1. Adaptive Real-Time Visual Processing Under Latency and Resource Constraints

Supervisor: Oleksandr Muliarevych

Problem: Autonomous systems must react to changes in a visual environment despite variable image-processing, communication and control delays. Processing every frame may waste computational resources, while dropping or delaying frames may lead to outdated decisions and control errors. Under what conditions can adaptive visual processing improve reaction time and resource efficiency without significantly degrading control performance?

Available materials and environment: Open-source simulation environments such as Gymnasium CarRacing, DonkeyCar or Webots; OpenCV; lightweight CNN or YOLO models; ONNX Runtime; and configurable simulation of processing delays, frame loss and communication latency. Optional experiments may use Arduino-based controllers or NVIDIA Jetson devices, subject to availability.

Expected outcomes and evaluation criteria: A controlled simulation-based investigation comparing fixed-rate processing with one or two selected adaptive strategies. Teams will analyse the relationship between data freshness, computational load and control quality. Evaluation will include end-to-end reaction latency, control accuracy, missed deadlines, collision or trajectory errors, processing throughput, and CPU/GPU/memory utilisation.

2. Data-Efficient UAV–Satellite Geo-Localization Through Synthetic-to-Real Training

Supervisor: Roman Ilechko

Problem: UAV geo-localization requires matching aerial images with georeferenced satellite imagery. Collecting and annotating real UAV images is costly, while lightweight models must operate under computational and latency constraints. Synthetic training data may reduce the need for real observations, but their effectiveness depends on the training strategy and their ability to represent real-world conditions. How can synthetic data reduce real-data requirements while preserving localization accuracy and generalization to previously unseen geographic regions?

Available materials and environment: Python, PyTorch with CUDA, an NVIDIA GPU workstation, synthetic and real UAV imagery where available, and the proposed UAV platform with its Radxa ROCK 5B companion computer. Experiments should prioritise offline training and evaluation; onboard inference can be conducted where feasible.

Expected outcomes and evaluation criteria: A controlled comparison of selected synthetic-to-real training strategies and data proportions. Teams will analyse the relationship between the quantity of real training data, localization performance and generalization to geographically separate test regions. Evaluation will include localization accuracy, the amount of real training data required, inference latency, processing throughput and computational resource use.

3. Real-Time Smartwatch Sensor Intelligence: How Does Data Quality Affect Activity Recognition?

Supervisor: Olena Pavliuk

Problem: Smartwatch sensor measurements may contain noise, outliers, missing values and corrupted segments, which can reduce the reliability of human activity recognition. Data imputation and anomaly handling may improve recognition, but they can also introduce errors or increase computational cost. Which data-quality problems have the greatest impact on activity recognition, and which preprocessing strategies improve robustness under realistic conditions?

Available materials and environment: Existing smartwatch measurements stored in the Microsoft Azure cloud database, two Samsung Galaxy smartwatches for additional data collection where necessary, the existing data collection application, Python and Google Colab. The initial investigation should establish the available sensor modalities, sampling rates, activity labels and data-quality issues.

Expected outcomes and evaluation criteria: A controlled investigation of one selected data-quality problem and one or more corresponding preprocessing strategies. Teams will compare the results with a baseline that does not use the proposed intervention. Evaluation will include activity-recognition accuracy and macro-F1, per-class precision and recall, robustness under missing or corrupted measurements, inference latency and computational resource use. MAE or RMSE may be used where the selected task involves reconstructing missing measurements.

4. When Does Sensor Fusion Pay Off? Explaining Sensor Interactions in Lightweight Industrial AI

Supervisor: Myroslav Mishchuk

Problem: Industrial AI systems combine multiple sensor streams, but adding more sensors does not always improve decision quality. Depending on operating conditions, sensor streams may provide complementary information, duplicate existing information, or conflict with each other. How can we identify these situations and use this knowledge to make lightweight AI models more efficient and robust without significantly degrading their performance?

Available materials and environment: Public multimodal datasets, including nuScenes, RADIATE, SeeingThroughFog and MultiBench, as well as simulation tools such as NVIDIA Isaac Sim. Python-based machine learning frameworks and XAI methods for analysing sensor interactions can be used. Experiments can be conducted remotely; access to a real AGV is optional.

Expected outcomes and evaluation criteria: An experimental study identifying when sensor streams are complementary, redundant or conflicting. Teams will compare selected fusion strategies, evaluate the consistency of XAI explanations with ablation results, and investigate whether selected sensor streams can be removed or simplified. Evaluation will include prediction quality, fusion gain, explanation fidelity and stability, inference latency, memory consumption and, where feasible, energy use.

5. Agentic RAG: How Can We Find Sufficient Evidence at Lower Cost?

Supervisor: Pi-Wei Chen

Problem: Answering complex questions often requires combining evidence from multiple sources. A retrieval agent must decide whether to search again, follow a source link, inspect additional context or stop and answer. Stopping too early may produce unsupported answers, while unnecessary retrieval increases computational cost. How can an Agentic RAG system balance answer and evidence quality against the number of actions, token usage and response time?

Available materials and environment: The existing Agentic-RAG repository, its SEARCH, EXPAND, READ and FINISH actions, and the HotpotQA dataset with answer and supporting-fact annotations. Experiments can be conducted in Python using a suitable small open-weight language model and a manageable benchmark subset.

Expected outcomes and evaluation criteria: A controlled comparison of the baseline retrieval policy and at least one alternative strategy. Teams will analyse when additional retrieval actions improve the evidence and when they introduce unnecessary cost. Evaluation will include Exact Match or token F1, supporting-fact recall or F1, number of

actions and model calls, token usage, median and P95 response time, and unsupported-answer frequency.

6. Learning to Grasp Diverse Objects: How Does Training Variety Affect Generalization of a Reinforcement Learning Policy?

Supervisor: Jakub Musiał

Problem: An industrial robot must pick up objects that differ in shape, size, mass and surface friction. A reinforcement learning policy trained on a narrow set of objects may fail on objects it has not seen, while training on a broad, randomized set increases training cost and may slow down or destabilize learning. How does the variety of objects and physical parameters seen during training affect the ability of an RL grasping policy to generalize to previously unseen objects, and at what cost in training effort?

Available materials and environment: NVIDIA Isaac Sim with Isaac Lab, a simulation model of the Universal Robots UR10e arm with a two-finger parallel gripper, a set of object models (geometric primitives and selected objects from public datasets such as YCB), Python, PyTorch, an RL library supported by Isaac Lab (e.g. RSL-RL or skrl.

Expected outcomes and evaluation criteria: A controlled comparison of a baseline policy trained on a single object with at least one strategy that increases training variety, such as randomization of object geometry and physical parameters or a curriculum. Teams will analyse the trade-off between training variety, learning cost and generalization. Evaluation will include grasp-and-lift success rate on training objects and on a held-out set of unseen objects, the number of environment steps and wall-clock time needed to reach a given success rate, grasp stability under an external disturbance, and variance across random seeds.

Scope and constraints: One robot and gripper configuration, one task (grasp an object and lift it to a target height), and observations taken from the simulator state (object pose, joint states) rather than from camera images. Vision-based observations are an optional extension. To support later transfer to the real UR10e, the policy should command joint or end-effector position targets within the real robot's joint, workspace and speed limits. Testing on the physical robot is not part of the project.