Implementing Object Detection on a Dataset

Mid-term Project

11:11PM

Tip: Use links to go to a different page inside your presentation.

How: Highlight text, click on the link symbol on the toolbar, and select the page in your presentation you want to connect.

Agenda

Topics Covered

Data Preparation

Model Implementation

Training

Evaluation

Challenges

Conclusion

Start

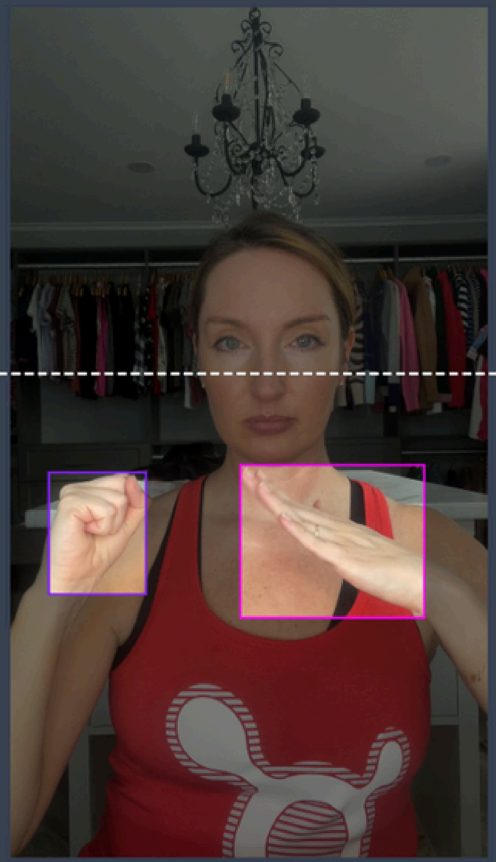
SELECTED DATASET & OBJECT DETECTION ALGORITHM

YOLO (You Only Look Once)

is a fast object detection model that detects multiple objects in a single pass by dividing images into grids and predicting bounding boxes and class probabilities simultaneously.

SSD (Single Shot Multibox Detector)

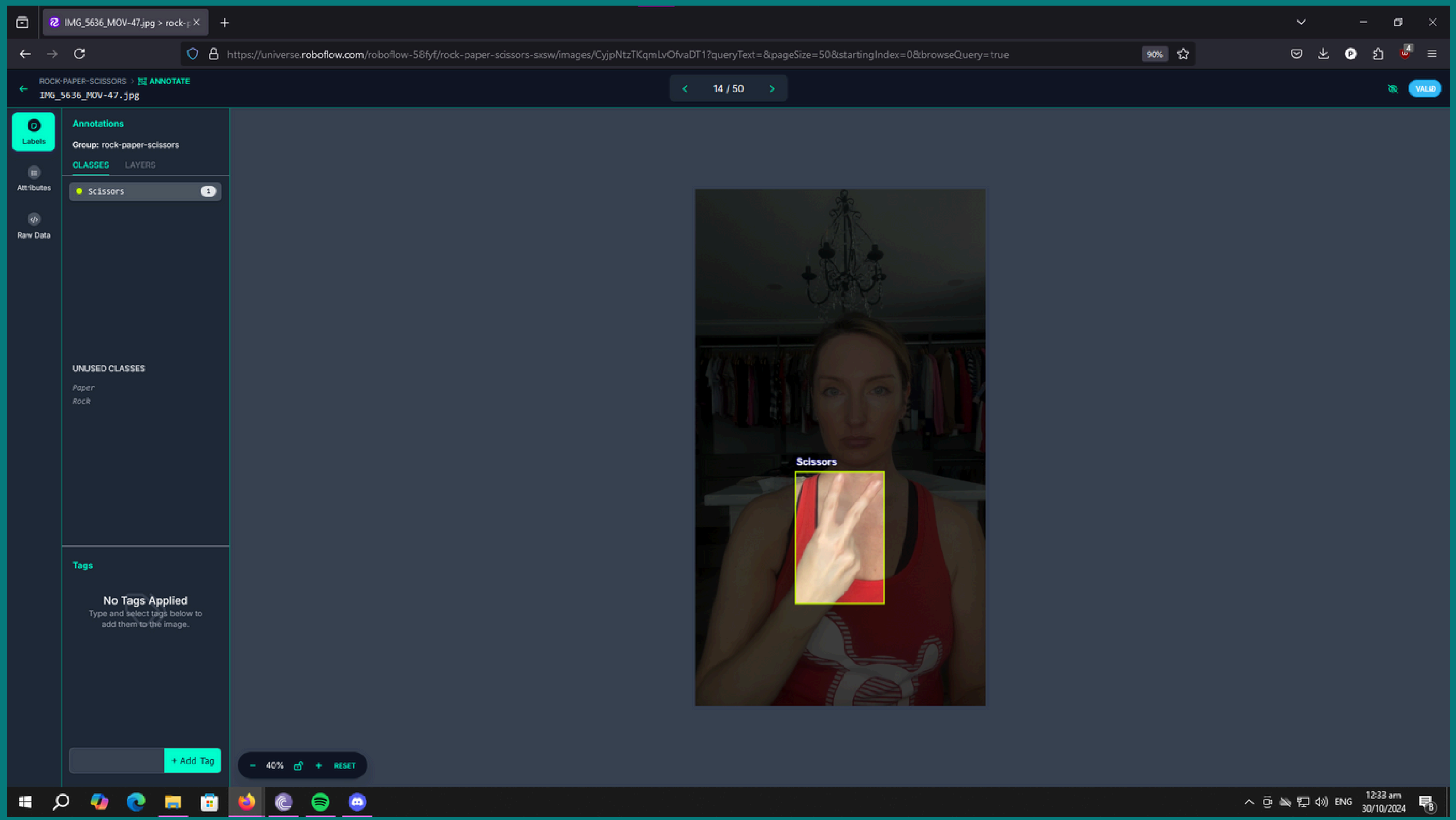
is an efficient object detection model that predicts bounding boxes and class scores in a single pass, balancing speed and accuracy for real-time applications.



Back to Agenda Page

DATA PREPARATION

Each image has a group of objects against a variety of backgrounds, like a hand gesturing a rock, paper, or scissors. To annotate the data set, Roboflow has been used to correctly label and locate each gesture thereby giving rise to an exhaustive training set for tasks including object detection and classification.



Back to Agenda Page

Model Implementation

WHY YOLO ?

- Object Detection in Real Time
- High Precision
- Single-Shot Detection
- Excellent Results on a Small Object
- Effective GPU Utilization
- Capacity to Manage Various Scales

WHY SSD ?

- Real-Time Performance
- Single-Stage Detection
- Multi-Scale Feature Maps
- High Accuracy at the Cost of Speed
- Adaptability and Expandability
- Effective for Mobile Use

Back to Agenda Page

Model Implementation

YOLO (You Only Look Once)

Load Model

```python
from ultralytics import YOLO

# Load a COCO-pretrained YOLOv8n model
model = YOLO("yolov8n.pt")

# Display model information (optional)
model.info()
```

```
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#ultralytics-settings.
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n.pt to 'yolov8n.pt'...
100%|██████████| 6.25M/6.25M [00:00<00:00, 336MB/s]
YOLOv8n summary: 225 layers, 3,157,200 parameters, 0 gradients, 8.9 GFLOPs
(225, 3157200, 0, 8.8575488)
```

Back to Agenda Page

Model Implementation

SSD (Single Shot Multibox Detector)

Initial Setup

```
!wget http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz
!tar -xvf ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz
```

```
--2024-10-22 06:44:09--  http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_320x320_coco17_tpu-8.tar.gz
```

```
ssd_mobilenet_v2_320x320_coco17_tpu-8/saved_model/variables/variables.data-00000-of-00001
ssd_mobilenet_v2_320x320_coco17_tpu-8/saved_model/variables/variables.index
```

```
# Clone the tensorflow models repository from GitHub
!pip uninstall Cython -y # Temporary fix for "No module named 'object_detection'" error
!git clone --depth 1 https://github.com/tensorflow/models
```

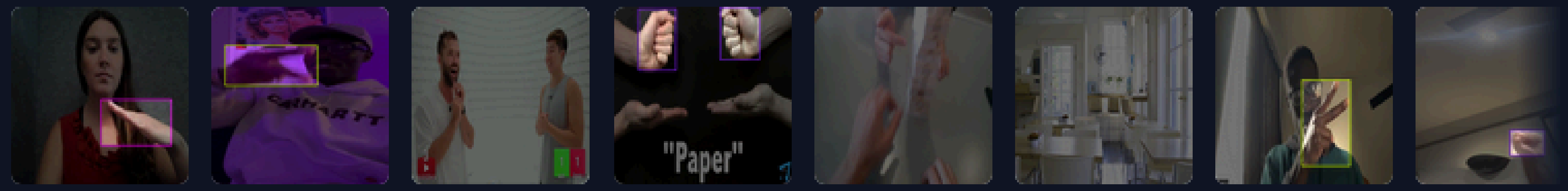
```
Found existing installation: Cython 3.0.11
Uninstalling Cython-3.0.11:
  Successfully uninstalled Cython-3.0.11
Cloning into 'models'...
```

Back to Agenda Page

Model Training

7521 Total Images

View All Images →



Dataset Split

TRAIN SET 88%	VALID SET 8%	TEST SET 4%
6588 Images	604 Images	329 Images

Preprocessing

Auto-Orient: Applied
Resize: Stretch to 640x640

Augmentations

Outputs per training example: 3
Flip: Horizontal
Crop: 0% Minimum Zoom, 30% Maximum Zoom
Grayscale: Apply to 10% of images
Hue: Between -10° and +10°
Saturation: Between -25% and +25%
Brightness: Between -25% and +25%
Exposure: Between -15% and +15%

Back to Agenda Page

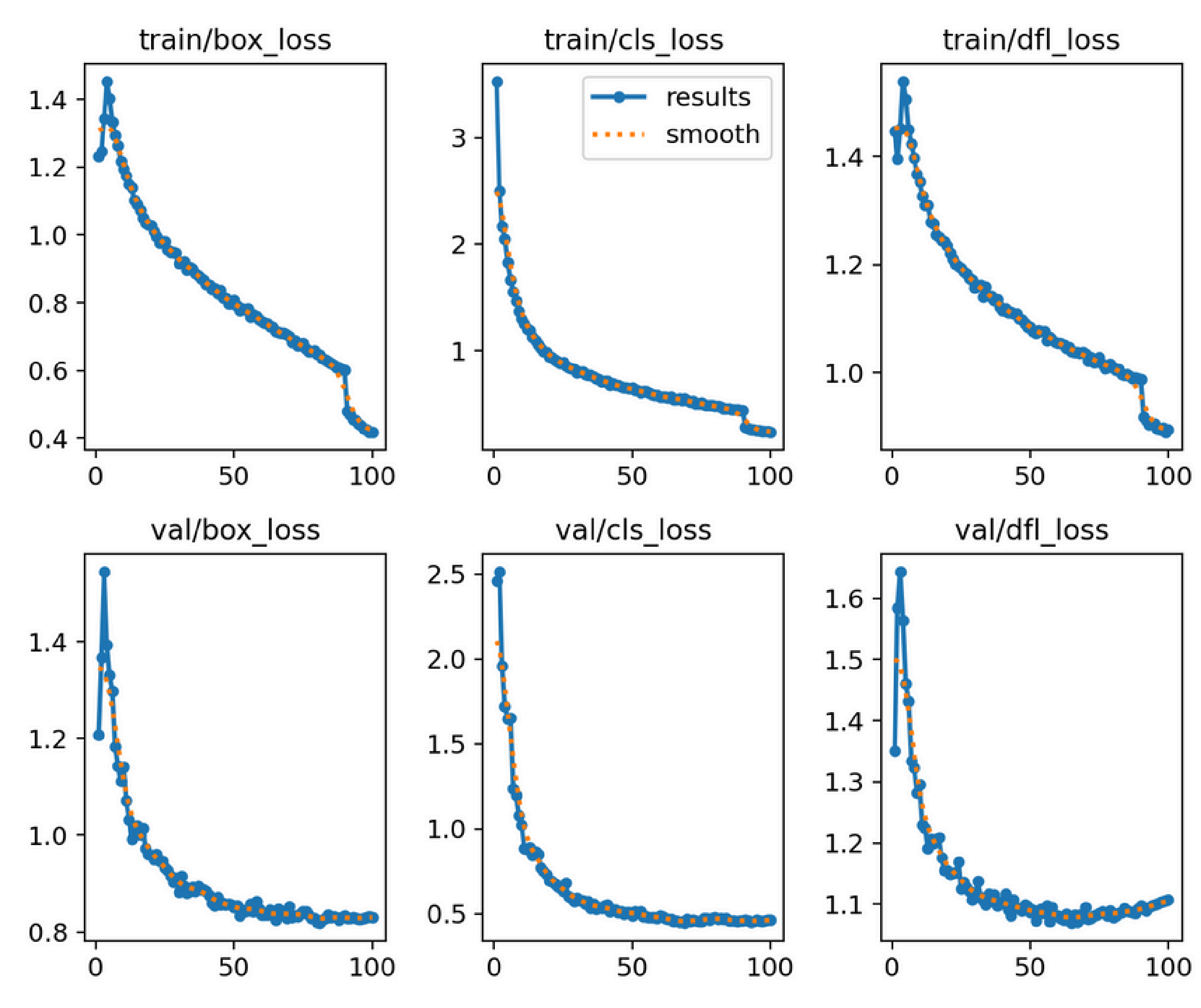
Model Training

YOLO (You Only Look Once)

Train Model

```python
[ ]  1 results = model.train(data="/content/data.yaml", epochs=100, imgsz=640)
     2
```

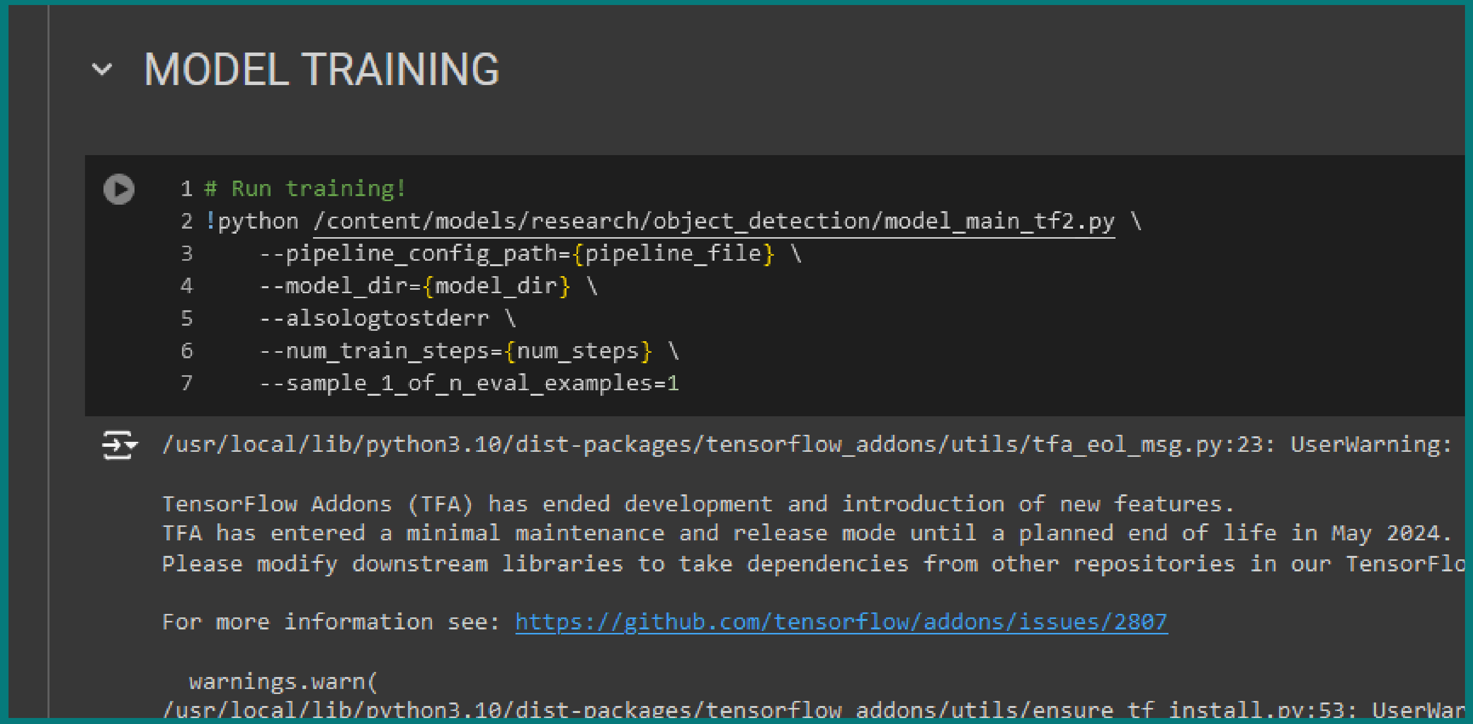
```
Ultralytics 8.3.20 🚀 Python-3.10.12 torch-2.4.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8n.pt, data=/content/data.yaml, epochs=100, time=None, patience=1
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 132MB/s]
Overriding model.yaml nc=80 with nc=3

                   from  n    params  module                                       arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
  2                  -1  1      7360  ultralytics.nn.modules.block.C2f             [32, 32, 1, True]
  3                  -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
  4                  -1  2     49664  ultralytics.nn.modules.block.C2f             [64, 64, 2, True]
  5                  -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
  6                  -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  8                  -1  1    460288  ultralytics.nn.modules.block.C2f             [256, 256, 1, True]
  9                  -1  1    164608  ultralytics.nn.modules.block.SPPF            [256, 256, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  1    148224  ultralytics.nn.modules.block.C2f             [384, 128, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  1     37248  ultralytics.nn.modules.block.C2f             [192, 64, 1]
 16                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  1    123648  ultralytics.nn.modules.block.C2f             [192, 128, 1]
```



Back to Agenda Page

Model Training

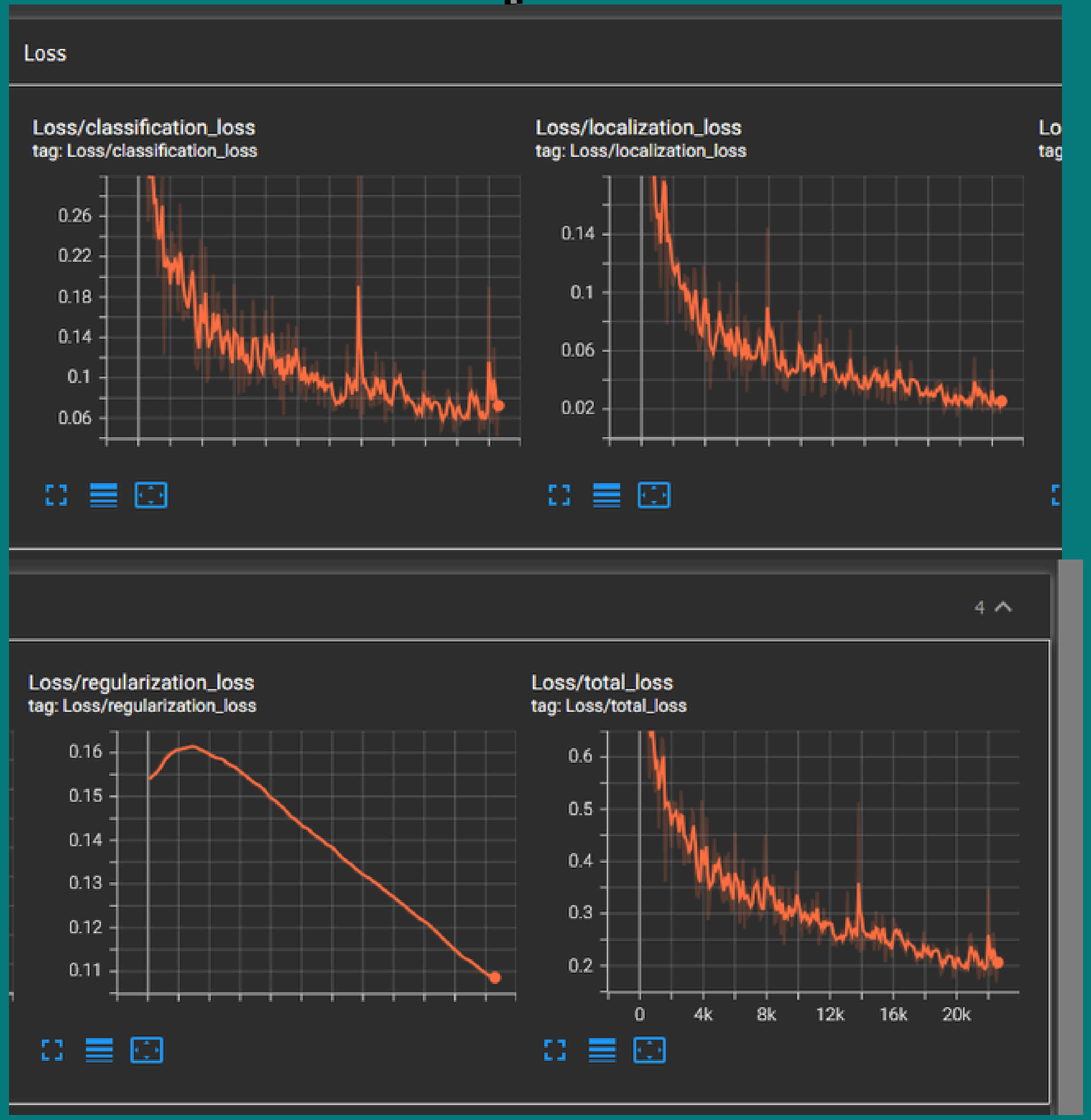
SSD (Single Shot Multibox Detector)



MODEL TRAINING

```
# Run training!
!python /content/models/research/object_detection/model_main_tf2.py \
    --pipeline_config_path={pipeline_file} \
    --model_dir={model_dir} \
    --alsologtostderr \
    --num_train_steps={num_steps} \
    --sample_1_of_n_eval_examples=1
```

```
/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/tfa_eol_msg.py:23: UserWarning:

TensorFlow Addons (TFA) has ended development and introduction of new features.
TFA has entered a minimal maintenance and release mode until a planned end of life in May 2024.
Please modify downstream libraries to take dependencies from other repositories in our TensorFlo

For more information see: https://github.com/tensorflow/addons/issues/2807

  warnings.warn(
/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/ensure_tf_install.py:53: UserWar
```



Back to Agenda Page

Testing and Evaluation

YOLO (You Only Look Once)

```
from ultralytics import YOLO

# Load a model
model = YOLO("/content/content_11/runs/detect/train/weights/best.pt")

# Customize validation settings
validation_results = model.val(data="/content/data.yaml", imgsz=640, batch=16, conf=0.25, iou=0.6, device="0"
```

```
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://
Ultralytics 8.3.23 🚀 Python-3.10.12 torch-2.5.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 22.3MB/s]
val: Scanning /content/valid/labels... 576 images, 238 backgrounds, 0 corrupt: 100%|██████████| 576/576 [00:00<0

                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 36/36
                   all        576        400      0.954      0.919      0.952      0.798
                 Paper        132        139      0.962      0.919      0.963      0.809
                  Rock        121        141      0.943      0.937      0.939      0.785
              Scissors        116        120      0.958        0.9      0.952      0.799
Speed: 0.7ms preprocess, 4.1ms inference, 0.0ms loss, 2.3ms postprocess per image
Results saved to runs/detect/val
```

Metric	RESULT
mAP50	95.2%
mAP50-95	96.3%
Precision	95.4%
Recall	91.9%
Inference Time	7.1ms

Back to Agenda Page

Testing and Evaluation

SSD (Single Shot Multibox Detector)

```
1 %cd /content/mAP
2 !python calculate_map_cartucho.py --labels=/content

/content/mAP
Calculating mAP at 0.50 IoU threshold...
84.98% = Paper AP
89.89% = Rock AP
92.08% = Scissors AP
mAP = 88.98%
Calculating mAP at 0.55 IoU threshold...
83.40% = Paper AP
89.89% = Rock AP
92.08% = Scissors AP
mAP = 88.46%
Calculating mAP at 0.60 IoU threshold...
80.16% = Paper AP
89.89% = Rock AP
90.44% = Scissors AP
mAP = 86.83%
Calculating mAP at 0.65 IoU threshold...
73.71% = Paper AP
85.74% = Rock AP
```

Metric	RESULT
mAP50	88.98%
mAP50-95	63.83%
Precision	73.20%
Recall	92.26%
Inference Time	70ms

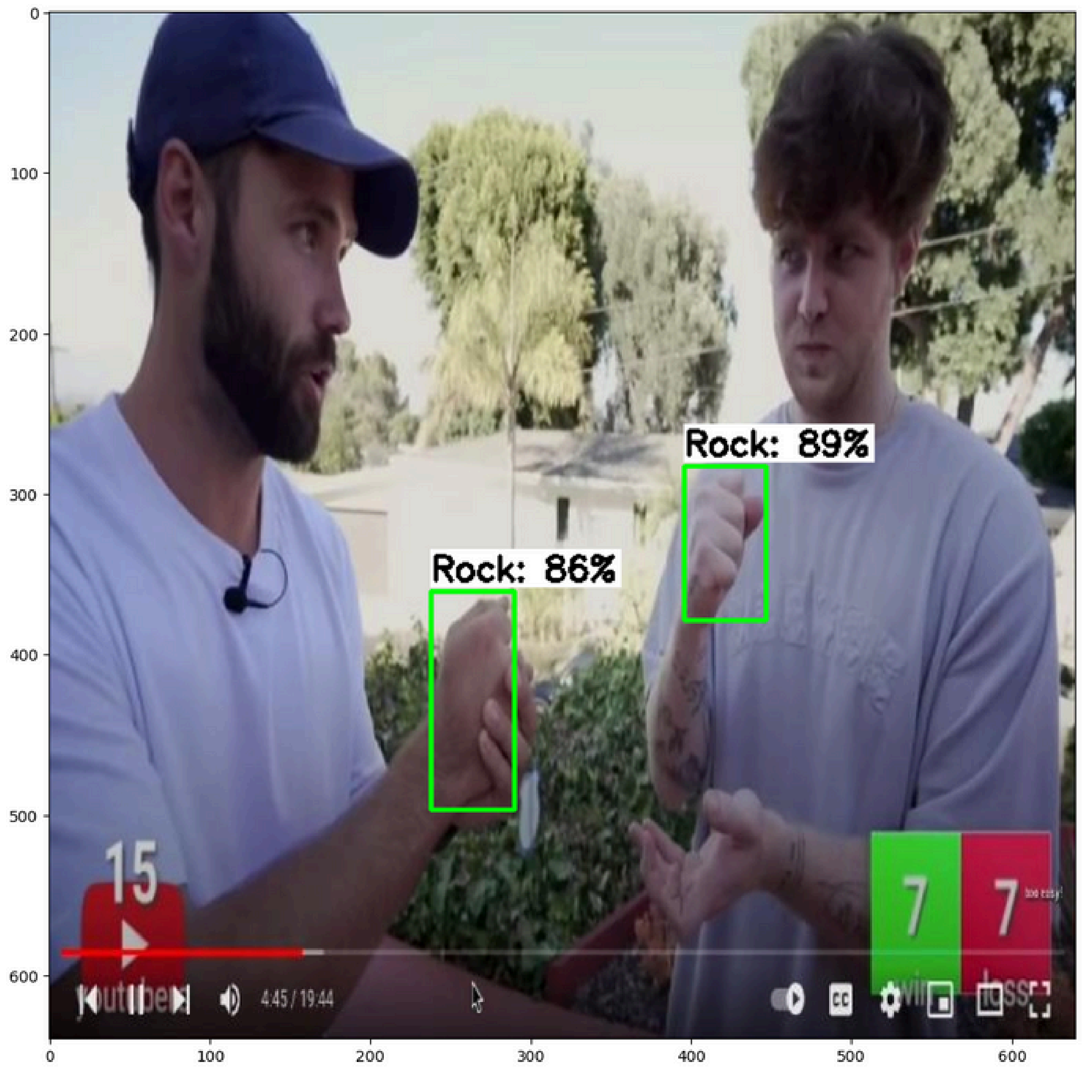
Back to Agenda Page

Testing and Evaluation

Visual Results

YOLO (You Only Look Once)

SSD (Single Shot Multibox Detector)



Back to Agenda Page

Discussion of Challenges

Challenges

- A real challenge for such models is that blur in images drastically impairs their capacity to accurately predict and detect objects.
- YOLOv8 tends to be more computationally intensive, while MobileNet is optimized for extreme resource efficiency. The whole process of training both can be a chore, specifically with small hardware and also trying to optimize the hyperparameters of each model.
- Such box annotations have to be very accurate, and small mistakes do impact the performance of models. In the case of complex scenes, labelling is a very painful process and is prone to lots of errors.

Back to Agenda Page

Discussion of Challenges

Learnings

- YOLO truly excels where speed matters.
- SSD is lighter in design; hence, it is more advantageous for deployment on small devices.
- In terms of speed and accuracy, YOLO is better than SSD.
- Any detection model largely depends on the high-quality, diverse training data; thus, careful selection and preparation of the dataset are an important prerequisite.

Back to Agenda Page

Conclusion and Next Steps

Conclusion

- The evaluation clearly shows that YOLO works better than SSD with much higher mean Average Precision (mAP) scores and higher precision and recall. Due to this, YOLO is a safe bet for scenarios where precision matters. SSD, due to its lightweight nature, it is well-suited to deployment on low-end hardware, even if it doesn't quite match YOLO's overall success. Therefore, SSD becomes a viable choice in those situations when efficiency is paramount and computing resources are scarce. The decision between YOLO and SSD mainly depends on the specific requirements of the application, maintaining a balance between the restrictions imposed by hardware capabilities and the demand for precision.

Back to Agenda Page

Conclusion and Next Steps

Next Steps

- Strategies to improve object detection models such as YOLO and SSD would include data augmentation techniques that expand the training dataset, model architecture improvements, such as transfer learning, enhancement of post-processing techniques, real-time optimization, and continuous evaluation. Hybrid models will combine the strengths of YOLO and SSD. Systematic hyperparameter optimization can enhance convergence and performance. Transfer learning accelerates training on larger datasets and advanced training techniques such as ensemble learning and curriculum learning may improve results. Techniques for post-processing include improvement in the Non-Maximum Suppression algorithm, reducing false positives, and improving detection quality. Real-time optimization and continuous evaluation improve model performance in resource-constrained environments.

Back to Agenda Page

A

Thank You for Listening

Back to Agenda Page