Low-Power Person Detection for UAV Search and Rescue Operations

Tiny Machine Learning

Students: Jakub Schwenkbeck, Timon Coucke, Gabriele Pattarozzi

Supervisor: Rahim Rahmani

Introduction

UAVs are often deployed in search and rescue operations due to their ability to quickly cover large areas. However, live video streaming from UAVs increases power consumption a lot, reducing flight time and mission effectiveness. To overcome this, on-board person detection models can be used, enabling the UAV to transmit video only when a potential person is detected.

The project's aim is to design and benchmark lightweight, low-power object detection models that balance accuracy, speed, and efficiency. The project will show insights into the trade-offs between model complexity and real-time inference performance in UAVs.

Problem Statement

In search and rescue operations, UAVs can be used to cover large and often inaccessible areas. The problem is that transmitting live video consumes a lot of power, limiting flight duration due to short battery life. There is a need for fast and energy-efficient on-board person detection models that can recognize humans in real time, allowing the UAV to send data only when necessary. This problem is significant because improving model efficiency directly improves UAV flight time, which increases coverage area and improves the chances of locating missing persons while keeping energy consumption to a minimum. That's why developing a balance between model size, accuracy, and inference speed is critical for practical, low-power SAR applications.

Research Questions and Objectives

Our project aims to investigate the following:

1. Which lightweight person detection model (e.g., SSD-MobileNet, YOLO-Tiny) provides the best balance of accuracy, speed, and power efficiency for Edge UAV deployment?
2. How can pruning and quantization techniques from our lecture be applied to reduce model size and energy consumption without significantly losing detection accuracy?
3. How do different edge hardware platforms (e.g., Coral Edge TPU, OAK-D, CPU) affect model performance, inference speed, and power draw?
4. What is the best combination of model and hardware for a solar-powered UAV performing real-time search and rescue operations?

Methodology

Our approach will follow this structure:

1. **Model selection**: consider trade-offs between model size, accuracy, and inference speed of different lightweight person detection models (such as SSD-MobileNet and YOLO-Tiny).
2. **Model optimization**: apply the learned methods to reduce the model size for edge deployment on the selected model. This includes **pruning** to remove redundant or less important network connections, reducing model size. We will also utilize **quantization** to convert model weights to lower-precision formats (8-bit) to enhance inference speed and reduce memory consumption. Finally, we will fine-tune the optimized model to maintain acceptable detection accuracy.
3. **Hardware Benchmarking**: to actually verify and further research what we have achieved, we then deploy the optimized models on various edge platforms such as Coral Edge TPU, OAK-D, and standard CPUs. For some metrics, we will acquire TestData from a public Dataset fitting our problem domain. We measure performance metrics, including:
    - **Mean Average Precision (mAP)** for detection accuracy
    - **Frames Per Second (FPS)** for detection speed
    - **Intersection over Union (IoU)** for possible bounding boxes
    - **Power consumption** for energy efficiency
    - **Model size** in KB
4. **Solar-powered UAV design recommendation**: finally, we want to compare model performance across different hardware setups and identify the optimal combination of model and hardware for a solar-powered UAV performing SAR operations.

Tools, technologies, and frameworks

For this project, we'll be using different tools from our lectures and labs and some of our personal experience. We will write code in **Python** and use **TensorFlow** and **TensorFlow Lite / Tiny** to build and optimize our detection models. We also plan to use **Edge Impulse** from Lab 1 to help streamline model development and deployment. For collaboration and sharing, **GitHub** and other platforms will allow us to manage our work within the team and, when appropriate, make it available to the public. We will search **HuggingFace** and **Kaggle** for Datasets and models.

Timeline

Task	**W38**	**W39**	**W40**	**W41**	**W42**	**W43**	**W44**	**W45**
Project plan	20/9							
Model selection		28/09						
Model optimization				12/10				
Hardware benchmarking					19/10			
Solar-powered UAV design						26/10		
Project report and code							30/10	

Expected Outcomes

- For Edge deployment optimized lightweight person detection models.
- Benchmark results for accuracy, speed, power use, and model size across edge devices.
- A recommendation for the most suitable model-hardware setup for UAV search and rescue.
- A final report and codebase showing the process and results.

The impact of these outcomes is to allow for longer UAV flight times, better search coverage, and faster detection of people. This can directly improve the effectiveness of UAVs in real-world rescue operations.

Challenges

Some of the challenges we might face could be the lack of specific datasets related to the project and the unavailability of actual hardware (UAVs and cameras) to capture them. To face this, we could rely on public datasets like RoboFlow or Kaggle. Another challenge will be keeping the accuracy high while at the same time optimizing the models aggressively. To face this challenge, we will use the algorithms and techniques explained in the lectures to try to maintain both.

The main risk of the project will be to fail in finding a suitable solution for obtaining real-time inference on highly detailed images, while having to deploy it in a very power-limited vehicle like a solar-powered UAV.

References

- Edge Impulse. Available at: https://edgeimpulse.com/
- TensorFlow Official Website. Available at: https://www.tensorflow.org/
- TensorFlowLite GitHub Repository. Available at: https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite
- TensorFlow Lite Micro GitHub Repository. Available at: https://github.com/tensorflow/tflite-micro
- Kaggle. Available at: https://www.kaggle.com/
- HuggingFace. Available at: https://huggingface.co/
- Lecture Slides
- Lab Notebooks