Project report

Hand Digit Recognition Project

Introduction

This project focuses on creating a custom dataset and training a model to recognize hand digits from images. I took a structured approach to dataset creation, preprocessing, augmentation, and model selection to achieve reliable performance.

Dataset Creation

To create the dataset, I used **Roboflow** for its robust tools for annotation and augmentation. I manually labeled hands and fingers in each image, ensuring precise annotations to allow the model to learn effectively. The dataset comprises **117 images**, including both original images and augmented versions.

Problems I faced about annotating (labeling the dataset):

I tried 3 approaches (at the end selected 3rd approach):

Annotating Only Fingers:

- Likely resulted in poor performance because context (like hand placement) was missing.

Annotating Only Hands:

- Improved context but lost accuracy of finger information, which may explain the slightly better but still poor outcome.

Annotating Hands and Fingers Separately:

- Combines the advantages of context (hands) and tiny detail (fingers), enabling the model to recognize both detailed properties.

Preprocessing

Preprocessing steps were crucial to standardize the dataset and downsize images for smaller file sizes and faster training.:

- **Auto-Orient**: Ensured consistent image orientation.

- **Static Crop**: Focused on regions of interest by cropping 2-99% horizontally and 3-100% vertically.
- **Resize**: Adjusted all images to fit within a 512x512 resolution to balance detail retention and computational efficiency.

These steps ensured uniformity and compatibility with the model input requirements.

Data Augmentation

To enhance dataset variability, I implemented augmentations that simulated real-world conditions. Each original image was augmented with:

- **Horizontal Flip**: To account for mirroring (e.g., left vs. right hands).
- **Rotation (-15° to +15°)**: Introduced random tilts for varying orientations.
- **Crop (0-5% Zoom)**: Simulated different distances by zooming in and out.
- **90° Clockwise Rotation**: Further diversified hand orientations.
- **Saturation and Brightness Adjustments**: Ranged from -50% to +50% saturation and -25% to +25% brightness to handle lighting variations.
- **Blur (up to 1.8px)**: Simulated real-world imperfections. Add random Gaussian blur to help the model be more resilient to camera focus.
- **Noise (up to 1.45%):** Add noise to help the model be more resilient to camera artifacts.

These augmentations expanded the dataset's diversity, ensuring the model could generalize well to unseen data.

Model Selection

I chose **YOLOv8**, a leading object detection model, for its ability to detect multiple objects in image. YOLOv8's efficiency and accuracy made it ideal for identifying hands and fingers, which require precise bounding box predictions.

Model's performance based on Training

```
Model summary (fused): 168 layers, 3,006,038 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all          9         39      0.927       0.94      0.961      0.707
                finger          9         27      0.922      0.881      0.928      0.583
                  hand          9         12      0.932          1      0.995      0.831
Speed: 0.1ms preprocess, 1.5ms inference, 0.0ms loss, 1.6ms postprocess per image
Results saved to runs/detect/train
```

Summary of Metrics

- **Overall Model Performance (Class: all):**

  - **Precision (P):** 0.927
  - **Recall (R):** 0.94
  - **mAP@50:** 0.961
  - **mAP@50-95:** 0.707

- **Performance by Class:**
  - **Finger:**

    - Precision: 0.922
    - Recall: 0.881
    - mAP@50: 0.928
    - mAP@50-95: 0.583

  - **Hand:**

    - Precision: 0.932
    - Recall: 1.0
    - mAP@50: 0.995
    - mAP@50-95: 0.831

Analysis

1. **Overall Accuracy:**

   - The **mAP@50** score of 0.961 indicates that the model performs very well in terms of locating objects correctly within acceptable IoU thresholds.
   - The **mAP@50-95** score (0.707) suggests a decrease in performance as IoU thresholds become stricter. This is common especially for small objects or complex shapes like "fingers."

2. **Class-Specific Performance:**

   - The **"hand"** class performs better with near-perfect scores (mAP@50 = 0.995, Recall = 1.0). This suggests the model has learned to identify and localize "hands" accurately.
   - The **"finger"** class has slightly lower metrics:

     - Lower Recall (0.881) means the model sometimes misses detecting "finger" instances.

- The **mAP@50-95** of 0.583 indicates challenges with precise bounding boxes, especially for stricter IoU thresholds.

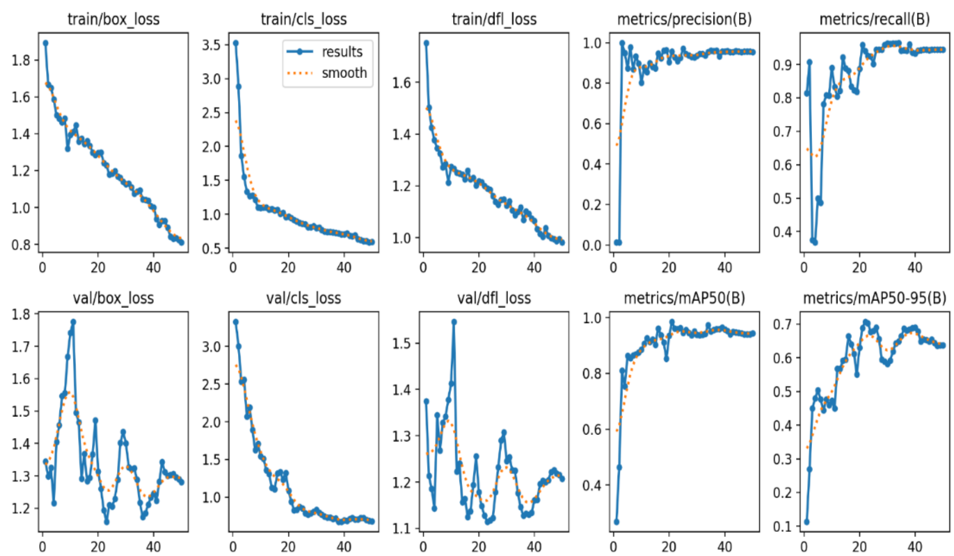
- **Precision vs. Recall:**

  - Both **Precision** and **Recall** are high, indicating that the model minimizes false positives and false negatives effectively. However, slight dips in Recall for "finger" suggest some missed detections.

- **Speed:**

  - The inference speed of 1.5 ms per image is efficient and suitable for real-time applications.

Curves:



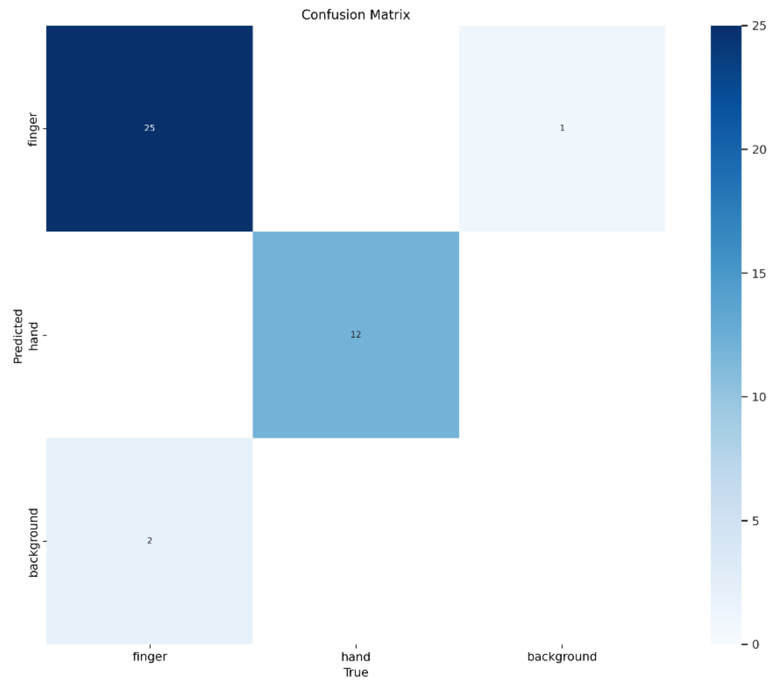
Issues:

The fluctuations in validation losses may indicate slight overfitting, particularly for certain classes (likely "finger"). This might also be caused by an imbalanced dataset or high variability in the validation set.

My takeaways to improve the model:

- Dataset Balance - reducing the imbalance of dataset
- Use regularization methods like Dropout
- Hyperparameter Tuning
- Analysis of Validation Dataset

Confusion matrix:



Matrix Observations

1. **True Positives:**

   - **Finger:** 25 instances correctly classified as "finger."
   - **Hand:** 12 instances correctly classified as "hand."

2. **Misclassifications:**

   - **Background classified as Finger:** 1 instance of "background" was misclassified as "finger." (False Positive)
   - **Finger classified as Background:** 2 instances of "finger" were misclassified as "background." (False Negative)

Try on Realtime in your browser:

https://demo.roboflow.com/hand-digit-identifying/5?publishable_key=rf_UBEq7wtaiFUzDFgSdBHCGBSd7xE3

Try on mobile