CS M213A / ECE M202A (Fall 2025)

Context-Aware Hybrid Object Detection for Autonomous Vehicle Perception

Hee Jean Kwon

Motivation and Objectives

- Autonomous Vehicles need to detect objects quickly and accurately
- Running powerful vision models on the vehicle reduces latency but limited by hardware -> using the cloud gives better accuracy but dependent on network condition
- A runtime optimizer that can decide when to use local vs cloud processing
  - Faster and safer decision for autonomous vehicles
  - More efficient use of limited resources
- Goals
  - Design a context-aware decision system that chooses between local and cloud models in real time, using scene complexity and network conditions to balance accuracy and resource usage under a time budget

Related work

- Edge-Cloud Offloading
- Feasibility of Cloud-Assisted Autonomous Vehicles
  - Cloud GPUs run detection models 4–19× faster than Jetson Orin
- Context Aware Local computation
  - Fully local inference but dynamically switches configurations using context to improve streaming accuracy
  - Switch EfficientDet depends on how difficult the scene is

Technical Approach and Novelty

- Offloading between cloud and local using context
- Use multi-modal context (scene, vehicle, and system) to predict best model
- Data-driven policy instead of hand-tuned rules
    - Instead of rule based, CNN is trained using data
- Investigate different decision architectures to identify optimal model selection

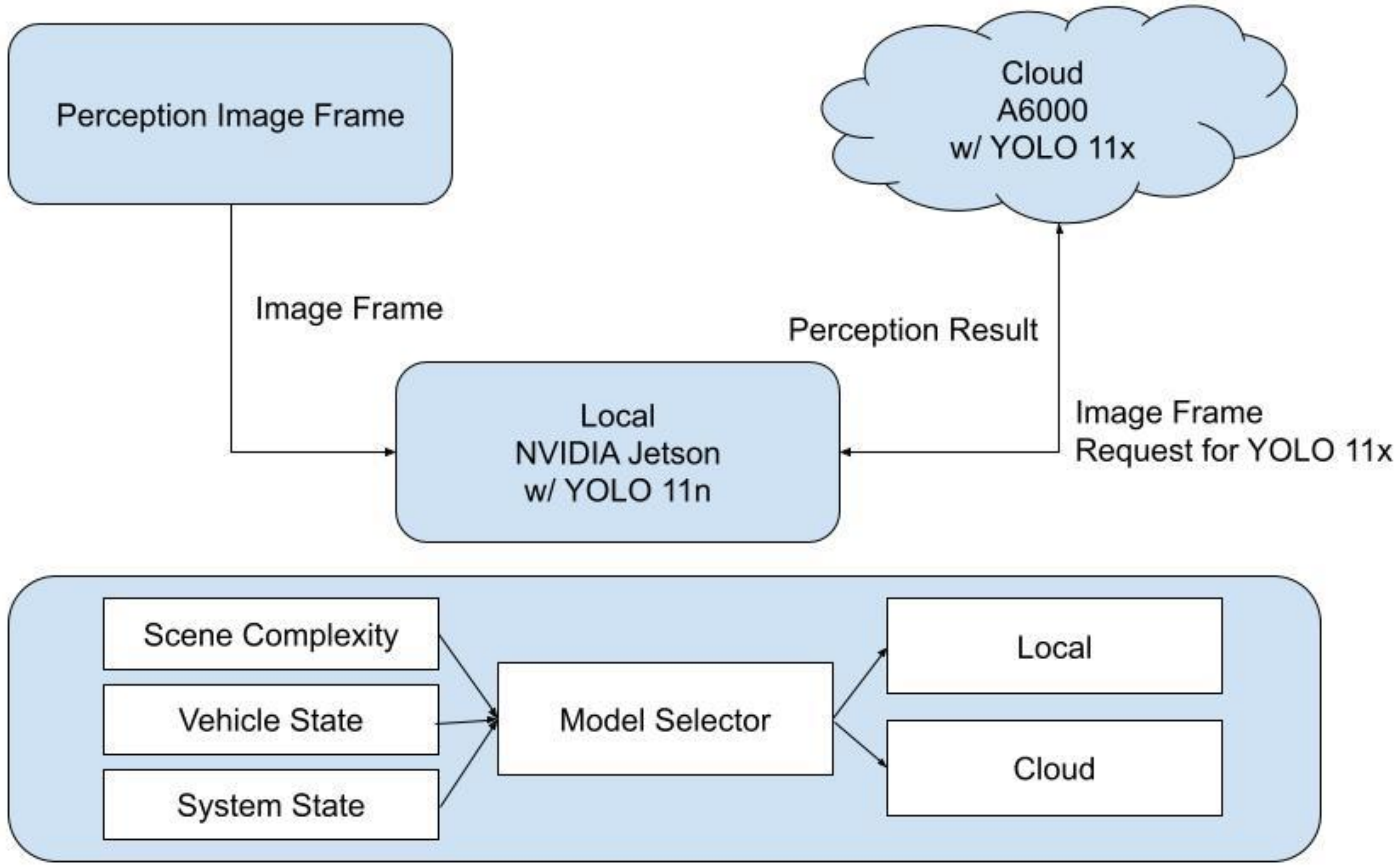

Figure 1 System overview

Why Hybrid

- Cloud computing is expensive
  - NVIDIA H100 rental cost $2.49 per hour
  - Network cost in the US
    - 50Mbps upload, $16.88 per house
  - Total $19.37 per hour
- Despite cloud's clear benefits, it is necessary to find how much it can be offloaded

Methods

- Data Set
  - Waymo Perception Data set
- Platform
  - NVIDIA Jetson Orin for local processing
  - A6000 for cloud processing
- Use Machine Learning Model to design a model selector
  - Scene complexity: Number of vehicles and pedestrians, brightness
  - Vehicle state: ego vehicle speed
  - System state: Network reachability and Cloud availability

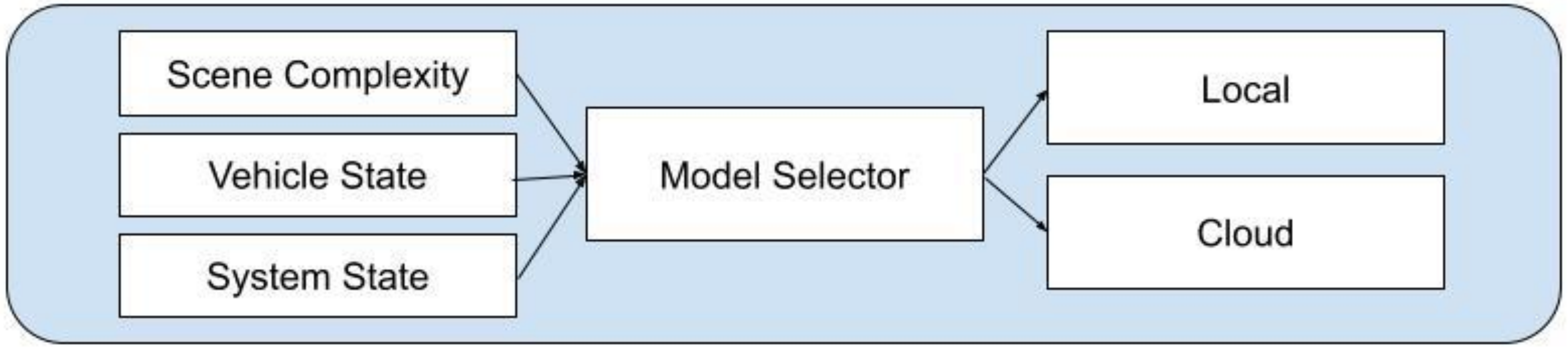

Figure 2 Model Selector Overview

Usage of MLP

- Inputs
  - # of vehicles, # of pedestrians, brightness, and ego vehicle speed
- Outputs
  - 0 = simple scene, 1 = complex scene
- Misclassified scenes are on boundary
- Feature Importance
  - 28% # of vehicles, 20% ego vehicle speed, 15% # of pedestrians
- Need labeled data -> no for AV

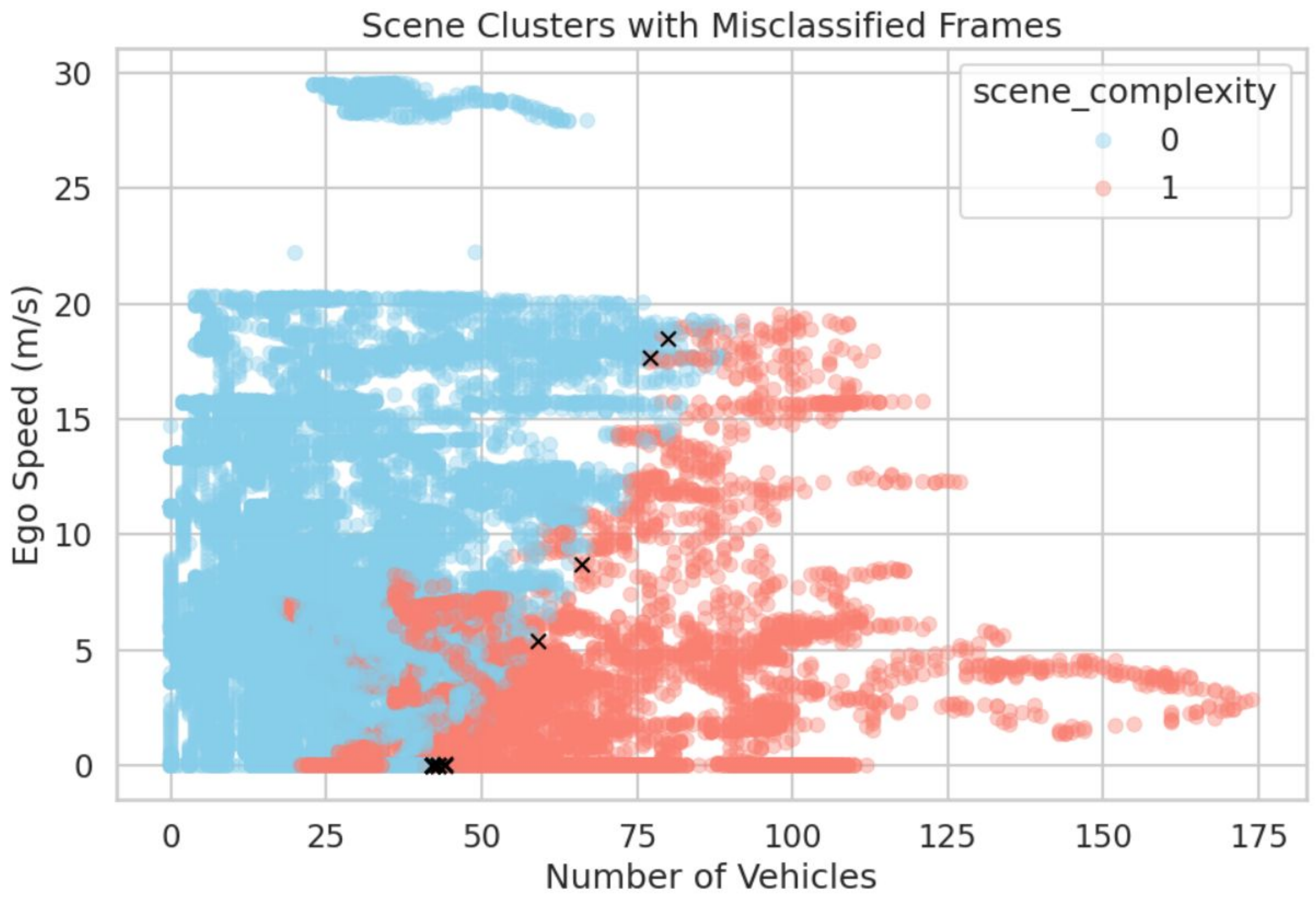

Figure 3 Scene Clusters with Misclassified Frames on MLP

Example of Simple and Complex Scenes

Usage of VLM

- Qwen2-VL with 2B parameters
- Inputs: JPEG image of one frame from Waymo dataset
- Output: Local or Cloud
- Prompt VLM to check the amount of vehicles, pedestrians, and brightness of the given image and make a decision
- Brightness has the strongest influence unlike MLP
- Its reasoning and the scenario do not match

Reasoning: The scene is complex with multiple vehicles and pedestrians, indicating a busy street, which requires processing tasks in the cloud for real-time safety and accuracy.

Figure 4 Example of an image frame with its VLM reasoning

Usage of CNN

- Input: JPEG image of one frame from Waymo dataset
- Output
  - 0 = simple -> Jetson YOLO 11n
  - 1 = complex -> A6000 YOLO 11x
- Convolutional blocks → flatten → two fully connected layers → binary output
- Simple model to achieve a quick execution
  - avg execution time of 1.267ms
- Achieves Validation accuracy 98.98%

CNN - Saliency map

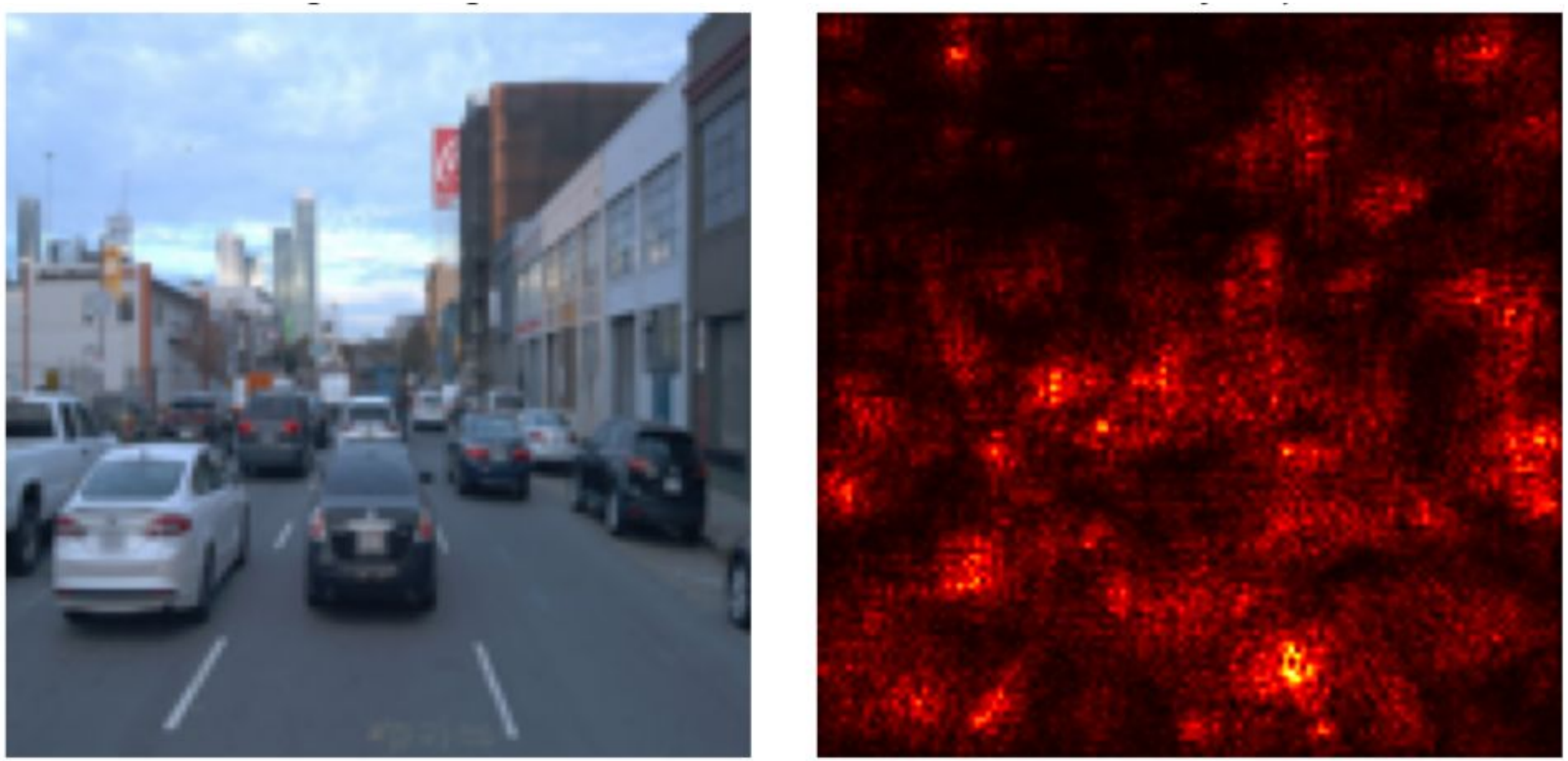

Figure 5 Saliency map for CNN

CNN Feature Correlations

- Scene complexity is strongly tied to object density
- Complexity scene score increases when there are more objects (traffic participants) in the frame
- CNN properly learned to act on crowdess (busy urban scenes)

Table 1 CNN and Feature Correlations

Scene features	Correlations
# of vehicles	0.7
# of pedestrians	0.55
Brightness	0.11

YOLO 11n vs YOLO 11x

- Execute both YOLO 11n and 11x on the dataset to compare accuracy
- For each frame, the model with the smaller absolute error (|prediction – ground truth|) was counted as more accurate.
- YOLO11x
  - closer to the ground truth in 76.2% of all frames
  - avg execution time of 28.4ms on A6000
  - significantly more accurate
- YOLO11n
  - closer to the ground truth in only 3.9% of frames
  - avg execution time of 62.6ms on Jetson Orina Nano

System Flow

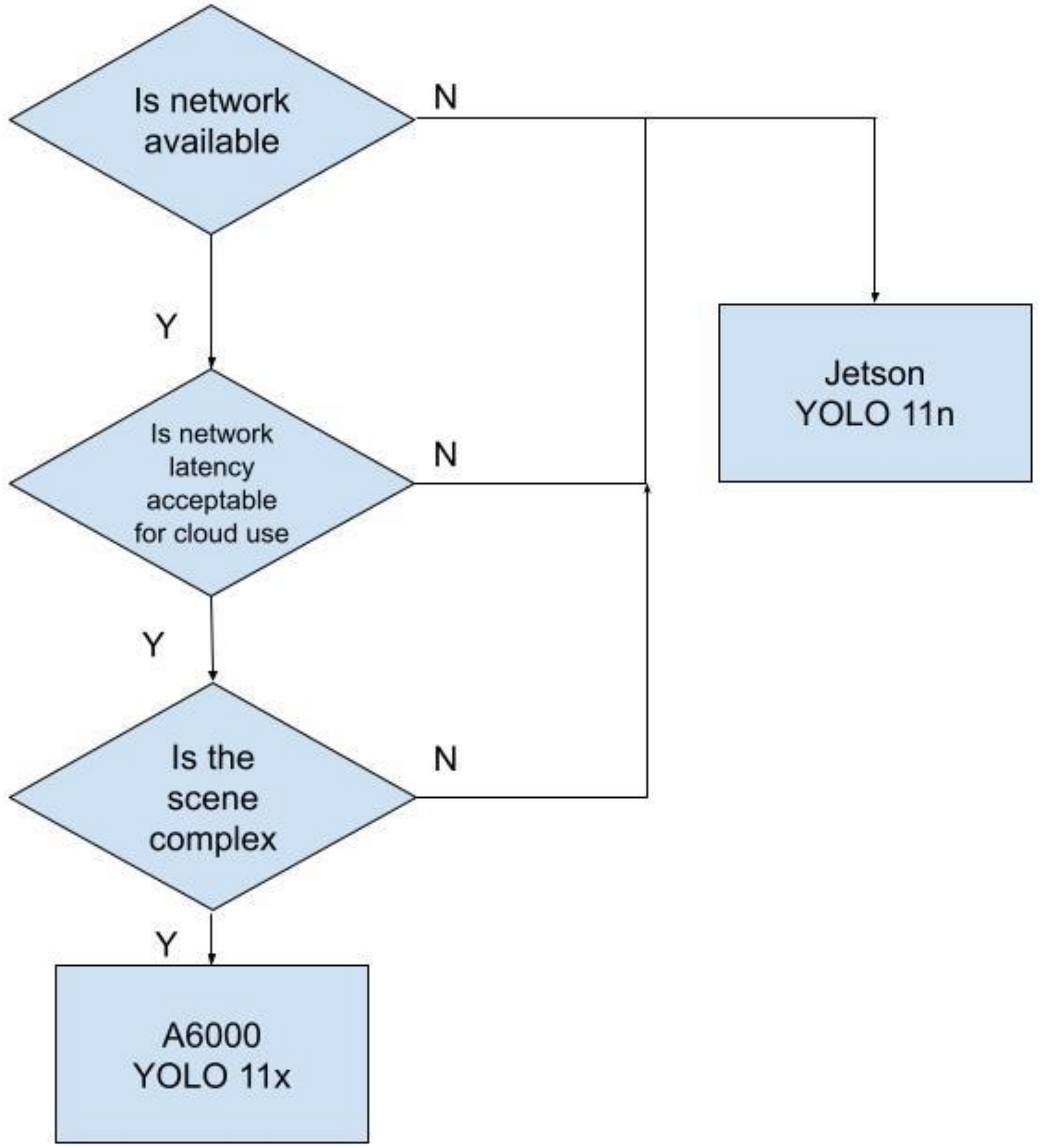

Figure 5 System Flow

Assumption for Experiment / Set Up

- Camera Frames are coming in 10 FPS
  - Each frame needs to be processed within 100ms
  - L_max = 90ms
  - L_max is compared with the latest cloud frame YOLO execution time + current network latency
- Network availability is expressed in binary
  - If the network is available, the cloud is available at its full capacity
- Use Jetson Yolo 11n result if the cloud YOLO result doesn’t arrive back in 90ms
- Total 7967 Frames used as input jpeg

Hybrid Policy Evaluation - Performance

- Total frames used in YOLO 11n : 68% of the entire frames
- Total frames used in YOLO 11x: 32% of the entire frames
- Hybrid retains ≈75% of the accuracy benefits of cloud offloading while using cloud only ⅓ of the time

Table 2 Local vs Cloud vs Hybrid Perception

Policy	Cloud %	Pedestrian Mae	Vehicle Mae
Always YOLO 11n	0	2.79	13.6
Hybrid Perception	31.6	2.17	12.16
Always YOLO 11x	100	2.09	10.58

Hybrid Policy Evaluation - Latency

- YOLO 11x is much faster computationally
- If the latest RTT + latest cloud model execution time is smaller than the maximum Latney allowed, cloud is allowed

Table 3 Latency of Hybrid Perception

	Mean (ms)
YOLO 11n on Jetson	62.1
YOLO 11x on A6000	28.4
Jetson <-> Cloud RTT (Network Latency)	8.9

Conclusion

- Hybrid policy retains most cloud-model accuracy using only ~32% cloud compute.
- Provides quantitative evidence of when cloud is worthwhile under realistic AV constraints.
- Hybrid Perception Framework on Jetson-to-cloud AV perception pipeline
  - utilizing lightweight CNN gating, latency measurement, and fallback logic
  - Support different detection models
  - Forms a foundation framework for adaptive AV perception research

Future Directions

- Integration of relationship between
  - Cloud availability vs YOLO execution speed
  - max perception latency ms allowed vs current vehicle speed
- Modeling on latency vs accuracy payoff
- End to End Simulation on CARLA