Real-Time Mapping at Traffic Intersections

Menna El-Shaer
Ph.D. Candidate, Electrical and Computer Engineering Department

Introduction

DIVING IN

Real-time Mapping

Summary

The Ohio State University

Who am I?

Computer Engineer

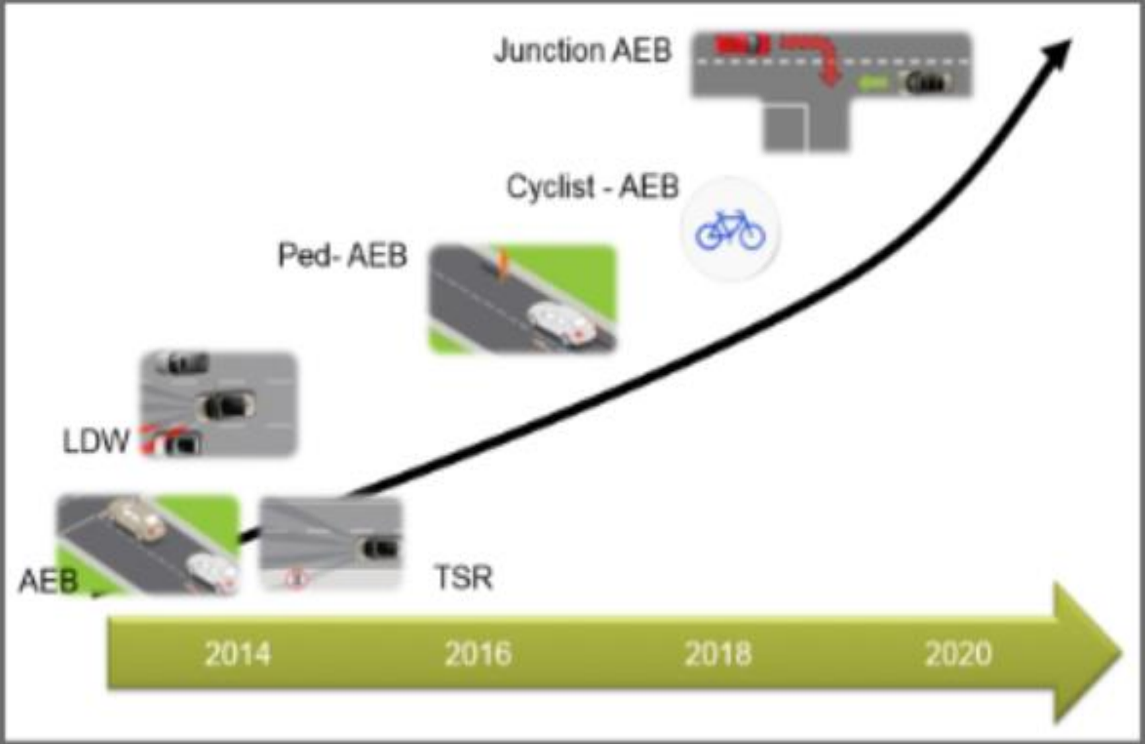
- Computer Vision
- Machine Learning
- Computer Architecture

Research Activities

NHTSA Vehicle Research and Testing Center

Center for Automotive Research: http://car.osu.edu/
Crash Imminent Safety University Transportation Center: http://citr.osu.edu/CrIS/
Transportation Research Center: http://www.trcpg.com/

Situational Awareness and Vehicle Perception



"**Perception** of elements in the environment within a volume of time and space, **comprehension** of their meaning, and **projection** of their state into the future",
Endsley 1995

Traffic Urban Intersection Challenges

vs. Freeway Driving

Safety Hazards

Geometry

Meterology

Human Decision Errors

Sensing

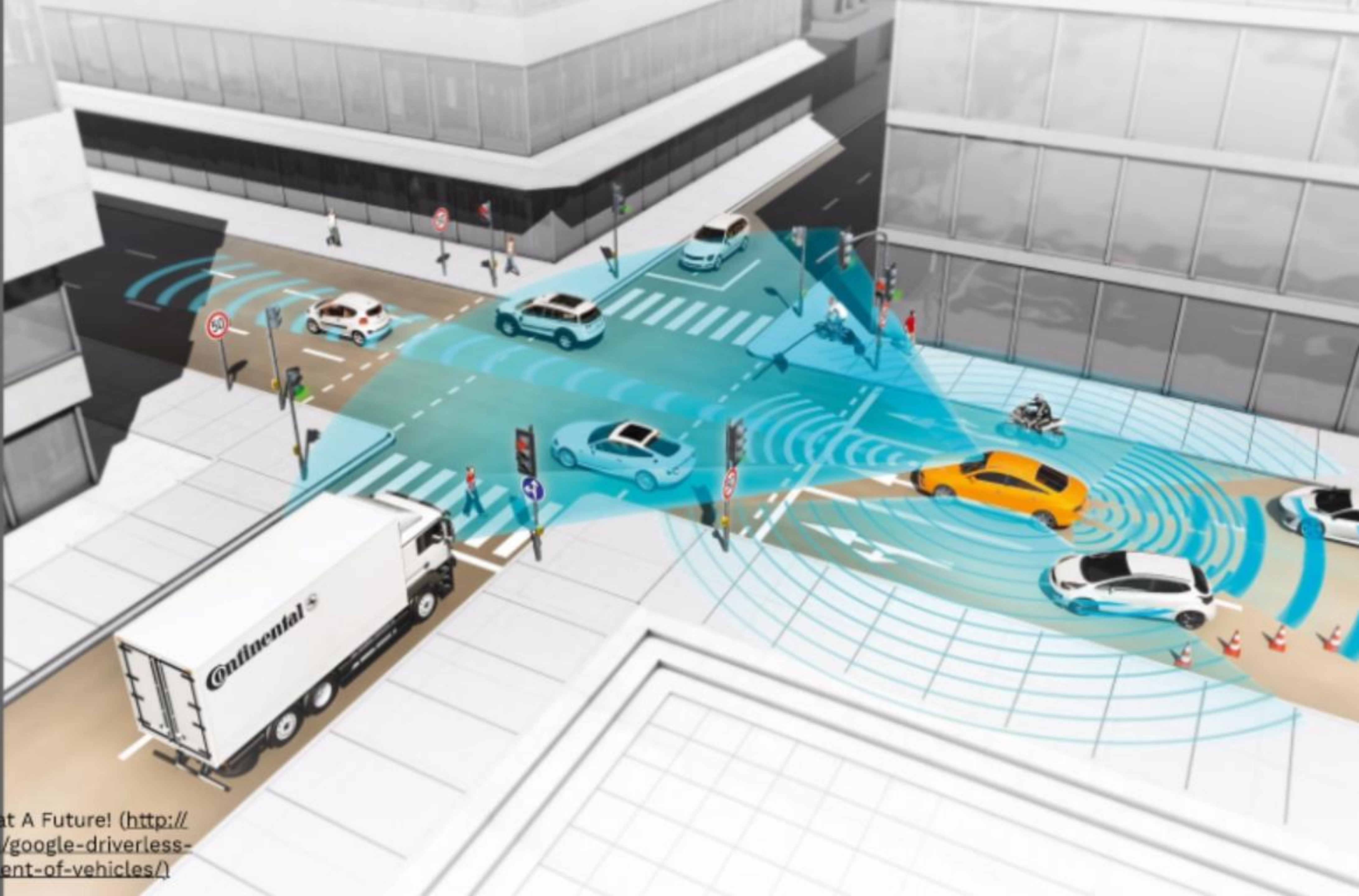
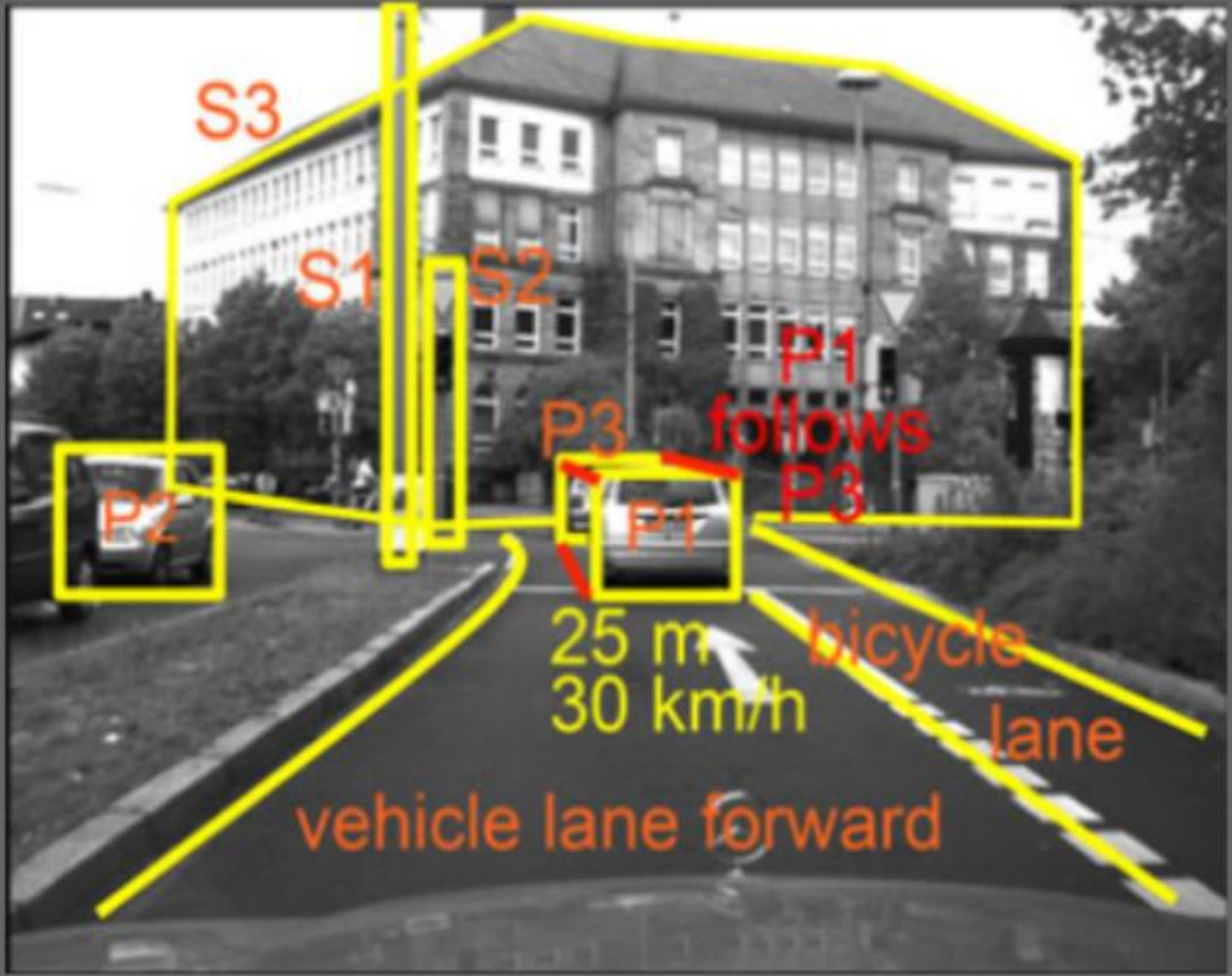


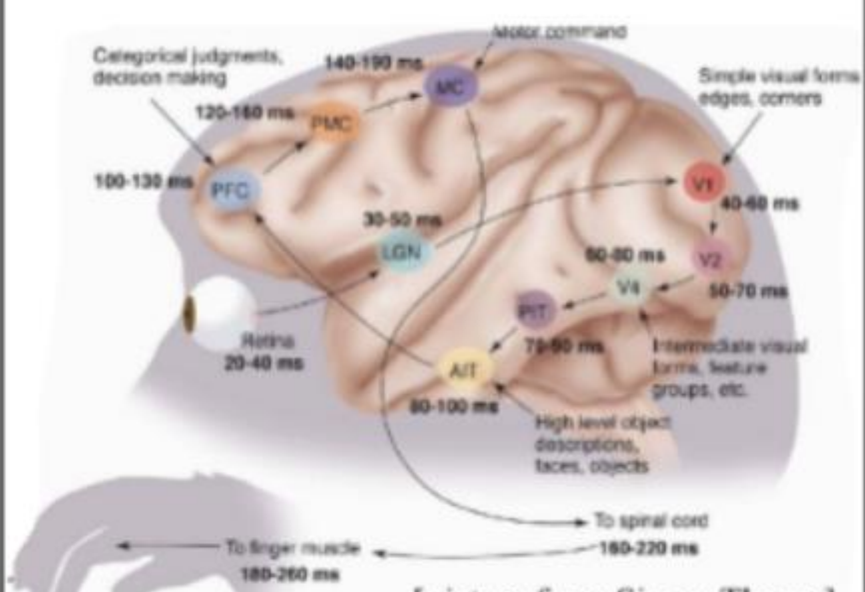
Image Courtesy of What A Future! (http://www.whatafuture.com/google-driverless-car-predicting-movement-of-vehicles/)

Scene Understanding



B. Ranft and C. Stiller, "The Role of Machine Vision for Intelligent Vehicles.", IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 8-19, Apr. 2016

Cognition



[picture from Simon Thorpe]

[Gallant & Van Essen]

Feature Extraction

Complex Non-Linear Representation

Deep Feature Learning

Hierarchical Layers

DIVING IN

Definitions

Generative Models

Probabilistic Graphical Models

NVIDIA Tegra X1 SoC

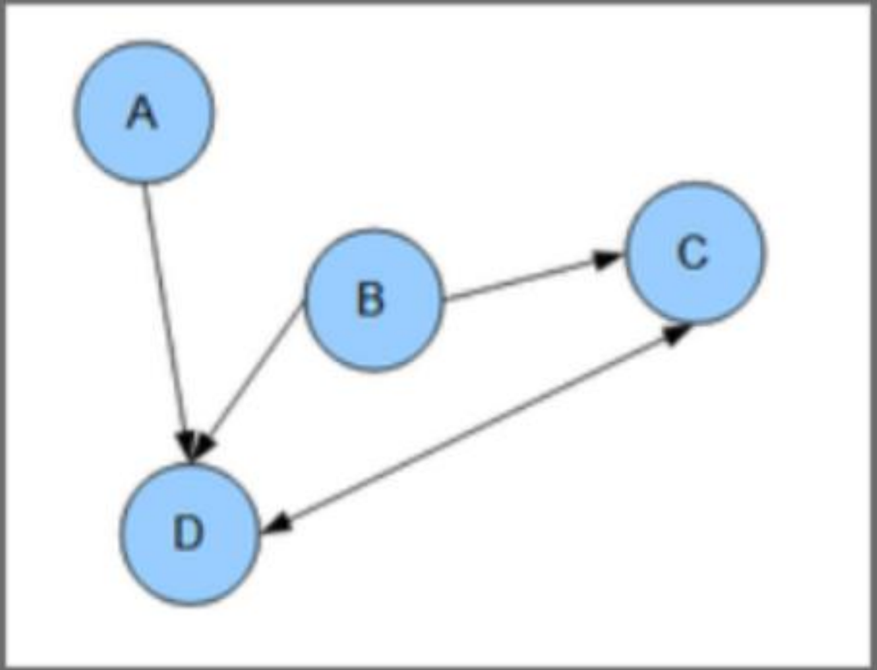
Generative Models

vs. Discriminative Models which model the conditional distribution of the target variables given the observed variables $p(y|x)$

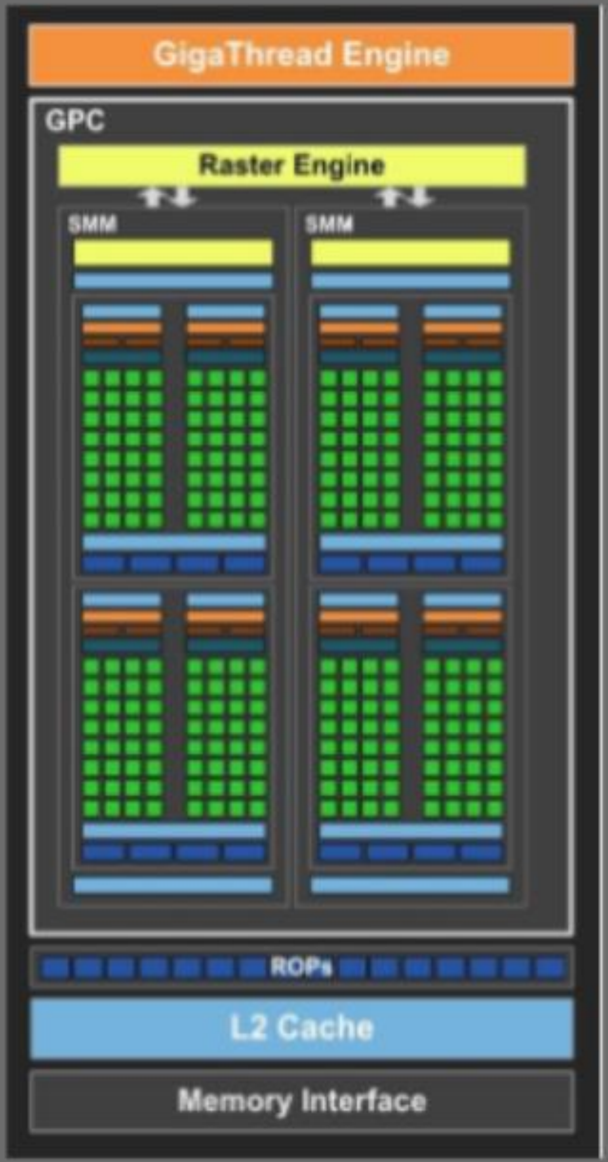
can express the complex relationships between variables by modeling the joint probability distribution $p(x, y)$

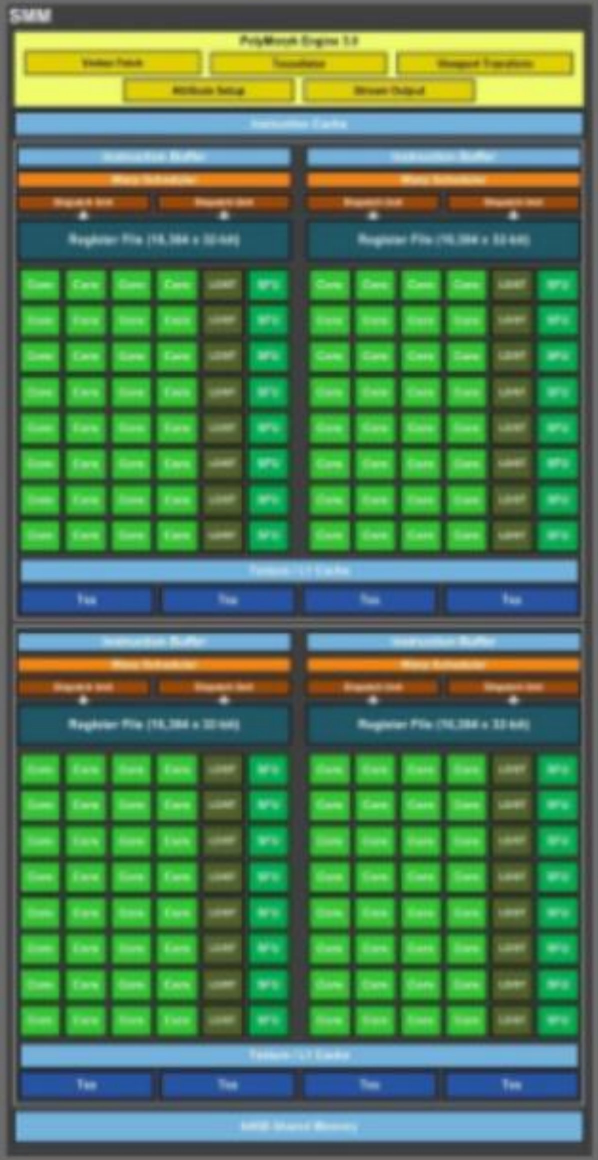
Probabilistic Graphical Models

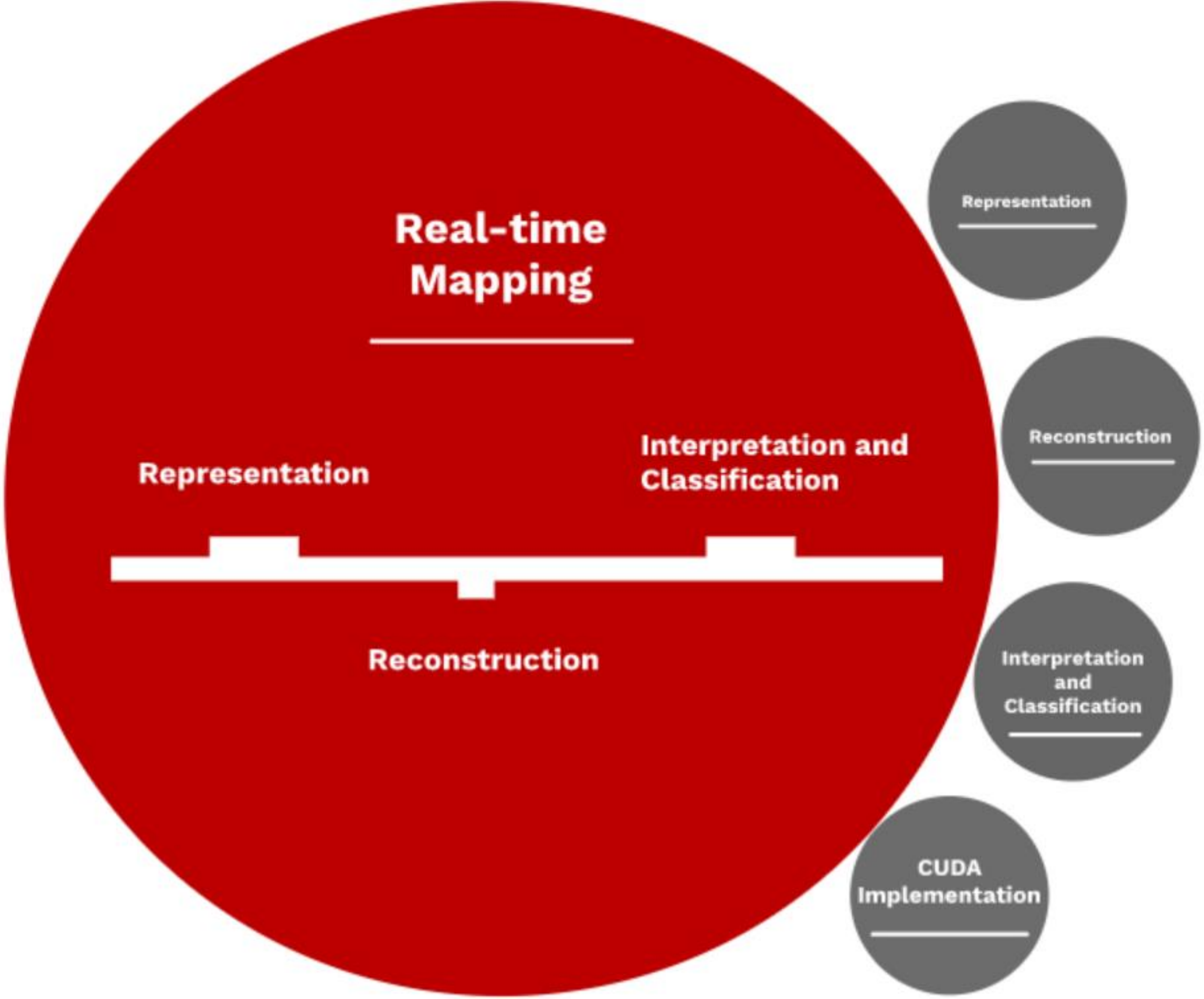
express the relationship and conditional dependence between variables in a graphical structure.



Why the NVIDIA Tegra X1 SoC?







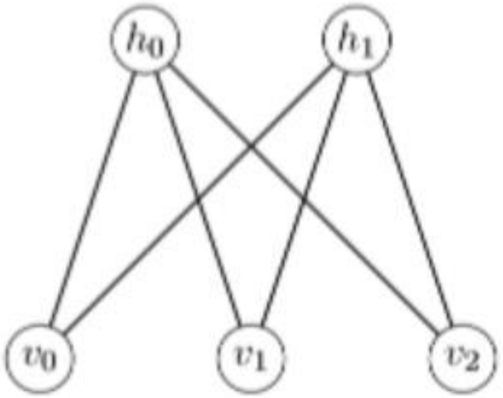
Real-time Mapping
Representation
Interpretation and Classification
Reconstruction
Representation
Reconstruction
Interpretation and Classification
CUDA Implementation

Representation

$$P(v,h) = \frac{e^{-E(v,h)}}{Z}$$

$$v \in \mathbb{R}^{\mathbb{V}}$$

$$h \in \{0,1\}^{H}$$



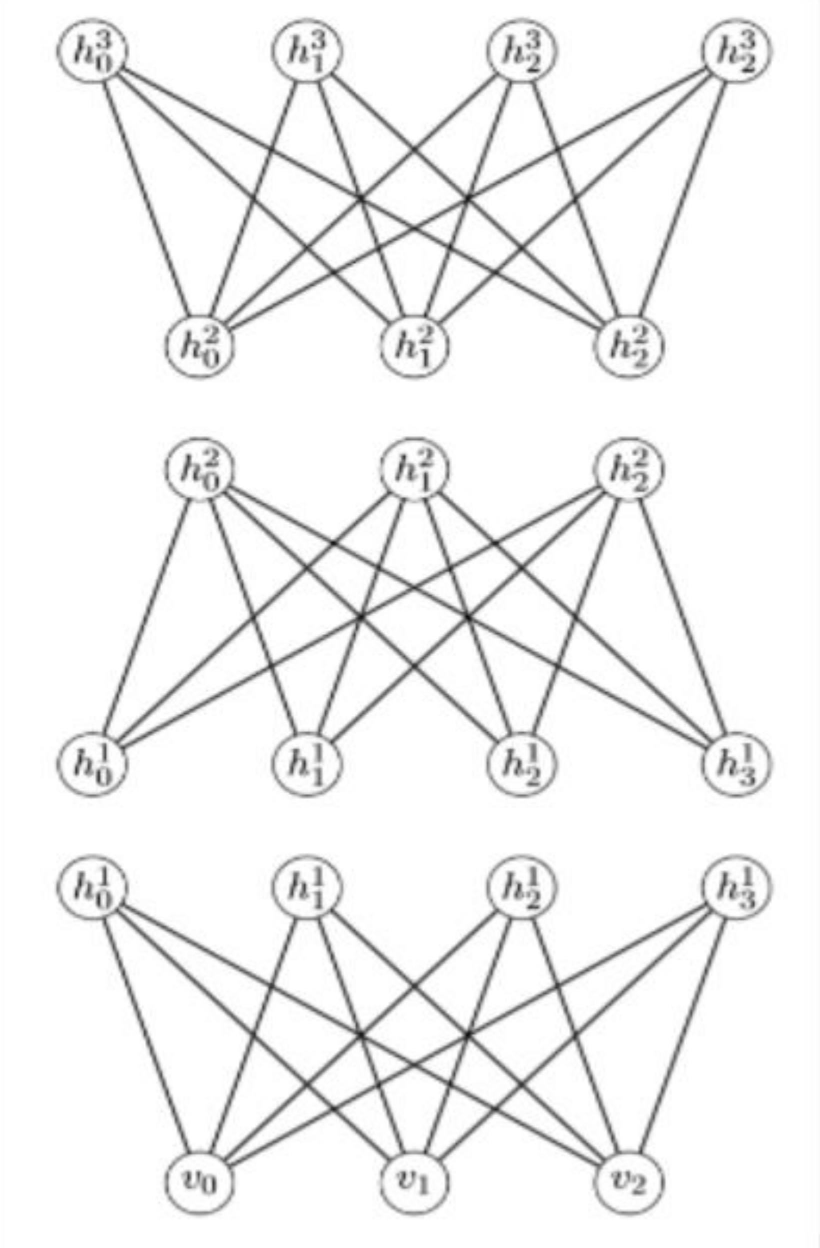
A graphical representation of a Restricted Boltzmann Machine

$$E(v,h) = -\sum_{i=1}^{V}\sum_{j=1}^{H} W_{ij}\frac{v_i}{\sigma}h_j - \sum_{i=1}^{V}\frac{(v_i - b_i)^2}{2\sigma^2} - \sum_{j=1}^{H} c_j h_j$$

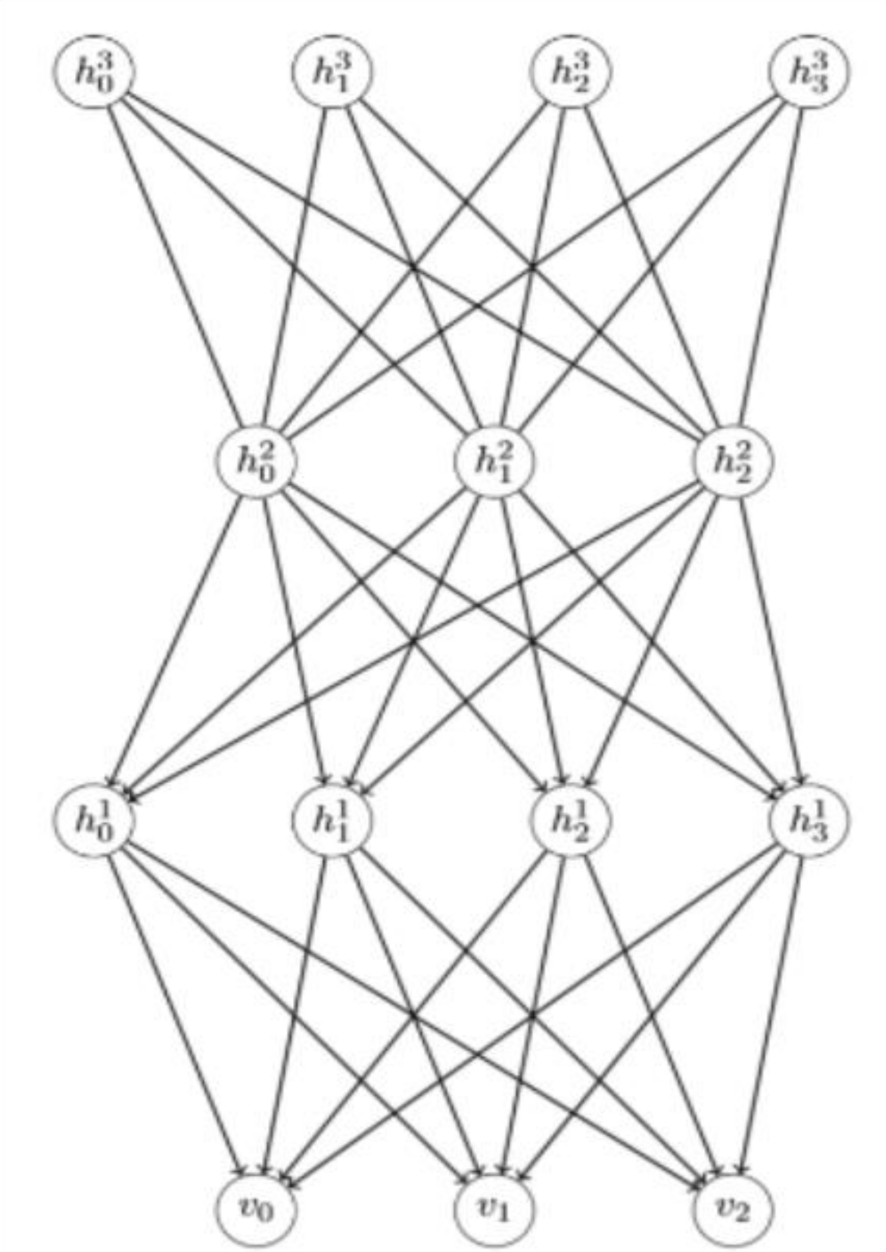
Blocked Gibbs Sampling

$$P(h|v;\theta) = \prod_j p(h_j|v), p(h_j = 1|v) = g(\sum_i W_{ij}v_i + b_i)$$

$$P(v|h;\theta) = \prod_i p(v_i|h), p(v_i = 1|h) = g(\sum_j W_{ji}h_j + c_j)$$



Three single-layered RBMs



Three-layer Deep Belief Network

Training

1. Fit the input data to the first RBM layer, i.e. find the parameters θ^1.

2. Use θ^1 as initial θ^2 to ensure that the 2-layer network is at least as good as the single RBM (θ^2 = transpose(θ^1)). Sample h^1 from the approximate posterior distribution $Q(h^1|v)$, which is the true distribution initially $P(h^1|v;\theta^1)$, and use as training data for θ^2.

3. Use θ^2 as initial θ^3 as θ^3 = transpose(θ^2)). Sample h^2 from the distribution $Q(h^2|h^1)$ = $P(h^2|h^1;\theta^2)$ and use as training data for θ^3.

4. Repeat recursively for the remaining layers till all θ^i are learned.

Training

After training L layers, the model's joint distribution **P** and its approximate posterior **Q**

$$P(\mathbf{v}, \mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(L)}) = P(\mathbf{v}|\mathbf{h}^{(1)}) \ldots P(\mathbf{h}^{(L-2)}|\mathbf{h}^{(L-1)}) P(\mathbf{h}^{(L-1)}, \mathbf{h}^{(L)}),$$

$$Q(\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(L)}|\mathbf{v}) = Q(\mathbf{h}^{(1)}|\mathbf{v}) Q(\mathbf{h}^{(2)}|\mathbf{h}^{(1)}) \ldots Q(\mathbf{h}^{(L)}|\mathbf{h}^{(L-1)}).$$

Approximate Inference

To infer the values of the top-hidden variables, a single bottom-up pass is used where sampling is done from a fully-factorized approximate posterior distribution:

$$\tilde{Q}(h^1, h^2, ..., h^L|v) = \prod_{i=1}^{L} \tilde{Q}(h^i|v)$$

instead of the non-factorized multimodal form :

$$Q(h^1, h^2, ..., h^L|v) = Q(h^1|v)Q(h^2|h^1)...Q(h^L|h^{L-1})$$

Reconstruction

Minimizing the KL-Divergence

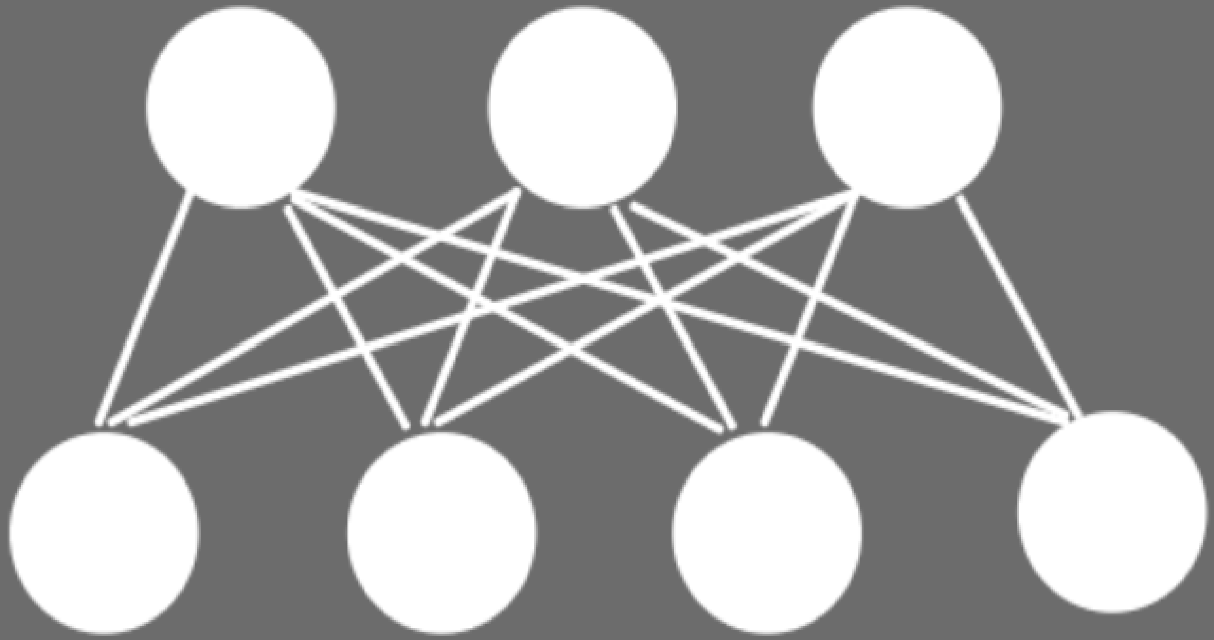
$$d_{KL}(D||M) = \sum_i D(i) log \frac{D(i)}{M(i)}$$

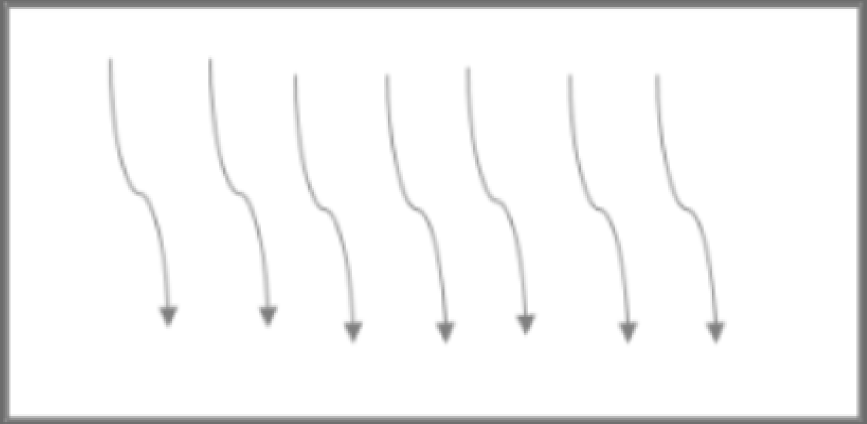
Interpretation and Classification

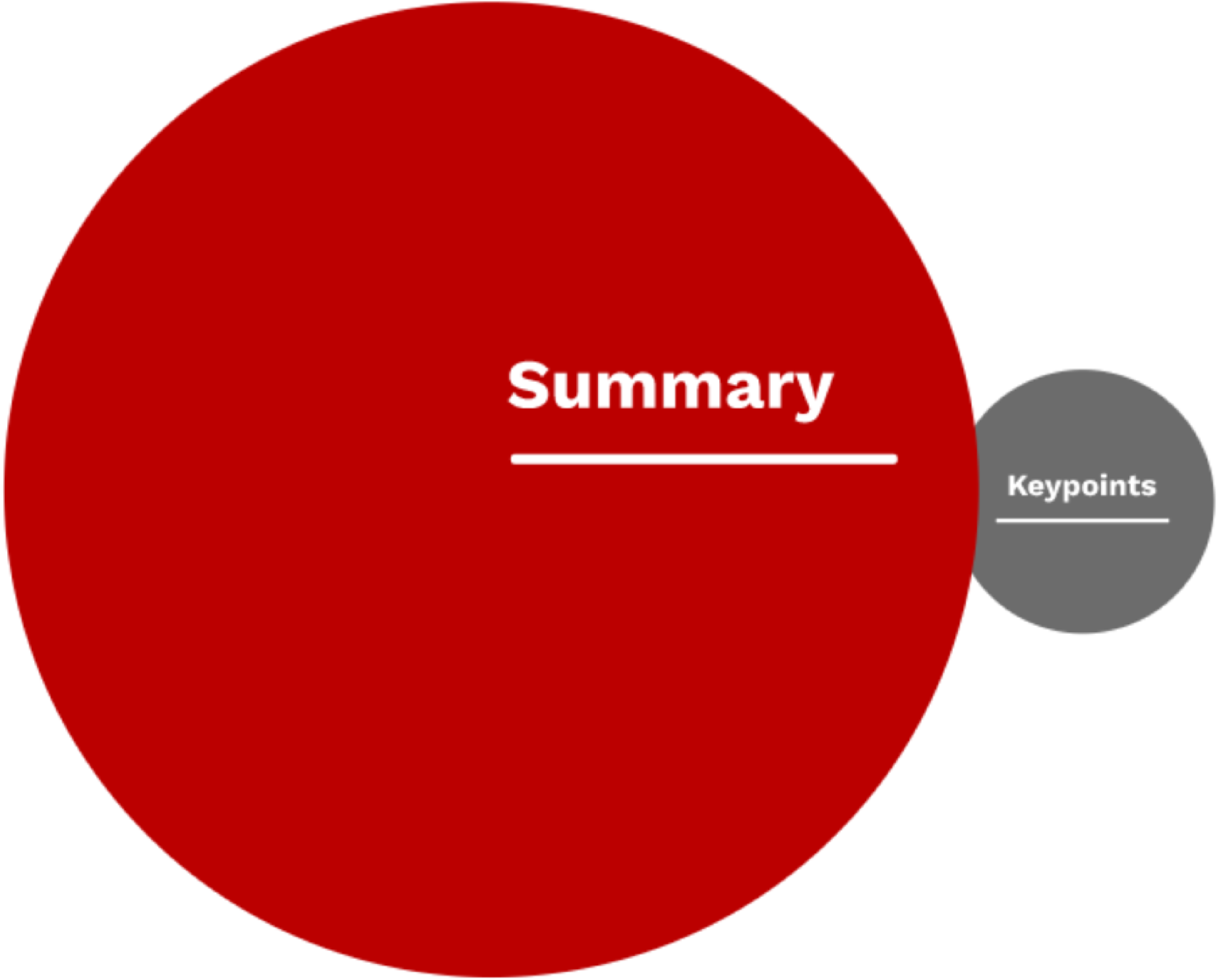
$$P(pixel, R, G, B, D; w) = \alpha . P(pixel|R; w) . P(pixel|G; w) . P(pixel|B; w) . P(pixel|D; w)$$

$$P(pixel = class|R, G, B, D; w)$$

CUDA Implementation






Summary
Keypoints

Keypoints

Scene Understanding in urban traffic environments is important and challenging

Real-time Implementation is important in automated driving and ADAS applications

Dynamic Traffic Environment is a challenge

Menna El-Shaer **el-shaer.1@osu.edu**