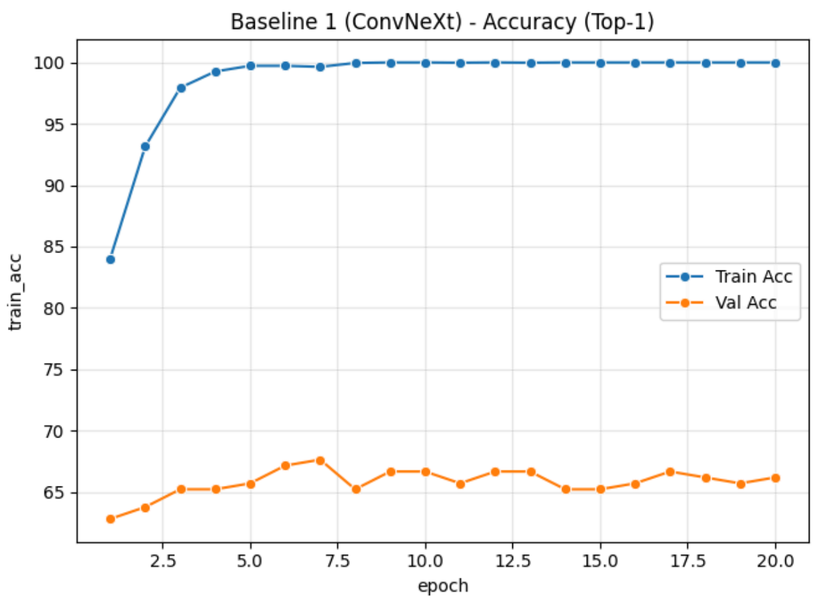
- **Model:** ConvNeXt-Base (Pre-trained on ImageNet)
- **Strategy:** Partial Freezing
  - **Frozen:** Early layers (Stages 0 & 1) to retain generic image features
  - **Fine-Tuned:** Later layers (Stages 2 & 3) + Classification Head

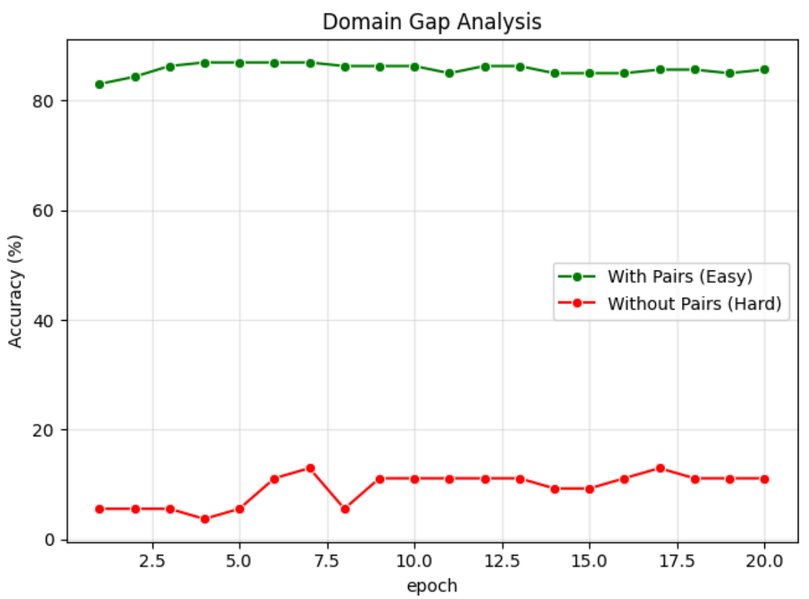
Configuration & Setting

- **Learning Rate:** 0.0001
- **Optimizer:** AdamW
- **Weight Decay:** 0.0001
- **Training Epochs:** 20

Training Results and Curves







Subset	Top-1 Acc	Top-5 Acc	Precision	Recall	F1-Score	Balanced Acc	Count
Overall	67.63	77.29	57.27	67.63	59.79	57.83	207
With-Pairs	86.93	96.08	88.54	86.93	85.76	86.39	153
Without-Pairs	12.96	24.07	12.04	12.96	12.35	15	54

APPROACH 2: DINOV2 AS FEATURE EXTRACTOR+ SVM

Field Image

Herbarium Image

dinov2_patch14_reg4_onlyclassifier_then_all

juliostat/dinov2_patch14_reg4_onlyclassifier_then_all

image-based plant species prediction model

Model Card | Code (1) | Discussion (0) | Competitions (0)

Model Details

Model Summary

The model is an image-based prediction models of plant species from the flora of southwestern Europe. The model is based on a ViT base patch 14 architecture pre-trained with the SSL (Self-Supervised Learning) Dinov2 method (https://arxiv.org/pdf/2309.16588.pdf), finetuned on the entire model, backbone and classification head. The model has been trained with the Exponential Moving Average (EMA) option for even better performances.

Downloads

720

572 in the last 30 days

Usability

3.33

APPROACH 2: DINOV2 AS FEATURE EXTRACTOR+ SVM

Model Architecture

Input Image

- Herbarium + field images
- Paried & unpaird datasets

DINOv2 Backbone (Frozen)

- ViT-B/14 Plant-pretrained model
- No gradient updates

Feature Embeddings (7680dim)

- High-level semantic features
- Standardized before classification

SVM Classifier

- RBF kernel
- Grid search (C, gamma)
- Lightweight fast training

Class Prediction

Top-1 : **72.95**
Top-5 : **81.64**

APPROACH 2: DINOV2 AS FEATURE EXTRACTOR+ SVM

Configuration & Training Setup

Phase 1 (Head only / Linear Probing)

Epochs: 3

LR: 1e-3

Optimizer: AdamW

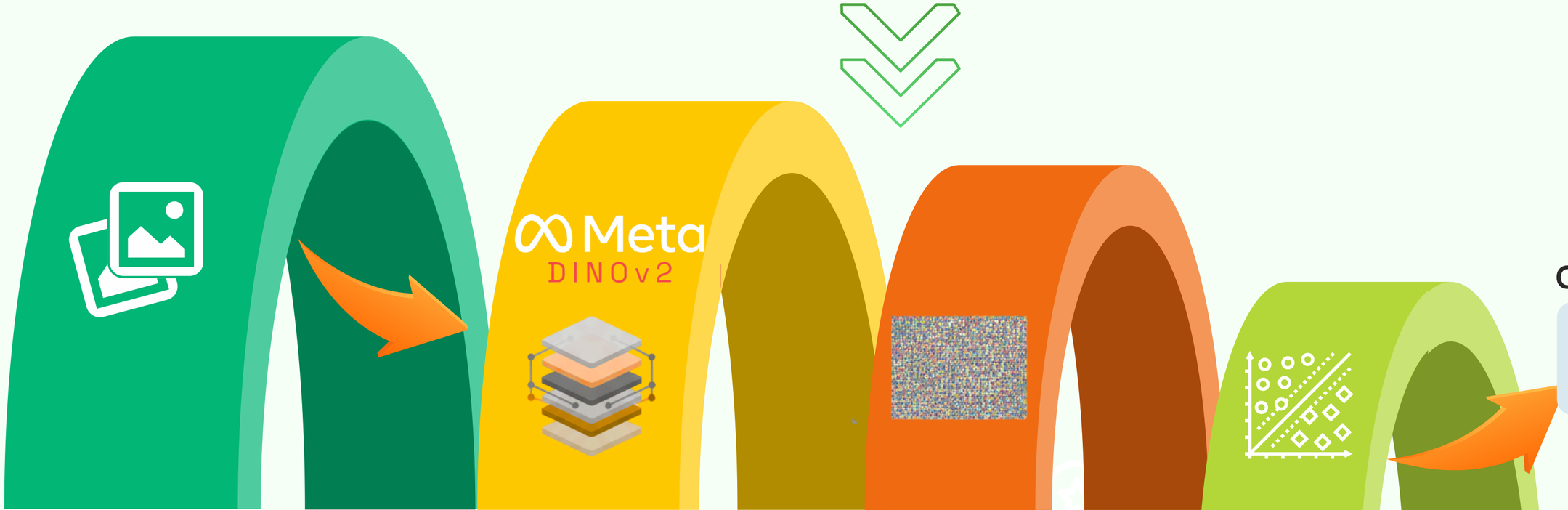
Freeze Backbone

Batch size: 32

Transforms:

Resize → Center

Crop → Normalize



Class Prediction

Top-1 : **72.95**

Top-5 : **81.64**

APPROACH 2: DINOV2 AS FEATURE EXTRACTOR+ SVM

Configuration & Training Setup

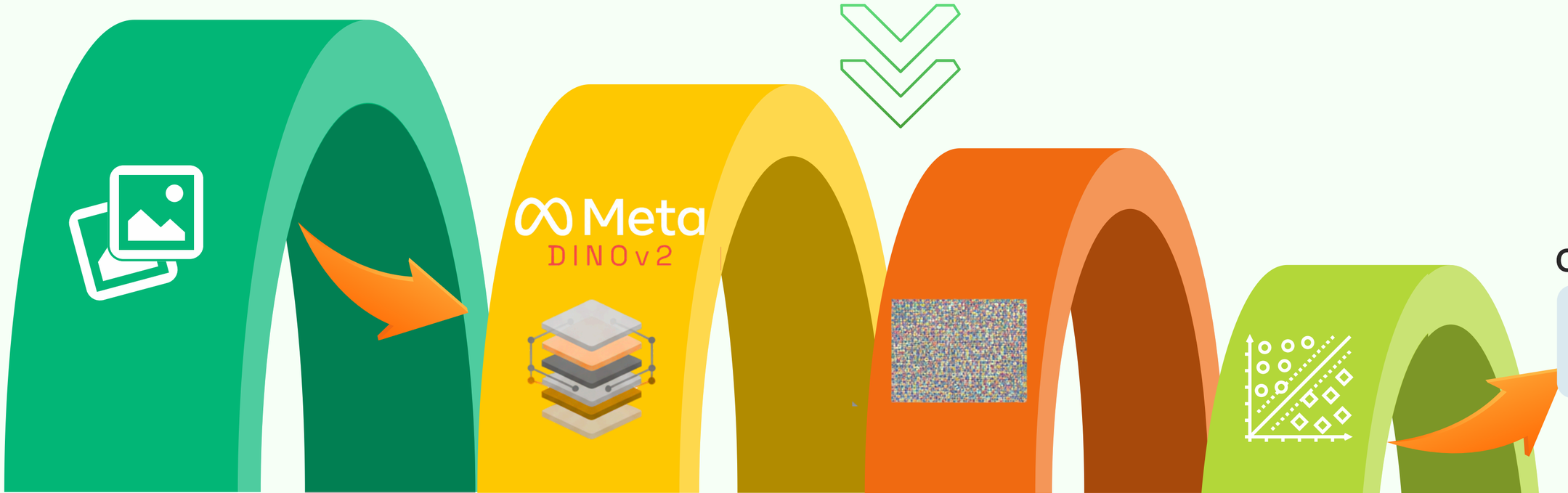
Phase 2 (Fine-tuning Backbone)

Epochs: 9
LR: 1e-4
Unfreeze last 2 Transformer blocks + Norm
Optimizer: AdamW
Scheduler: ReduceLROnPlateau

Batch size: 32

Transforms:

Resize → Center Crop → Normalize

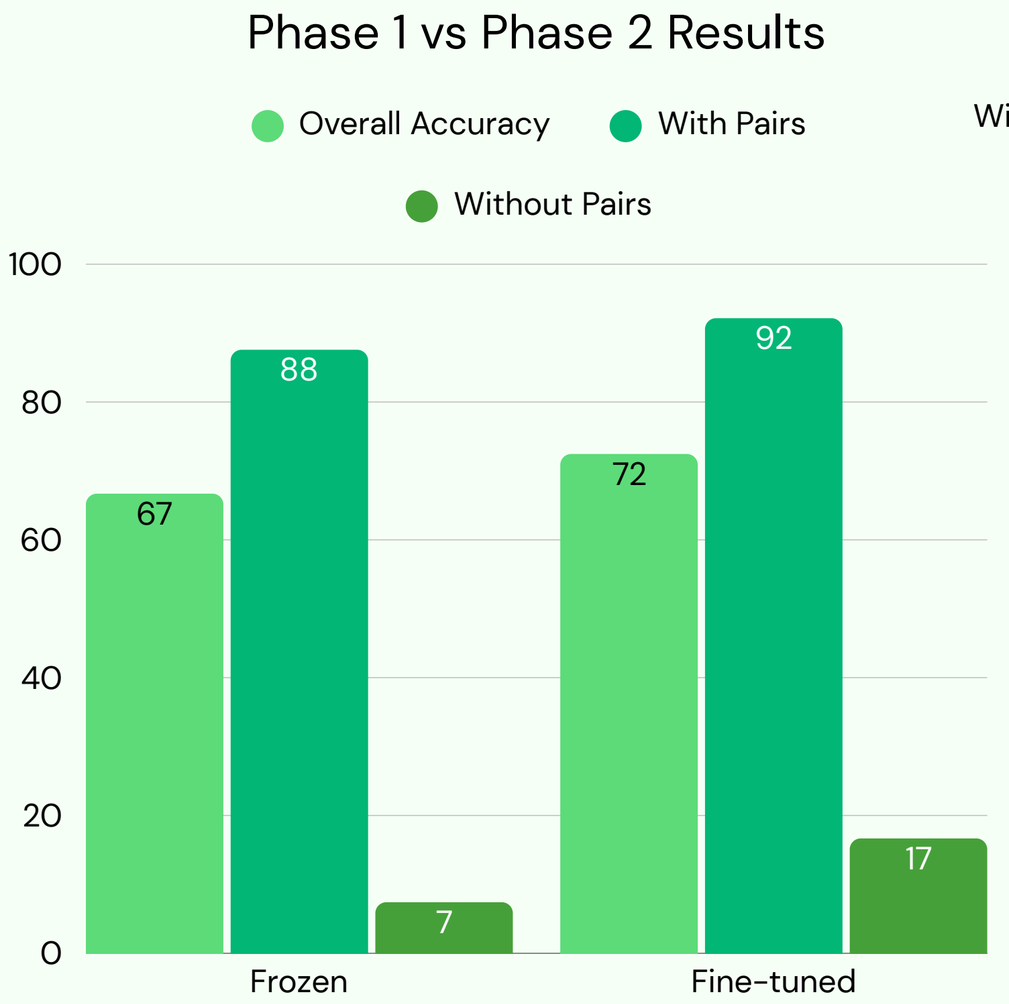


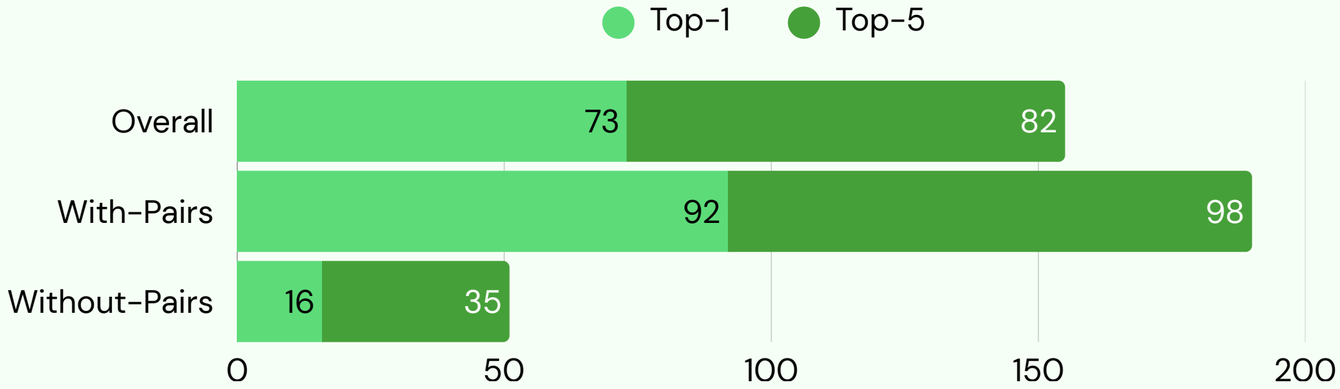
Class Prediction

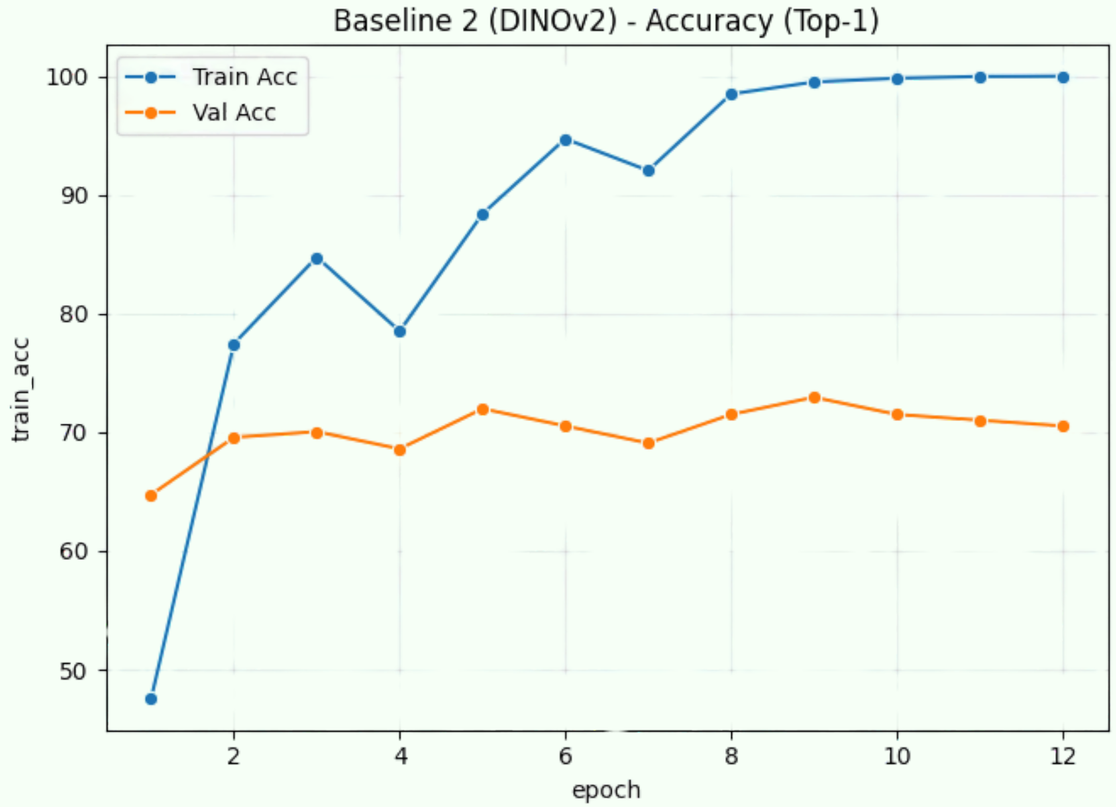
Top-1 : **72.95**
Top-5 : **81.64**

APPROACH 2: DINOV2 AS FEATURE EXTRACTOR+ SVM

Training Results and Curves



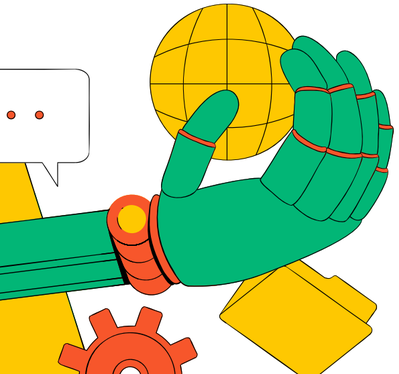




APPROACH 3: CUT + DINOV2

Model Architecture

1. Cross-Domain Image Translation
    a. Converts Herbarium → Field-like images to bridge domain gap
    b. Uses CUT (Contrastive Unpaired Translation) with:
        i. PatchNCE loss
        ii. Identity loss disabled
        iii. Contrastive layers: 4,8
        iv. Patch sampling: 128 patches
        v. Resize: 192 → crop 128



Step 1: Train the CUT Model with existing training sample

→

```
%cd contrastive-unpaired-translation

!python train.py --dataroot "{FASTCUT_ROOT}" \
  --name herb2field_cut_fast \
  --model cut \
  --no_dropout \
  --gpu_ids {GPU_ID} \
  --n_epochs 10 \
  --n_epochs_decay 10 \
  --batch_size 2 \
  --save_epoch_freq 10 \
  --print_freq 200 \
  --load_size 192 \
  --crop_size 128 \
  --nce_layers 4,8 \
  --num_patches 128 \
  --no_html \
  --display_id -1

%cd -
```

APPROACH 3: CUT + DINOV2

Model Architecture

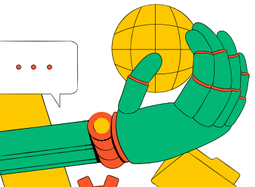
2. Synthetic Data Generation
   a. All herbarium images form unpaired classes are translated to fake field photos
   b. Integrated into training using ID-matched pairing
   c. Total of 1744 synthetic images and 6488 training images

3. Final Classifier: ViT-Base DINOv2
   a. Backbone: vit_base_patch14_reg4_dinov2.lvd142m
   b. Plant-pretrained model
   c. Head: Linear layer with 100 classes

```
%cd contrastive-unpaired-translation

!python test.py --dataroot "{FASTCUT_ROOT}" \
  --name herb2field_cut_fast \
  --model cut \
  --no_dropout \
  --phase test \
  --serial_batches \
  --results_dir "{DRIVE_ROOT}/models/fastcut_results" \
  --num_test 999999 \
  --gpu_ids {GPU_ID} \

%cd -
```



Step 2: Generation of Synthetic Data →

```
processing (0000)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/1000.jpg']
processing (0005)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/1011.jpg']
processing (0010)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/1130.jpg']
processing (0015)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/11765.jpg']
processing (0020)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/123297.jpg']
processing (0025)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/135654.jpg']
processing (0030)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/1396.jpg']
processing (0035)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/145749.jpg']
processing (0040)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/154130.jpg']
processing (0045)-th image... ['/content/drive/MyDrive/COS30082_Cross_Domain/models/fastcut_work/fastcut_data/testA/1566.jpg']
```

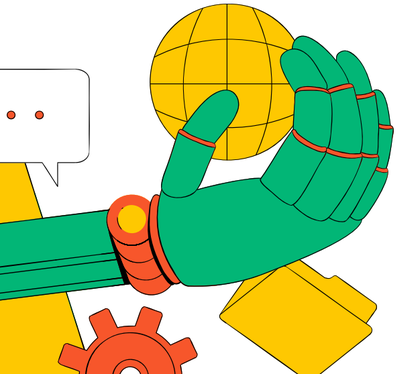
APPROACH 3: CUT + DINOV2

Data Transformation

Transform:

- Training
  - Resize 518 x 518
  - Random Horizontal Flip
  - Normalize
- Validation
  - Resize 518 x 518
  - Normalize

```
train_tf = transforms.Compose([
    transforms.Resize((518,518)),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225])
])
val_tf = transforms.Compose([
    transforms.Resize((518,518)),
    transforms.ToTensor(),
    transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225])
])
```



APPROACH 3: CUT + DINOV2

Configuration & Setting

DINOv2 Wrapper

- Backbone outputs feature vector (feat_dim = 768)
- Classification Head： Linear (feat_dim → 100)

Model Training

Phase 1: Linear-Probe (Head-Only Training)

- Freeze DINOv2 Backbone
- Train only classifier head
- Epochs: 10
- LR: 1e-4
- Optimizer: AdamW
- Weight decay: 1e-4

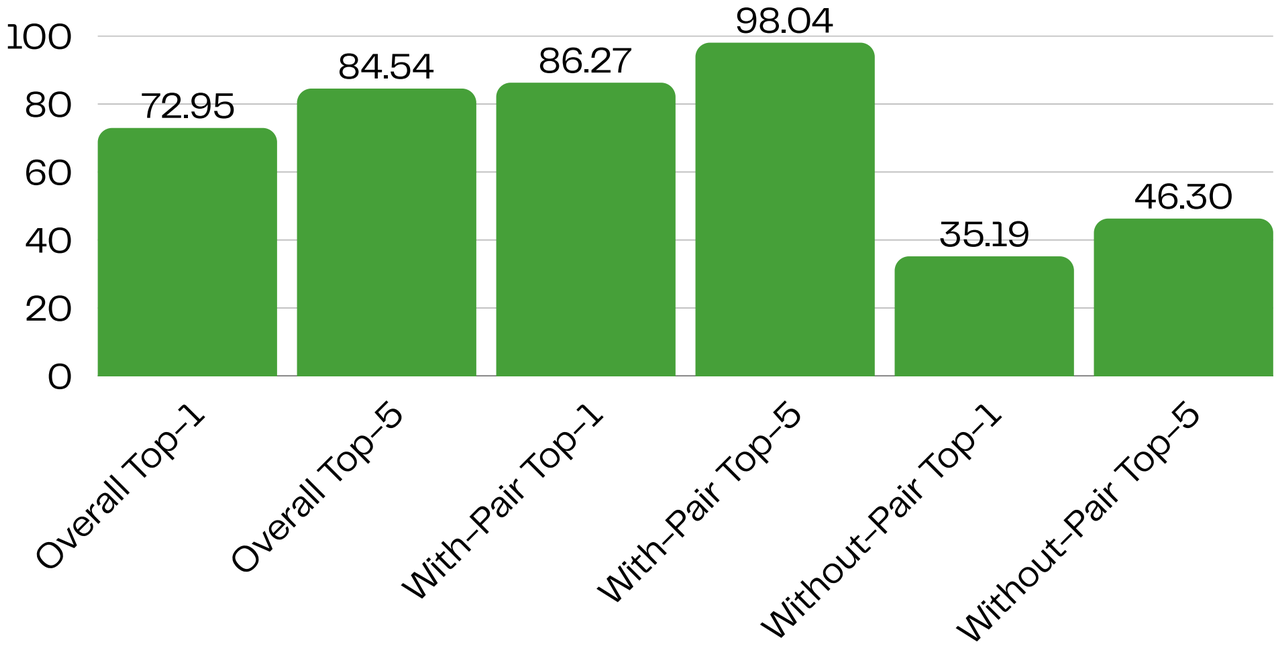
Phase 2: Full Fine-Tune

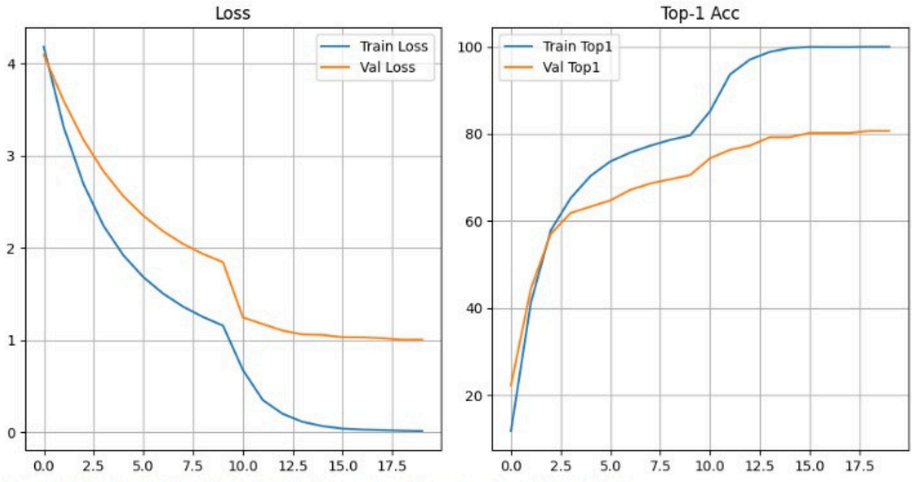
- Unfreeze all backbone weights
- Epochs； 10
- LR: 1e-5
- Optimizer: AdamW

Loss Function: Cross-Entropy

APPROACH 3: CUT + DINOV2

Training Results and Curves





Trends after unfreezing backbone:

- Loss decreases
- Accuracy raises

Possible Improvements

- Higher quality synthetic images
- Prototype-based classification
- Domain adaptation techniques

APPROACH 3B: DUAL-STREAM ENSEMBLE

Model Architecture

Hybrid Model (Classification)

Backbone (DinoV2)

Linear
Classifier Head

Metric Model (Embedding)

Input

Backbone (DinoV2)

Projection Head
(Linear, BatchNorm, ReLU)

Decision Gate

Output

APPROACH 3B: DUAL-STREAM ENSEMBLE

Configuration & Setting

Common configuration (shared by both models)

- **Backbone:** DINOv2 (ViT-Base) Pre-trained
- **Fine-Tuning Strategy:** Initial layers frozen; only the Last 5 Blocks are trainable
- **Optimizer:** AdamW with Cosine Annealing Scheduler
- **Decision Gate Threshold:** 0.93 (Selected via Grid Search)

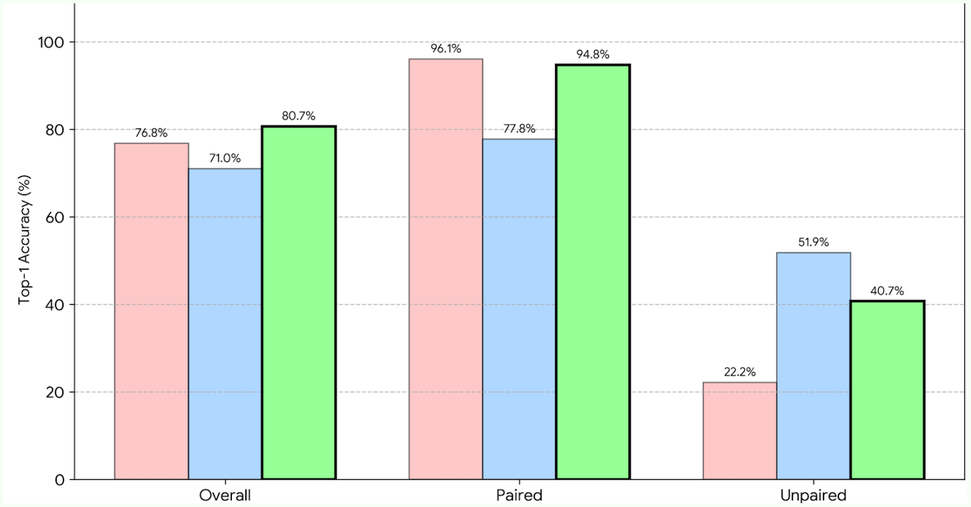
Separated configuration

Difference	Hybrid (Specialist)	Metric (Generalist)
Task-Specific Head	Linear Classifier (100 classes)	Projection Head (Linear -> BN -> ReLU)
Loss Function	Cross-Entropy + Triplet Loss	Triplet Loss
Triplet Margin	0.3	0.2

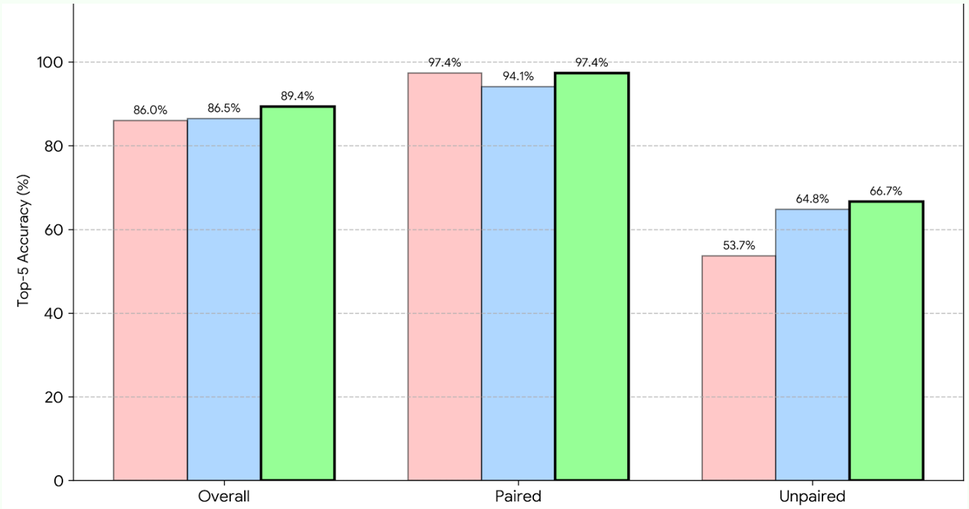
APPROACH 3B: DUAL-STREAM ENSEMBLE

Training Result

Top 1 Accuracy



Top 5 Accuracy

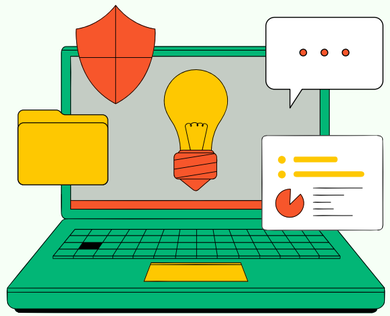


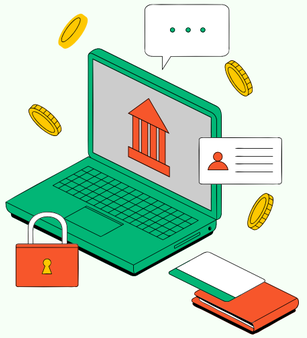
Hybrid Model (Specialist) | Metric Model (Generalist) | Ensemble (Proposed)

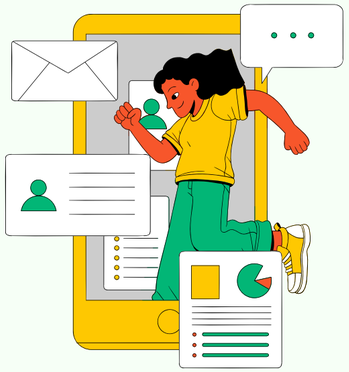
OVERALL EVALUATION RESULTS

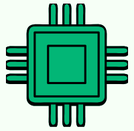
Approach / Accuracy	1: CovNeXt	2: DINOv2 + SVM	3: Cut + DINOv2	4. Dual-Stream Ensemble	5. DANN + Taxonomy
Overall Top-1	67.63%	**72.95%**	**72.95%**	**80.68%**	74.88%
Overall Top-5	77.29%	81.64%	84.54%	**89.37%**	85.99%
With-Pair Top-1	86.93%	92.81%	86.27%	**94.77%**	90.20%
With-Pair-Top-5	96.08%	98.04%	98.04%	**97.39%**	95.42%
Without-Pair Top-1	**12.96%**	**16.67%**	**35.19%**	**40.74%**	**31.48%**
Without-Pair Top-5	24.07%	35.19%	46.30%	**66.67%**	59.26%

DEMO









COS30082 APPLIED MACHINE LEARNING

WE LEARN FOR THE FUTURE

THANK YOU!

PRESENTED BY: GROUP 12