Project 2: The classification contest

Identifying the problem:

- Identifying and tracking big felines through image classification aids in their conservation by providing insight into their population, distribution, and behavior, and a classification/object detection model can help us to complete the task more effectively.

Solution: Object detection / classification

- To address the problem, a two-fold approach involving fine-tuning of object detection and classification models using a dataset of big cats was implemented.
- The output models were then used for real-time big cat identification in video streams. This two models' results (mean average precision) were then compared in order to provide with the answer of which model was most effective.

Model 1: You Only Look Once (YOLO) V5

- YOLO v5, short for "You Only Look Once," follows a single-shot detection approach. Divides an image into a grid and predicts bounding boxes and class probabilities for objects within each grid cell.
- It is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection, instance segmentation and image classification tasks.[1]
  - **Install Dependencies:** Clone the YOLOv5 repository from GitHub and install the required dependencies.
  - **Download a Dataset:** Install the roboflow library and authenticate using an API key. After that, download the dataset of big cat images from Roboflow.[2]
  - **Model Configuration:** Read dataset information from a YAML file and defines YOLOv5 model architecture in a YAML file.
  - **Training:** Train YOLOv5 model on the custom dataset for 20 epochs and save training results and weights.
  - **Evaluation:** Training graphs and metrics are saved for evaluation on a ".csv" file. The evaluation method was the mean average precision (mAP).
- **Inference with the video:**

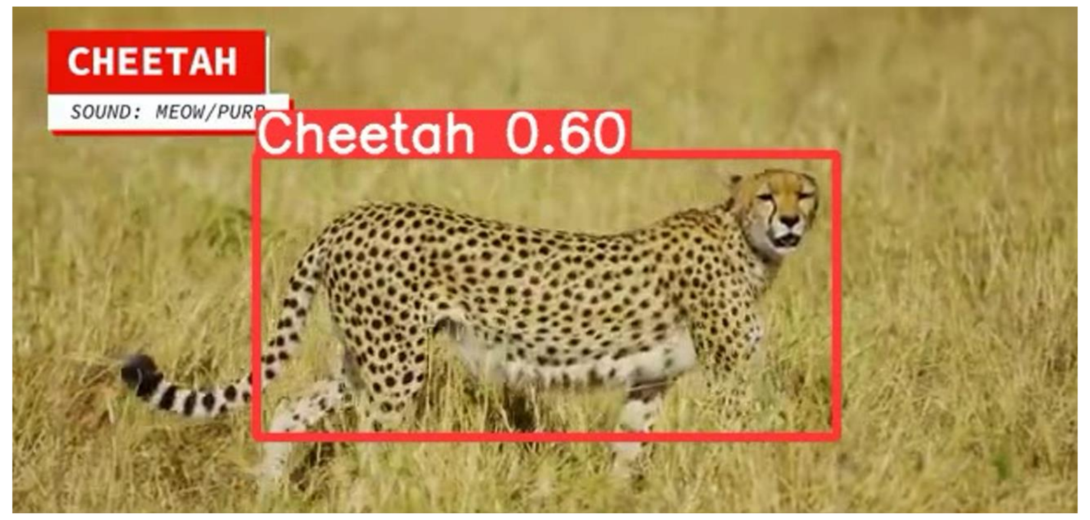

Fig 1. Frame extracted from the result video after doing inference with the fine-tuned YOLOv5 model.

Model 2: Faster R-CNN

- Faster R-CNN is a two-stage object detection model comprising a Region Proposal Network (RPN) and a Fast R-CNN detector. The RPN proposes potential object regions using anchor boxes, and the Fast R-CNN refines and classifies these proposals.
- Faster R-CNN is known as being a slower model than some other choices (like YOLO or MobileNet) for inference but in return is more accurate.[3]
  - **Install dependencies:** Clone a Faster R-CNN PyTorch training repository and install required packages.
  - **Dataset Configuration:** Download and unzip the big cat dataset from Roboflow using a provided link.[2] Create a YAML file specifying dataset directories, class names, and other settings.
  - **Training:** Execute the training script (train.py) using the specified custom dataset configuration. Train the Faster R-CNN model with the ResNet50 backbone for 5 epochs.
  - **Evaluation:** The trained model's performance on the validation set is later saved on a ".csv" file. The evaluation method was the mean average precision (mAP).
- **Inference with the video:**

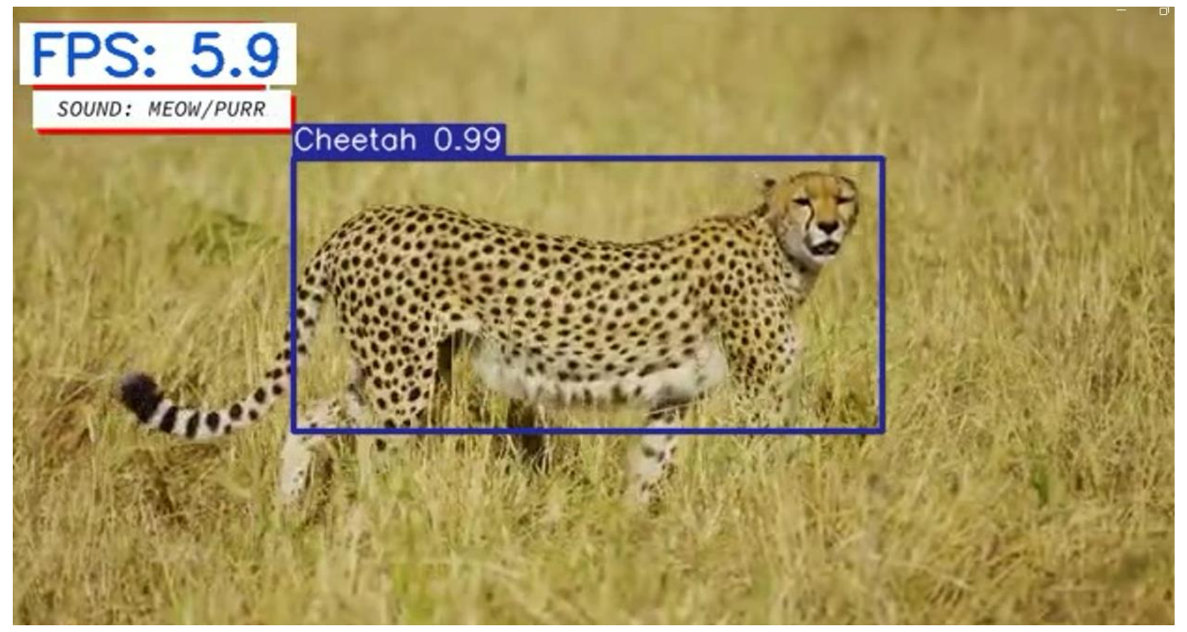

Fig 2. Frame extracted from the result video after doing inference with the fine-tuned Faster R-CNN model.

Results:

- Both models generated individual CSV files containing the mean Average Precision (mAP) scores obtained during their training processes. The data extracted from these files played a crucial role in facilitating a comprehensive and quantitative comparison between the two models, since the contest between Faster R-CNN and YOLOv5 was informed by the insights derived from these mAP scores.
- After the data was extracted from their respective CSV files, the maximum value obtained during training was singled out and then compared with the other model's max mAP.
- The results show that the Faster R-CNN model obtained a top mAP score of 97.43%, while the YOLOv5 model obtained a 89.77% score.

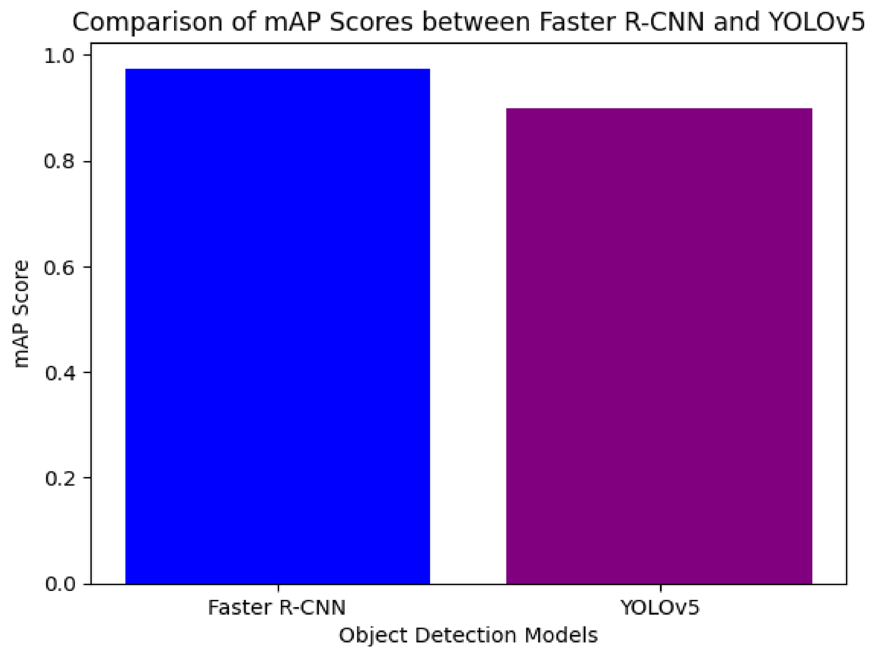

Fig 3. Ablation graph comparing the Mean Average Precision (mAP) between the fine-tuned models.

```
Faster_rcnn results = 0.9743891536344732
yolov5 results = 0.89774
The model Faster R-CNN obtained a better mAP score: 97.44%
```

Fig 4. Maximum mAP score extracted from each model's CSV file, comparison and winner results printed in the code.

Conclusion

- In the classification contest between Faster R-CNN and YOLOv5, the results favor the Faster R-CNN model, which demonstrated superior performance. The Faster R-CNN model achieved a mean Average Precision (mAP) score of 97.43%, surpassing the YOLOv5 model's score of 89.77% by a difference of 7.66%. This substantial difference proves the effectiveness of Faster R-CNN in accurately classifying big cats in the provided dataset. This results make Faster R-CNN the winning model in this comparative evaluation.

References

- [1]. Ultralytics. "YOLOV5." PyTorch, https://pytorch.org/hub/ultralytics_yolov5/. Accessed 19 November 2023.
- [2]. "BIG CATS Computer Vision Project." Roboflow Universe, https://universe.roboflow.com/xyz-mithp/big-cats. Accessed 18 november 2023.
- [3]. "What is Faster R-CNN?" Roboflow, https://roboflow.com/model/faster-r-cnn. Accessed 20 november 2023.