3. Object Detection using ANN

This report details the implementation and results of an object detection task using the YOLOv8n model. The objective was to detect three classes: bottles, cups, and plates.

A key component of this project was an experiment to compare two training methodologies:

1. **Transfer Learning:** Fine-tuning a YOLOv8n model (yolov8n.pt) that was pre-trained on the COCO dataset.

2. **Training from Scratch:** Training the YOLOv8n architecture (yolov8n.yaml) with randomly initialized weights.

The models were trained, validated, and finally evaluated on a dedicated test set to determine the most effective approach.

3.1. Methadology

3.1.1. Environment and Dataset

- **Environment:** The experiment was conducted in a Google Colab environment, utilizing a Tesla T4 GPU.

- **Library:** The ultralytics library (version 8.3.217) was used for all training and inference tasks.

- **Dataset:** The Bottles-Cups-Plates-Combined.v1 dataset was used. The notebook's output logs show the dataset was split as follows:

  - Training: 105 images
  - Validation: 30 images
  - Test: 15 images

3.1.2. Data Preprocessing and Augmentation

The ultralytics YOLOv8 framework automatically handles all necessary preprocessing and augmentation during the model.train() call. This ensures the model learns from a robust and varied set of images, which is critical given the small size of the training set (105 images).

The following preprocessing and augmentation steps were applied to the training data for both experiments:

1. **Dataset Preparation :** The dataset was provided in the standard YOLO format, pre-split into train, valid, and test directories. Each image had a corresponding .txt label file specifying the class and bounding box coordinates.

2. **Automatic Preprocessing:** All images were resized to the specified input size of imgsz=640 (640x640 pixels). Pixel values (originally 0-255) were automatically scaled to a 0.0-1.0 range, which is the format the neural network expects.

3. **Data Augmentation (From Training Log)**: The framework applied a variety of "on-the-fly" augmentations during training to create new image variations and prevent overfitting. The logs confirm:

   – **Geometric Augmentations**:
      - **Mosaic**: mosaic=1.0 (Enabled for the first 40 epochs). This technique combines four different training images into one, forcing the model to learn objects in different locations and partial states.
      - **Horizontal Flip**: fliplr=0.5 (50% chance of flipping the image horizontally).
      - **Scaling & Translation**: scale=0.5 and translate=0.1.

   – **Color Space Augumentations**: The model applied random adjustments to Hue, Saturation, and Value (HSV) to make it robust to different lighting conditions (hsv_h=0.015, hsv_s=0.7, hsv_v=0.4).

   – **Albumentations Pipeline**: The log also shows an active albumentations pipeline, which included Blur(p=0.01), MedianBlur(p=0.01), ToGray(p=0.01), and CLAHE(p=0.01).

These built-in preprocessing and augmentation steps were a key factor in the high performance of the transfer learning model, as they artificially expanded the small training dataset and helped the model generalize.

3.1.3. Justification for Model Selection

The YOLOv8n (nano) model was selected for this task. This decision was based on its optimal balance of performance and efficiency, which is particularly suitable for this project's constraints:

– **High Speed and Small Size:** YOLOv8n is the most lightweight and fastest model in the YOLOv8 series. Its small file size (approx. 3.8-6 MB) allows for rapid downloading, fast training, and quick inference.

– **Resource Efficiency:** The model is "designed for environments with limited computational resources". This makes it an ideal choice for the free Google Colab environment, which provides a Tesla T4 GPU. Its low memory (RAM and VRAM) requirements ensure that training can be completed without hardware limitations.

– **Ideal for Transfer Learning:** The experiment's core comparison relied on transfer learning. YOLOv8n comes with pre-trained weights from the COCO dataset, providing a powerful feature-extraction baseline. This is crucial for achieving high accuracy when training on a small custom dataset (like the 105 images in this project).

– **Strong Performance Baseline:** Despite being the smallest variant, YOLOv8n offers a state-of-the-art "balance between speed and accuracy". This makes it an excellent benchmark for a proof-of-concept or experimental project.

3.1.4. Training Process

Two separate training experiments were conducted, both using the same dataset and core parameters:

– **Epochs:** 50
– **Image Size:** 640x640
– **Optimizer:** AdamW (automatically selected)

Experiment 1: With Transfer Learning A YOLO('yolov8n.pt') model was loaded, which comes with weights pre-trained on the COCO dataset. The model was then fine-tuned on the custom dataset for 50 epochs.

Experiment 2: Without Transfer Learning A YOLO('yolov8n.yaml') model was initialized, which loads only the model's architecture. The model was then trained from scratch on the custom dataset for 50 epochs.

3.1.5. Evaluation and Results

After training, both models were evaluated on the 15 images in the unseen test split. The performance metrics clearly demonstrate the difference between the two approaches.

The "best" weights (based on validation performance) from each training run were used for this final evaluation.

Performance on Test Set (15 images)

Table 4. Performace metrics on the test set

Metric	Model 1: Transfer Learning	Model 2: From Scratch
mAP50-95 (Primary Metric)	0.926	0.389
mAP50 (IoU > 0.5)	0.995	0.724
Precision	0.994	0.654
Recall	1.000	0.688

Analysis

The results are conclusive: The transfer learning model performed exceptionally well, while the model trained from scratch performed poorly.

– The transfer learning model achieved a mAP50-95 of 0.926 (92.6%), indicating high accuracy and correctly placed bounding boxes. Its precision (0.994) and perfect recall (1.000) on the test set show it correctly identified all 24 object instances without making false-positive predictions.

– The "from scratch" model only achieved a mAP50-95 of 0.389 (38.9%). While its mAP50 was higher (0.724), this suggests it could find objects but struggled to place the bounding boxes with high precision.

This vast performance gap is expected. The dataset (105 training images) is too small for a model to learn robust object features from scratch. The transfer learning model leverages the powerful, general features (edges, textures, shapes) learned from the massive COCO dataset, allowing it to easily adapt to the new, specific classes.

3.2. Conclusion & Final Predictions

The experiment confirms that transfer learning is the correct and most effective method for this task. Based on these results, the transfer learning model selected as the final, best-performing model.

This model was then used to run predictions on all 15 images in the test set.

```
47ed77a.jpg: 640x640 2 bottles - v1 2024-10-30 2-22pms, 8.4ms
50eb7c0.jpg: 640x640 2 bottles - v1 2024-10-30 2-22pms, 7.3ms
09f2eb3.jpg: 640x640 1 bottles - v1 2024-10-30 2-22pm, 7.2ms
208131c.jpg: 640x640 1 bottles - v1 2024-10-30 2-22pm, 7.3ms
f5c972f.jpg: 640x640 1 ------------------------------, 2 Cups and Plates - v2 cupsandchairs9-15pms, 7.3ms
d36d99a.jpg: 640x640 1 ------------------------------, 1 Cups and Plates - v2 cupsandchairs9-15pm, 7.3ms
8e8b369.jpg: 640x640 1 bottles - v1 2024-10-30 2-22pm, 7.3ms
f0adceb.jpg: 640x640 1 ------------------------------, 1 Cups and Plates - v2 cupsandchairs9-15pm, 7.1ms
ed604d4.jpg: 640x640 1 ------------------------------, 1 Cups and Plates - v2 cupsandchairs9-15pm, 11.6ms
fd52e7d0.jpg: 640x640 1 bottles - v1 2024-10-30 2-22pm, 11.5ms
8aaae976.jpg: 640x640 1 ------------------------------, 1 Cups and Plates - v2 cupsandchairs9-15pm, 10.6ms
36d4a810.jpg: 640x640 2 bottles - v1 2024-10-30 2-22pms, 11.8ms
24a9609c.jpg: 640x640 1 ------------------------------, 1 Cups and Plates - v2 cupsandchairs9-15pm, 12.5ms
90e145ea.jpg: 640x640 1 ------------------------------, 1 Cups and Plates - v2 cupsandchairs9-15pm, 9.8ms
4334ee2f.jpg: 640x640 1 ------------------------------, 1 Cups and Plates - v2 cupsandchairs9-15pm, 8.6ms
 at shape (1, 3, 640, 640)
```

Fig. 10 Object detection results