Uncertainty-Aware Mask2Former for robust perception in AV’s

Sukeerth Atmakuri
University of California, Riverside
satma005@ucr.edu

Dr. Bir Bhanu
University of California, Riverside
birbhanu@ucr.edu

Abstract

Reliable camera-only perception in autonomous vehicles is severely challenged not only by extreme weather but also by moderate weather transitions and varying light exposure. Phenomena such as sun glare, headlight blooming, wet reflective roads, and sudden rain can cause deterministic segmentation models to fail silently. In this work, we propose an uncertainty-aware panoptic segmentation framework designed to remain robust under these conditions. In this first phase, we have successfully established the data infrastructure by converting the BDD100K dataset—which lacks ready-to-use panoptic annotations in our environment—into a format compatible with Transformer-based panoptic architectures. We detail our custom “Semantic-to-Panoptic” conversion pipeline and present the architectural design of our future model: a Mask2Former baseline extended with Evidential Deep Learning (EDL) heads to quantify semantic and spatial uncertainty.

1 Introduction

Camera sensors are the primary modality for semantic understanding in autonomous driving due to their high spatial resolution and rich contextual information. Modern models like Mask2Former [1] achieve strong performance on panoptic segmentation benchmarks under normal conditions. However, these architectures are deterministic: they output a single class label and mask for each region, with no explicit indication of uncertainty. As a result, they may still produce high-confidence but unsafe predictions when the image is degraded by glare, rain, fog, or low light.

From a safety perspective, a perception system should not only answer “What is in the scene?” but also “How sure am I about each prediction?”. This motivates **uncertainty-aware panoptic perception**, where the model produces both panoptic segmentation and uncertainty measures.

This project aims to build an **Uncertainty-Aware Panoptic Segmentation** model for camera-only autonomous driving. We adopt a “quality over quantity” philosophy: we focus first on making the camera modality robust and self-aware before adding sensor fusion [3]. This report summarizes the work completed in the first phase:

- transforming BDD100K into a panoptic-style dataset suitable for query-based architectures;
- designing a Mask2Former baseline with a Swin Transformer backbone; and
- designing an uncertainty-aware extension using evidential and variance-based heads.

The second phase will focus on training, uncertainty calibration, and evaluation.

2 Methodology

Our methodology shifts from deterministic segmentation to probabilistic perception. At a high level, the project is split into two phases:

Preprint.

- **Baseline Phase.** Train a state-of-the-art Transformer (Mask2Former) as a strong panoptic baseline on diverse driving conditions. This phase focuses on (i) data preparation and (ii) setting up the baseline training and evaluation pipeline.
- **Uncertainty Phase.** Extend the baseline architecture with evidential and variance heads to explicitly model uncertainty. We use Evidential Deep Learning (EDL) [8, 9, 7] for semantic uncertainty and a variance branch for spatial uncertainty, enabling the model to flag unreliable predictions under adverse conditions.

Evidential Deep Learning models the network's output as parameters of a Dirichlet distribution instead of a single softmax vector, allowing the model to express both its belief and its ignorance. The additional variance head captures spatial ambiguity in the predicted masks (for example, at object boundaries in rain or fog).

3 Architecture Approach

In this section we describe the two main architectures: the deterministic baseline and the proposed uncertainty-aware extension. Both are designed around a Mask2Former-style structure with a Swin Transformer backbone.

3.1 Phase 1: Baseline Architecture (Mask2Former)

We adopt **Mask2Former** [1] as the core baseline architecture. The main components are:

1. **Backbone (Swin Transformer).** The input image is divided into patches and processed by a Swin Transformer [2], which uses shifted window attention to efficiently capture local and global context. This produces a set of multi-scale feature maps.
2. **Pixel Decoder.** A pixel decoder fuses multi-scale features into a unified dense representation at a chosen resolution. This representation acts as a memory that the queries attend to.
3. **Learnable Mask Queries.** We maintain N learnable mask queries, where each query is intended to represent a potential segment in the scene (a car, a pedestrian, the road region, etc.).
4. **Transformer Decoder.** A stacked Transformer decoder updates the mask queries through self-attention (queries interact with each other) and cross-attention (queries attend to pixel features). After several layers, each query specializes to a semantic region.
5. **Prediction Heads.**
   - **Class Head:** Maps each query embedding to class logits, followed by a softmax to obtain class probabilities for each query.
   - **Mask Head:** Maps each query to a mask embedding which is combined with pixel features to produce dense mask logits. Thresholding these logits yields binary masks per query.
6. **Panoptic Assembly.** The per-query class predictions and masks are combined to produce a panoptic segmentation map. Each pixel is assigned a semantic class and an instance ID. Overlapping masks are resolved according to scores and a predefined priority between "thing" and "stuff" classes.

Figure 1 summarizes this baseline pipeline in a flowchart-style diagram.

This baseline does not model uncertainty: the outputs are single class labels and masks, even in regions where the model is unsure (for example, glare-covered cars or foggy pedestrians).

3.2 Phase 2: Proposed Uncertainty-Aware Architecture

To handle glare, blur, and other challenging visual conditions, we extend the baseline with two new branches: an **evidential class head** for semantic uncertainty and a **spatial variance head** for boundary uncertainty. The extended architecture is illustrated in Figure 2.

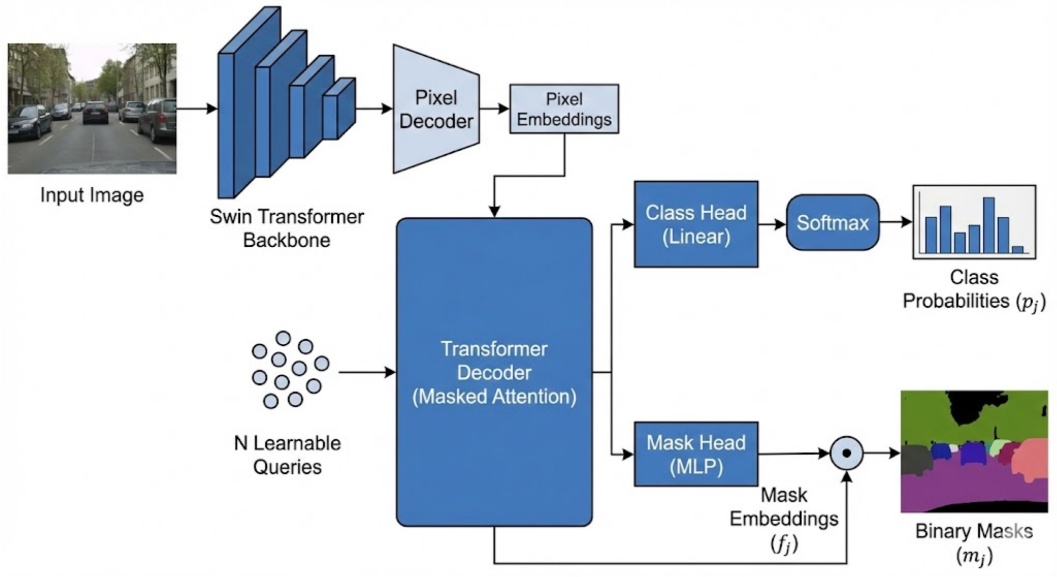

Figure 1: **Baseline Architecture.** The Swin Transformer backbone extracts multi-scale features. A pixel decoder fuses these into a dense feature map, and a Transformer decoder updates a set of learnable mask queries. Each query outputs a class label and a mask, which are combined into a deterministic panoptic segmentation map.

Evidential Class Head (Semantic Uncertainty). Instead of producing class logits that are passed through a softmax, the evidential head outputs non-negative evidence $e_k \geq 0$ for each class k. This evidence is converted into Dirichlet parameters:

$$\alpha_k = e_k + 1. \tag{1}$$

The resulting Dirichlet distribution Dir($\boldsymbol{\alpha}$) models a distribution over class probabilities. The predictive mean is:

$$\mathbb{E}[p_k] = \frac{\alpha_k}{\sum_j \alpha_j}. \tag{2}$$

The total evidence $S = \sum_j \alpha_j$ controls the level of semantic uncertainty. A simple uncertainty score is:

$$u_{\text{semantic}} = \frac{K}{S}, \tag{3}$$

where K is the number of classes. When the model sees foggy or glared regions, it can output low evidence (high u_{semantic}), reflecting its uncertainty.

Spatial Uncertainty Head (Boundary Doubt). Semantic uncertainty tells us how unsure the model is about *which class* is present. We also want to know where the *shape* or boundary of the object is unclear. For this, we introduce a spatial variance head:

- Each query produces a *variance embedding* that is projected into a dense log-variance map $\log \sigma^2(x, y)$.
- We treat the mask logit $z(x, y)$ at each pixel as a Gaussian random variable:

$$z(x, y) \sim \mathcal{N}(\mu(x, y), \sigma^2(x, y)). \tag{4}$$

- Regions with large $\sigma^2(x, y)$ correspond to uncertain boundaries: for example, along object edges in rain or fog, or in highly reflective road patches.

Visualization of the Uncertainty Outputs. During inference, each query yields:

- A Dirichlet distribution over classes (semantic prediction + uncertainty).

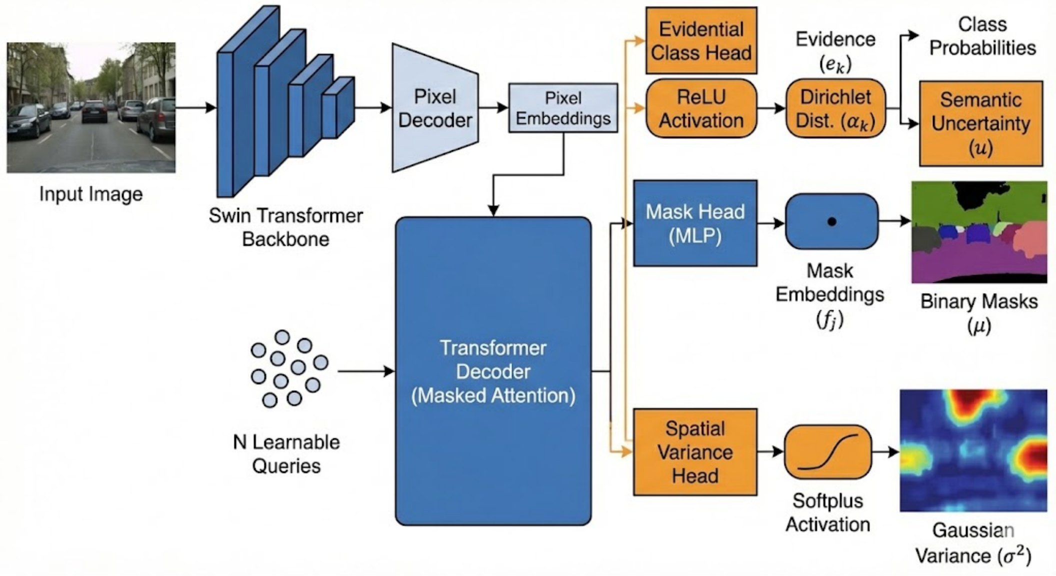

Figure 2: **Proposed Uncertainty-Aware Architecture.** Starting from the Mask2Former baseline, we replace the softmax class head with an evidential class head that outputs non-negative evidence for each class, parameterizing a Dirichlet distribution. A second branch predicts a variance embedding, which is projected into a spatial variance map over mask logits. These heads produce semantic and spatial uncertainty maps in addition to panoptic segmentation.

- A predicted mask along with a spatial variance map (boundary uncertainty).

We can visualize:

- the **panoptic segmentation map** (colored instances),
- a **semantic uncertainty heatmap** from u_{semantic} (highlighting low-evidence regions),
- a **spatial uncertainty heatmap** from $\sigma^2(x, y)$ (highlighting ambiguous boundaries).

Together, these outputs help identify where the model is likely to make mistakes, especially in adverse weather and lighting.

4 Dataset Selection

We selected the **BDD100K (Berkeley Deep Drive)** dataset as the primary benchmark for this project. Unlike datasets such as Cityscapes, which mostly contain clear daylight scenes, BDD100K covers a wide spectrum of real-world conditions:

- **Diverse lighting:** day, night, dawn, dusk.
- **Diverse weather:** clear, rainy, snowy, foggy.

This diversity is crucial for evaluating uncertainty-aware perception. For example, we can test whether the model's semantic and spatial uncertainty increase in foggy or rainy scenes compared to clear weather.

In our local setup, BDD100K provides semantic segmentation labels (per-pixel class IDs) and corresponding colorized label maps, but not COCO-style panoptic annotations. Mask2Former-style architectures require panoptic labels (class + instance IDs) or a COCO-style panoptic JSON interface for training. Therefore, a major part of this phase focused on bridging that gap by generating pseudo-panoptic labels from the available semantic masks.

5 Work Done: Data Preparation Pipeline

The key technical challenge in this phase was that BDD100K, in our environment, does not provide ready-made panoptic segmentation labels. To enable panoptic training, we implemented a custom **Semantic-to-Panoptic** conversion pipeline and organized the dataset into a panoptic-compatible structure.

Step 1: Instance Extraction from Semantic Masks

We first identify “thing” classes (e.g., car, bus, truck, person, rider, bicycle, motorcycle) and treat the rest as “stuff” classes (road, sidewalk, building, vegetation, sky, etc.). For each semantic label mask:

1. We compute a binary mask for each thing class.
2. We apply **connected component analysis** to split each thing-class mask into multiple components, each corresponding to one object instance.
3. We assign a unique instance ID to each connected component.

This yields an **instance map** where each pixel has an instance ID (for thing classes) or 0 (for stuff).

Step 2: Panoptic ID Encoding

To obtain a single panoptic label per pixel, we encode the semantic class and instance ID into one integer:

$$\text{panoptic\_id}(x, y) = \text{class\_id}(x, y) \times 1000 + \text{instance\_id}(x, y). \tag{5}$$

Stuff pixels have instance_id $= 0$, so all stuff pixels of the same class share an ID of the form class_id $\times 1000$. Different instances of the same class have different instance IDs and therefore different panoptic_id values. We store these panoptic maps as 32-bit PNG images.

Step 3: Folder Structure and Scale of Conversion

To support training and evaluation, we organized the data into the following structure:

- `images/train`, `images/val`: original RGB images.
- `labels/train`, `labels/val`: semantic ID masks.
- `panoptic_instance/train`, `panoptic_instance/val`: instance ID maps obtained by connected components.
- `panoptic_final/train`, `panoptic_final/val`: final panoptic ID maps (class*1000 + instance).

In this phase, we have successfully converted **approximately 7,000 training semantic masks** into panoptic-style labels. This provides a substantial subset of BDD100K ready for baseline Mask2Former training once GPU resources (e.g., Google Colab or a university workstation) are available. The conversion scripts generalize to the remaining training and validation images.

Figure 3 shows a qualitative example of the data preparation pipeline.

Summary of Completed Work

To summarize, the main accomplishments in this phase are:

- Implemented a semantic-to-panoptic conversion pipeline based on connected components.
- Generated panoptic-style labels for approximately 7,000 training images.
- Designed the baseline Mask2Former architecture with a Swin backbone and panoptic heads.
- Designed the uncertainty-aware extension with evidential and spatial variance heads, including the mathematical formulation for Dirichlet and variance-based uncertainty.
- Prepared the code and dataset structure so that baseline training can begin immediately once GPU access is available.

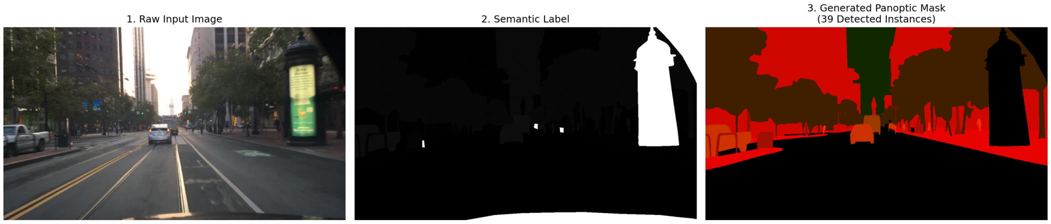

Figure 3: **Work Done Result.** Left: raw BDD100K image. Center: original semantic label (class-only). Right: our generated panoptic map, where each instance (e.g., different cars) is assigned a distinct ID and color. This panoptic representation enables training of Mask2Former-style models on BDD100K in our setup.

6 Work to be Done Next

In the second half of the project, we plan to perform the following steps:

1. **Baseline Training.** Train the standard Mask2Former baseline on the converted BDD100K panoptic labels. We will measure panoptic quality (PQ), segmentation quality (SQ), and recognition quality (RQ) as core metrics.
2. **Implementation of Uncertainty Heads.** Modify the Mask2Former decoder to integrate the evidential class head and the spatial variance head described in Figure 2. The implementation will follow the EDL formulations in [8, 9, 7].
3. **Loss Function Integration.** Replace or augment the standard cross-entropy loss with evidential loss terms that penalize overconfident errors. For the variance head, we will adopt a likelihood-based loss that encourages higher variance where prediction errors are large (uncertain regions).
4. **Uncertainty Evaluation.** Evaluate both standard panoptic metrics (PQ, SQ, RQ) and uncertainty-related metrics such as calibration error and reliability diagrams. We also plan to explore uncertainty-aware panoptic metrics (e.g., uPQ) and analyze how uncertainty behaves under different weather and lighting conditions.
5. **Qualitative Visualization.** Visualize panoptic outputs along with semantic and spatial uncertainty heatmaps for challenging scenes (night, rain, fog, glare). These visualizations will provide insight into whether the model's uncertainty aligns with human intuition about risky regions.

This completes the first phase of the project, which focuses on data preparation and architecture design. The next phase will turn these designs into trained models and detailed experimental results.

References

[1] B. Cheng et al. Masked-attention Mask Transformer for Universal Image Segmentation. In *NeurIPS*, 2021.

[2] Y. Li et al. Transformer-Based Visual Segmentation: A Survey. *IEEE TPAMI*, 2024.

[3] L. Wang et al. Benchmarking the Robustness of Panoptic Segmentation for Automated Driving. *arXiv:2402.15469*, 2024.

[4] K. Sirohi et al. Uncertainty-aware Panoptic Segmentation (EvPSNet). *IEEE Robotics and Automation Letters*, 2023.

[5] C. Deery et al. ProPanDL: A Modular Architecture for Uncertainty-Aware Panoptic Segmentation. *arXiv:2304.08645*, 2023.

[6] A. Cao et al. PaSCo: Urban 3D Panoptic Scene Completion with Uncertainty Awareness. In *CVPR*, 2024.

[7] S. Ancha et al. Deep Evidential Uncertainty Estimation for Semantic Segmentation. In *ICRA*, 2024.

[8] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential Deep Learning to Quantify Classification Uncertainty. In *NeurIPS*, 2018.

[9] A. Amini et al. Deep Evidential Regression. In *NeurIPS*, 2020.

[10] M. Mossina et al. Conformal Prediction for Image Segmentation. In *MICCAI*, 2025.

[11] EvPSNet GitHub Repository. `https://github.com/kshitij3112/EvPSNet`.

[12] PaSCo GitHub Repository. `https://github.com/astra-vision/PaSCo`.

[13] Evidential Deep Learning Repo. `https://github.com/aamini/evidential-deep-learning`.

[14] Y. Li et al. TEDL: A Two-stage Evidential Deep Learning Method. *arXiv:2209.05522*, 2022.

[15] T. Nagahama et al. Learning and Predicting the Unknown Class Using Evidential Deep Learning. *Scientific Reports*, 2023.

[16] Evidential DL for UQ and Calibration. `https://inspirehep.net/literature/2867076`.