

Set2 DL Answers:

Scenario 1: Pretrained Model Predicts Only One Class in Object Detection

Context: You fine-tuned a pretrained YOLOv5 model on a custom dataset (detecting helmets). After training, it always predicts only one class.

Problem: Despite having multiple classes in the dataset, the model outputs only “helmet” for every object

Logical Solution:

Check the number of classes

Verify label files

Confirm Model architecture

Training hyperparameters

Loss analysis

Scenario 2: Transfer Learning Model Trains Too Slowly on Small Dataset

Context: You are using a pretrained [ResNet50](#) model for face mask classification (2 classes). You fine-tune the entire model on a small dataset (500 images).

Problem: Training takes too long and doesn't converge well.

Freeze most layers-earlier layers

Replace the original fully connected layer with a simple head

Use a higher learning rate

Reduce batch size

Scenario 3: Transfer Learning Works on Training Data but Fails on New Environment

Context: You used a pretrained YOLOv8 model trained on outdoor images to detect tigers. But when used in night-mode CCTV footage, detection fails.

Problem: Domain shift between training data (bright daylight) and real-world application (night).

Collect or synthesis night mode

- Augment the existing daylight dataset with night time transformations
- Fine tune pretrained YOLOv8 model
- Validate performance

Scenario 4: Object Detection Model Shows Bounding Boxes but Misses Object

Context: You trained a YOLOv5 model using transfer learning on a small vehicle dataset. Boxes are

Problem: Bounding box coordinates are incorrect due to **wrong label format or image resizing mismatch.**

- Verify Label format

- Check Data Preprocessing

- Inspect Coordinate Conversion

- Debug Annotation Files

- Retrain Model

Scenario 5: Model Gives High mAP on Training But Low Real-Time Accuracy

Context: You used pretrained **YOLOv8** for multi-class object detection (helmet, vest, mask). mAP on test set is 90%, but real-time inference on webcam is inaccurate.

Problem: Difference in resolution, camera angle, or image quality between webcam and training data.

Poor generalization due to **dataset bias.**

- Fine tune the model with a dataset that matches the webcam's resolution

- Augment the training data

- Preprocessing webcam frame

- Evaluate the model

Scenario 6: VGG16 Model Very Slow During Training

Project: Fashion product classification using **VGG16**

Problem: Model takes too long to train even with a small dataset and batch size.

- Use GPU instead of a CPU

- Reduce Model Complexity

Optimize Data Pipeline

Mixed Precision Training

Simplify Dataset

Scenario 7: InceptionV3 Fails to Train Due to Input Size Mismatch

Project: Dog breed classification using InceptionV3

Problem: Runtime error due to input size mismatch.

Verify the expected input shape

Check the actual input size

Resize or preprocess

Cofirm the data pipeline