Cat Detection using TensorFlow Object Detection API

UNIVERSITY OF TURKU

Topic: Build a cat detection model using TensorFlow 2 Object Detection API

Author: Ha Do

Tools used:

TensorFlow 2 Object Detection API

Roboflow (labeling)

TensorBoard (training observation)

Anaconda Virtual Environment

CUDA 11.2, CuDNN 8.1.0 (for GPU acceleration)

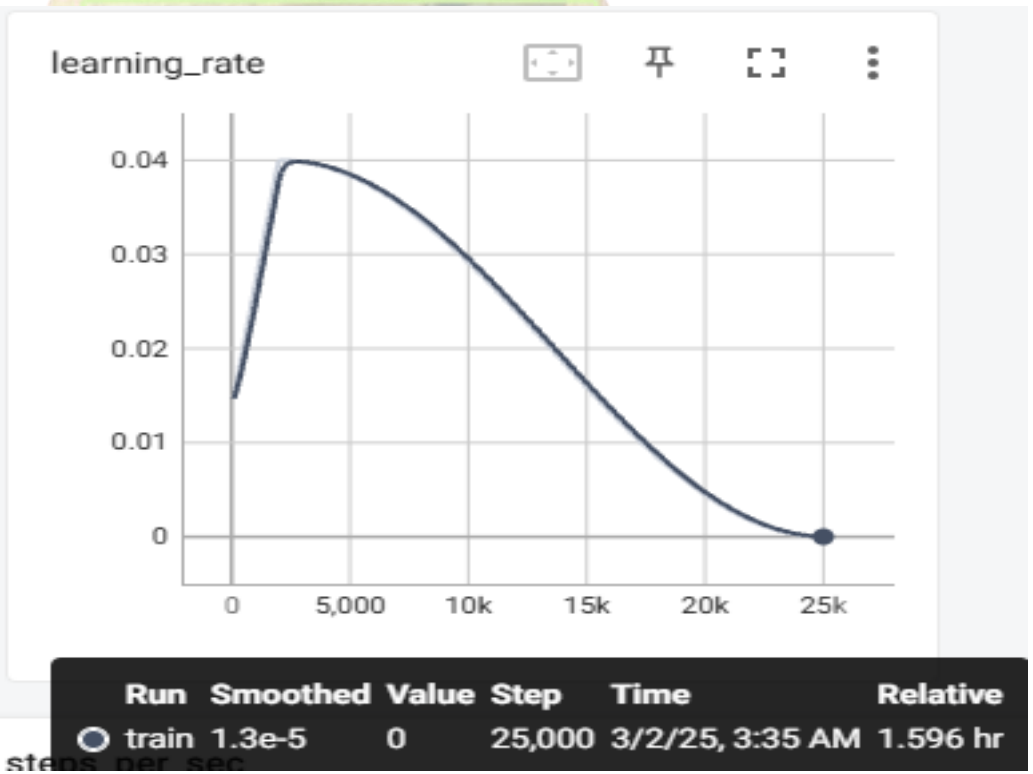
Dataset

Data source:

- 200 cat photos were collected from Google
- 150 training images, 20 validation images, 30 test images
- Format: JPG, Pascal VOC label (.xml)

Data preparation process:

1. Collect cat images from Google.
2. Label the bounding box using Roboflow.
3. Convert data to TFRecord for use with TensorFlow.



Model training

- **Training process:**

Usage model: Faster R-CNN with ResNet101 backbone

- Download the Faster R-CNN model with ResNet101 640 x 640 pre-trained model from the COCO dataset.
- Edit the pipeline.config file to match the dataset.
- Train the model on the cat dataset (~25,000 steps).
- Save the model and test the inference on the test set.
- Learning Rate gradually decreases → Stable optimization process.

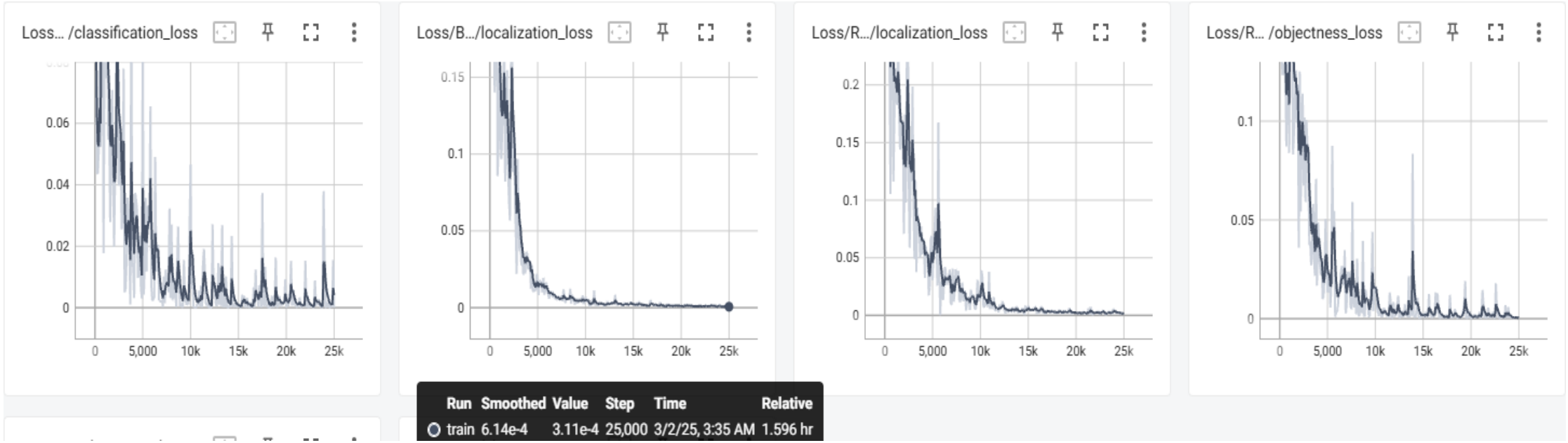
- **Training time**: 1 hour 36 minutes

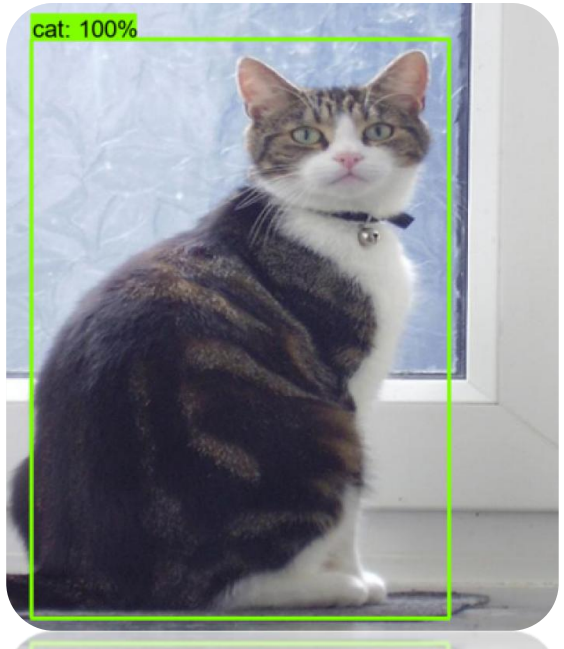
Result & TensorBoard

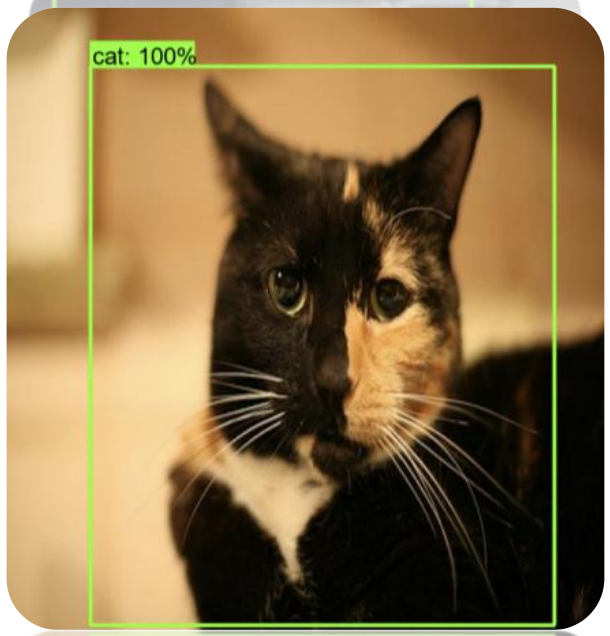
- **Inference on the test set:** Bounding box displays the correct cat position with high accuracy (100%)

- **TensorBoard Training Loss:**

Classification Loss, Localization Loss, Objectness Loss all gradually decrease → Good learning model.







Conclusion

Summary of results:

- The cat detection model is accurate with 100% confidence.
- TensorBoard shows a good learning model and stable loss reduction.
- Bounding box is clearly displayed in the test image.
- TensorBoard has been integrated into Notebook to track training loss.

Future improvement:

- Collect more data to improve accuracy.
- Experiment with other models such as RetinaNet, EfficientDet.
- Use Augmentation to increase image diversity.