U-Net and CBAM-Enhanced U-Net for Explainable Urban Scene Analysis on Cityscapes

Anh Tuan Tran and Jared Young
Department of Computer Science, Kent State University, Kent, Ohio, USA
CS64201 Advanced Artificial Intelligence

***Abstract*—Semantic segmentation is a central perception task for autonomous urban driving because safe street-scene understanding depends on assigning reliable semantic labels to drivable surface, sidewalks, vehicles, pedestrians, and traffic infrastructure. This paper studies a U-Net baseline and a CBAM-enhanced U-Net variant for Cityscapes-based urban scene segmentation, then uses their predicted masks as the input to an explainable downstream scene-analysis module. The segmentation component follows an encoder-decoder design with skip connections, while the attention-enhanced variant applies channel and spatial recalibration to selected skip features before decoder fusion. After inference, the predicted mask is restored to the original image resolution, rendered as a user-selectable overlay, and passed to a deterministic analysis layer that derives class coverage, planning-oriented semantic groups, connected-component region counts, layout priors, spatial relations, scene tags, and natural-language summary text. The scene-analysis stage is not a second learned network; it is a reproducible interpretation procedure built directly on the segmentation mask. Rather than claiming state-of-the-art benchmark performance, the paper presents a model-centered and application-grounded study of how a classical encoder-decoder architecture, a lightweight attention refinement, an interactive inference workflow, and an explainable reasoning layer can be combined for urban scene understanding.**

***Index Terms*—semantic segmentation, U-Net, CBAM, Cityscapes, urban scene analysis, explainable vision**

I. Introduction

One natural motivation for this work is autonomous driving. A self-driving vehicle operating in an urban environment must distinguish drivable surface from sidewalk, separate vehicles from pedestrians and cyclists, recognize traffic infrastructure, and interpret the broader structure of the street scene under changing lighting, clutter, and occlusion. These requirements make semantic segmentation especially attractive because it converts a camera image into a dense semantic map of the environment, and urban benchmarks such as Cityscapes were developed precisely to support this kind of street-scene understanding [7], [8].

Within that perception setting, the central modeling question is how to produce masks that preserve both broad scene structure and fine semantic boundaries. Encoder-decoder networks such as U-Net remain attractive because they combine deep contextual features with high-resolution skip connections [1]. Lightweight attention modules such as CBAM are also appealing in urban scenes because they selectively emphasize informative channels and spatial locations without requiring a completely different backbone [2]. For a project that needs a technically defensible baseline and an interpretable architectural extension, U-Net and U-Net plus CBAM form a natural pair of models to study.

At the same time, a useful perception model should support more than a colored mask alone. In a driving context, it is valuable to know not only that road, sidewalk, vehicles, pedestrians, and signs are present, but also how strongly they organize the observed scene. Prior work on scene parsing, urban-scene priors, and fuller scene representation suggests that dense predictions are most useful when they support broader interpretation rather than acting only as endpoint visualizations [9]–[11], [13].

This paper therefore treats the problem at two connected levels. The first level studies U-Net as the primary segmentation baseline and investigates CBAM-enhanced U-Net as a lightweight attention refinement. The second level uses the resulting masks as the input to a deterministic urban scene-analysis module that derives class coverage, approximate region counts, layout priors, spatial relations, and scene-level summary signals. The contribution is not a new state-of-the-art architecture or a completed benchmark campaign. Instead, it is an implementation-grounded AI study that connects a classical encoder-decoder model, an attention-augmented variant, a runnable application workflow, and an explainable downstream reasoning layer within a single urban-scene system.

The remainder of this paper is organized as follows. Section II reviews the background needed for semantic segmentation, encoder-decoder networks, and attention-guided feature refinement. Section III discusses related work in segmentation and scene parsing. Section IV presents the proposed approach, including the U-Net baseline, the CBAM-enhanced variant, and the downstream scene-analysis formulation. Section V describes the Cityscapes-based dataset setting, Section VI summarizes the implementation and training pipeline, and Section VII presents the reported experimental results. Section VIII describes the integrated application workflow, Section IX compares the present study to prior work, Section X outlines limitations and future directions, and Section XI concludes the paper.

II. Background

Semantic segmentation assigns a semantic class label to every pixel in an image, making it one of the most informative dense prediction tasks in computer vision. In urban street scenes, this representation is especially useful because the

scene is naturally composed of structured classes such as road, sidewalk, building, sky, person, rider, car, traffic sign, and vegetation. A model that segments these categories turns an image into a structured intermediate representation that can support downstream reasoning about mobility, built form, openness, and the presence of active transportation users. This makes semantic segmentation a strong backbone task for road-scene perception, robotics, and urban-scene interpretation [7], [8].

Encoder-decoder architectures are especially relevant to this setting because they combine coarse semantic context with high-resolution localization. In a U-Net-style model, the encoder progressively compresses the image into deeper feature maps while the decoder reconstructs spatial detail using skip connections from earlier layers [1]. Those skip connections matter in urban imagery because thin boundaries, sidewalk edges, pedestrian silhouettes, traffic-sign outlines, and vehicle contours can be lost if only coarse bottleneck features are used. Related segmentation work also shows that multi-scale context remains important, whether it is captured through fully convolutional dense prediction, atrous context aggregation, or more recent hierarchical backbones [3], [4], [6].

Attention mechanisms add another useful idea. Instead of changing the core encoder-decoder structure, they reweight intermediate features so that the model responds more strongly to semantically relevant channels and spatial regions. CBAM is a compact example of this strategy because it applies channel attention followed by spatial attention using pooled descriptors and a lightweight convolutional projection [2]. In an urban segmentation setting, such recalibration is appealing because visually crowded scenes often contain a mixture of large background regions and small but important foreground objects.

Once a segmentation mask is available, it can also be used for more than visualization. Class coverage, connected regions, coarse layout structure, and local semantic adjacency can all be estimated directly from the mask, making the prediction a useful intermediate representation for scene parsing and downstream interpretation [9], [10], [12]. That observation motivates the second half of this paper, where mask-derived scene analysis is treated as an explainable layer built on top of the segmentation model rather than as a separate learned network.

III. Related Work

A. Segmentation Architectures

The segmentation literature most directly relevant to this study begins with U-Net, which established a highly influential encoder-decoder design for dense prediction with symmetric skip connections between contracting and expanding paths [1]. Although originally introduced for biomedical segmentation, the underlying mechanism is general: local detail is preserved by copying higher-resolution encoder features into the decoder, which helps recover boundaries that would otherwise be blurred by repeated downsampling. That makes U-Net a strong baseline for structured urban scenes where both global layout and fine object contours matter.

CBAM provides a lightweight way to refine such a baseline without replacing the backbone itself. The module sequentially applies channel attention and spatial attention, allowing the network to suppress less relevant responses and amplify useful semantic cues [2]. In the context of this paper, CBAM is not treated as an independent segmentation family. Instead, it is studied as a targeted enhancement to U-Net, with the goal of improving the quality of the features passed through skip connections before decoder fusion.

Other segmentation models provide useful comparative context. FCN established the modern view of semantic segmentation as end-to-end dense prediction [3]. DeepLabv3 emphasized multi-scale context through atrous convolution and atrous spatial pyramid pooling [4]. MobileNetV3 with LR-ASPP represents an efficiency-oriented direction for lighter segmentation pipelines [5]. Swin Transformer V2 illustrates a more recent hierarchical transformer approach to dense visual representation [6]. These models remain relevant to the broader experimental environment, but they are secondary to the main focus of this paper, which is the U-Net baseline and its CBAM-enhanced variant.

B. Scene Parsing And Urban Context

The scene-analysis portion of this work is more closely related to scene parsing and context-aware interpretation than to plain mask visualization. PSPNet is one of the clearest papers linking semantic segmentation to whole scene understanding by showing that dense prediction quality depends on global multi-scale context [9]. SPGNet pushes that idea further by treating intermediate semantic predictions as signals that can guide later stages of scene parsing [10]. HANet is especially relevant in the urban case because it exploits street-scene structure, arguing that classes in road scenes exhibit reliable vertical layout priors such as sky above road and traffic-related objects occupying constrained regions [11].

Older contextual work by Mottaghi et al. is also important because it frames semantic segmentation as one component in a larger scene-understanding pipeline rather than an isolated task [12]. Panoptic Segmentation provides a complementary perspective by formalizing a richer scene-level output that unifies stuff and thing reasoning [13]. The present study does not implement PSPNet, SPGNet, HANet, CRF-based holistic inference, or panoptic segmentation. However, these works collectively justify the paper's central framing: once a semantic mask exists, it is reasonable to use it as an intermediate representation for broader scene interpretation.

IV. Approach

A. Segmentation Model Approach

Although the ground-truth segmentation target is stored as a single dense label map $Y \in \{0, \ldots, C-1\}^{H \times W}$, the learning problem can also be viewed class-wise by decomposing that label map into one binary mask per class. For class c, the class-specific target channel is $Y^{(c)}(x, y) = \mathbf{1}[Y(x, y) = c]$,

and stacking these channels yields an equivalent one-hot target tensor in $\{0,1\}^{C\times H\times W}$. This representation is useful because it aligns the supervision signal with the model's per-class output channels, supports class-wise overlap computation during optimization, and allows classes to be separated or recombined as part of later reasoning without changing the underlying semantic labeling problem.

1) Original U-Net Architecture

The segmentation stage treats an urban image as an input tensor $I \in \mathbb{R}^{3\times H\times W}$ and learns a mapping $f_\theta(I) = Z$, where $Z \in \mathbb{R}^{C\times H\times W}$ contains per-class logits for the $C = 20$ labels used in the project. The predicted semantic mask is obtained by

$$\hat{M}(x,y) = \arg\max_{c\in\{0,\dots,C-1\}} Z_c(x,y). \tag{1}$$

This formulation is standard for multi-class semantic segmentation, but the architectural choice of f_θ strongly affects how well local detail and global context are preserved [3], [8].

The primary baseline in this work is U-Net. The implemented model follows the classical encoder-decoder pattern with repeated double-convolution blocks, max pooling in the encoder, transposed-convolution upsampling in the decoder, and skip connections that concatenate higher-resolution encoder features with the corresponding decoder activations [1]. In compact form, one decoder stage may be written as

$$D_\ell = \phi_\ell\left([\mathrm{Up}(D_{\ell+1}), E_\ell]\right), \tag{2}$$

where E_ℓ is the encoder feature map at level ℓ, $D_{\ell+1}$ is the deeper decoder feature map, $\mathrm{Up}(\cdot)$ denotes learned upsampling, $[\cdot,\cdot]$ denotes channel concatenation, and ϕ_ℓ denotes the convolutional fusion block. This structure is well suited to urban scenes because road layout, building mass, and sky occupancy depend on broad spatial context, while sidewalks, poles, pedestrians, and traffic-sign boundaries still require high-resolution localization.

2) CBAM Mechanism

CBAM provides a lightweight attention mechanism that can be applied to an intermediate feature map F without replacing the encoder-decoder backbone.

The channel-attention gate is

$$A_c(F) = \sigma\left(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))\right), \tag{3}$$

which produces the channel-refined tensor

$$F_c = A_c(F) \odot F. \tag{4}$$

Spatial attention is then computed as

$$A_s(F_c) = \sigma\left(f^{7\times7}\left([\mathrm{Avg}_c(F_c), \mathrm{Max}_c(F_c)]\right)\right), \tag{5}$$

and the final refined skip tensor becomes

$$\tilde{F} = A_s(F_c) \odot F_c, \tag{6}$$

where σ is the sigmoid function and $\odot$ denotes element-wise multiplication [2]. In general form, this sequential channel and spatial recalibration allows a network to suppress weaker responses and emphasize semantically informative features before they are passed to later layers.

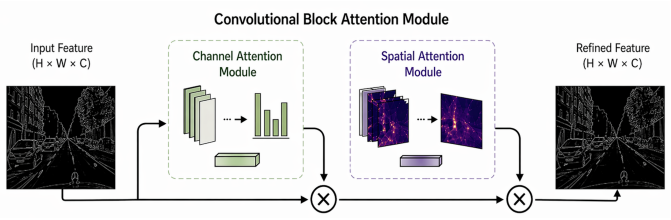

Fig. 1. CBAM Attention Mechanism

3) Proposed U-Net + CBAM Model

The previous subsubsections define the segmentation formulation, the U-Net baseline, and the CBAM mechanism; this subsection explains how they are combined in the proposed model used in this paper. Our proposed segmentation model starts from the U-Net baseline and enhances it with CBAM-based skip-feature refinement. Rather than replacing the encoder, decoder, or final projection head, the model preserves the standard U-Net topology and inserts CBAM modules on three higher-resolution skip tensors before they are fused back into the decoder.

Figure 2 illustrates our proposed U-Net + CBAM model. This design choice makes the architectural contribution specific and interpretable. The U-Net backbone continues to provide the multi-scale encoder-decoder workflow, while CBAM selectively recalibrates the skip features that carry fine spatial detail from the encoder to the decoder. In the implemented model, the refined skip tensors replace their unmodified counterparts during decoder concatenation, and a final 1×1 convolution still projects the last decoder tensor to the class-logit output space. The goal is not to replace U-Net with a heavier architecture, but to improve the quality of the semantic information passed through the skip pathway using a lightweight attention mechanism.

B. Downstream Scene-Analysis Approach

Once the full semantic mask has been produced, the system executes a deterministic scene-analysis pass. The predicted mask is first restored to the original image resolution so that all later measurements remain aligned with the source geometry. Let the restored semantic mask be $M \in \{0,\dots,C-1\}^{H\times W}$, where $N = H \cdot W$ is the total number of pixels. The first stage computes per-class coverage. For each class c,

$$P_c = \sum_{x,y} \mathbf{1}[M(x,y) = c], \qquad \mathrm{pct}_c = 100 \cdot \frac{P_c}{N}. \tag{7}$$

These statistics capture the semantic composition of the image and form the base layer for every later scene-level indicator.

The next step aggregates raw classes into planning-oriented semantic groups such as mobility surface, built environment, green and open view, vehicles, and people with active mobility. For a group G,

$$P_G = \sum_{c\in G} P_c, \qquad \mathrm{pct}_G = 100 \cdot \frac{P_G}{N}. \tag{8}$$

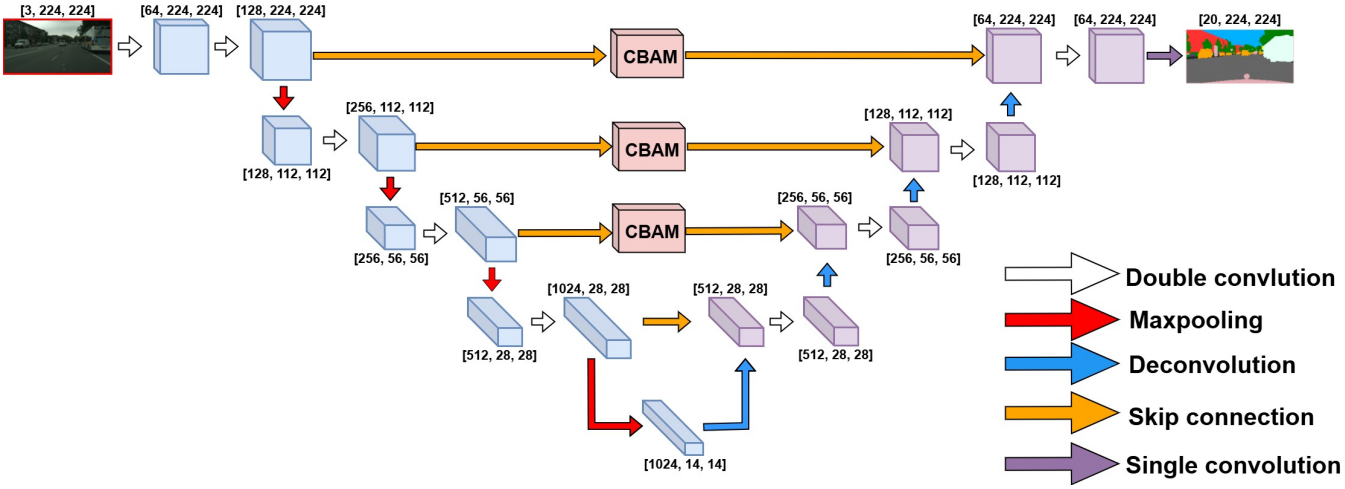

Fig. 2. U-Net + CBAM Architecture

This grouped representation does not replace the underlying segmentation mask. Instead, it creates a second interpretive layer that is easier to read at the scene level and better aligned with the broader scene-parsing perspective of PSPNet and SPGNet [9], [10]. It is also compatible with the class semantics emphasized by Cityscapes [7].

The analysis then extracts connected components for countable or region-like classes such as people, riders, vehicles, traffic lights, and traffic signs. For a class c, define the binary mask

$$B_c(x, y) = \mathbf{1}[M(x, y) = c]. \tag{9}$$

If K_i denotes a connected component in B_c and $A_{\min}$ is the minimum accepted area, then the approximate region count is

$$\text{count}_c = \#\{K_i \subseteq B_c : |K_i| \geq A_{\min}\}. \tag{10}$$

This is not equivalent to full instance or panoptic segmentation [13], but it provides a transparent approximation to the number of visible object-like regions inferred from the semantic mask.

For each surviving region, the system records a centroid, bounding box, coarse band location, and local semantic context. The local context is computed from a dilated shell around the region:

$$S_i = \text{dilate}(K_i, r) \setminus K_i, \tag{11}$$

where r is a small radius in pixels. The semantic labels in S_i are used to determine whether the region lies next to road, sidewalk, greenery, or built surfaces. This region-context view is conceptually consistent with work that treats segmentation as one layer within a broader scene reasoning pipeline [12], [13].

The next stage computes layout priors by dividing the image into coarse vertical and horizontal bands. If b denotes one such band with N_b pixels, then class coverage within that band is

$$\text{pct}_c^{(b)} = 100 \cdot \frac{1}{N_b} \sum_{(x,y)\in b} \mathbf{1}[M(x, y) = c]. \tag{12}$$

These band-wise statistics allow the system to ask whether sky is concentrated near the top, whether road dominates the lower portion of the frame, whether built form appears along the lateral edges, and whether the center behaves like an open corridor. The use of these coarse height- and position-dependent priors is especially defensible in urban imagery because similar structural regularity is emphasized by HANet [11].

Higher-level outputs are derived from these statistics through fixed heuristic formulas rather than learned parameters. If g denotes green surface coverage, sky visible sky coverage, r road coverage, m motorized-vehicle coverage, s sidewalk coverage, a active-mobility coverage, and n_a active-mobility region count, then representative scores are

$$\text{GreenSpace} = g, \tag{13}$$

$$\text{OpenView} = g + 0.5\, sky, \tag{14}$$

$$\text{RoadCarDominance} = r + 4m. \tag{15}$$

Pedestrian-support (PS) is defined more conservatively as a mixture of sidewalk allocation and visible activity:

$$\text{PS} = 0.75 \cdot \text{SB} + 0.25 \cdot \text{AP}, \tag{16}$$

where SidewalkBalance (SB) depends on the share of sidewalk within mobility surfaces and ActivePresence (AP) combines active-mobility coverage with active-mobility region counts. These scores are intentionally simple. Their purpose is not to approximate formal planning metrics, but to provide interpretable mask-derived indicators that can be inspected and explained.

Finally, the system assembles these intermediate outputs into spatial flags, relation flags, scene tags, warnings, and a summary paragraph. Scene tags are emitted only when multiple supporting signals agree, echoing the broader scene-parsing emphasis on contextual and holistic interpretation rather than isolated pixel evidence [9], [10], [12]. Because each stage depends only on the predicted mask and fixed heuristics, the full analysis remains explainable and reproducible.

V. Dataset

The experimental dataset setting for this work is grounded in Cityscapes-style urban scene understanding [7]. This is an appropriate choice for the paper's motivating domain because Cityscapes contains realistic street-view imagery with dense labels for drivable surface, sidewalk, buildings, vegetation, sky, pedestrians, riders, vehicles, and traffic infrastructure. The implemented label space uses 20 classes in total: 19 urban semantic classes plus an explicit `others` category. In the local preprocessing and training pipeline, ignored pixels are remapped into this `others` class so that the downstream system always works with a complete fixed label set.

Fig. 3. Sample In Dataset

Figure 3 shows a representative example from the dataset. It illustrates the kind of dense urban scene targeted in this work, where large background regions such as road, building, and sky appear together with smaller semantic categories such as traffic signs, poles, pedestrians, and vehicles. That mixture is useful for urban-scene analysis, but it also introduces a strong class-imbalance problem at the pixel level because visually dominant "stuff" classes occupy far more area than thinner or less frequent objects.

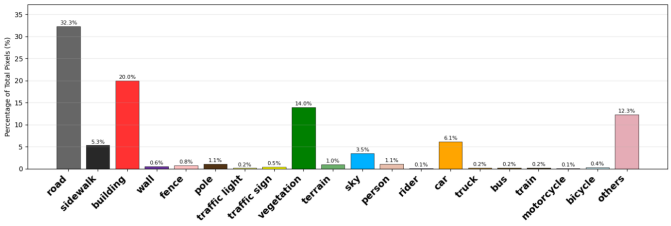

Fig. 4. Label Distribution in Training Set

As shown in Fig. 4, the training set is strongly imbalanced at the pixel level. Road is the dominant class at about 32.3%, followed by building at 20.0%, vegetation at 14.0%, and `others` at 12.3%. A smaller set of classes remains moderately represented, including car at 6.1%, sidewalk at 5.3%, and sky at 3.5%. In contrast, many semantically important classes occupy only a small fraction of the pixels, including wall (0.6%), fence (0.8%), pole (1.1%), traffic light (0.2%), traffic sign (0.5%), person (1.1%), rider (0.1%), truck (0.2%), bus (0.2%), train (0.2%), motorcycle (0.1%), and bicycle (0.4%).

This imbalance matters because the model sees far more pixels from large "stuff" classes such as road, building, vegetation, and sky than from thin, small, or infrequent object classes. As a result, traffic infrastructure, cyclists, riders, and other low-frequency categories are harder to learn consistently, even though they can be important for urban-scene understanding. The imbalance also affects how optimization and evaluation should be interpreted [8]. In particular, rare classes can have unstable or lower IoU values even when the dominant scene structure is segmented reasonably well. For that reason, the project uses Dice loss during training and reports class-wise IoU together with frequency-weighted IoU so that evaluation is not driven only by the largest classes in the label distribution.

The experiments use preprocessed NumPy arrays under a Cityscapes-derived directory structure. Images are loaded from split-specific `image` folders and masks from matching `label` folders. The local split policy is practical rather than benchmark-official: the `train` split uses all 2,975 training images, while the validation directory is divided into a 400-image validation subset and a held-out 100-image test subset. This matches the project logs, which report "Loaded train set: 2975 images," "Loaded val set: 400 images," and "Loaded test set: 100 images." This split makes the training and evaluation workflow straightforward to manage in the local environment, but it also means the paper should describe the dataset usage as a Cityscapes-derived experimental setup rather than as an official test-server submission protocol. In that sense, the dataset section supports the evaluation of U-Net and U-Net plus CBAM on urban semantics while still reflecting the practical constraints of the local data split used in this project.

VI. Implementation

A. Training And Inference Pipeline

The implementation supports both model development and interactive inference. Training is centered on `train.py`, with configuration loaded from `configs.json` and may be overridden from the command line for dataset path, output location, image size, batch size, epoch count, and debug mode. For the configuration emphasized in this paper, the model receives three-channel RGB input and all samples are standardized to 224×224 resolution before entering the network. The training pipeline uses lightweight augmentation in the form of resizing and random horizontal flipping, while validation and test images are resized without additional augmentation so that evaluation remains stable across epochs. Mini-batches are formed with a batch size of 16 and two dataloader workers. The resolved configuration is saved into the run directory so that each training run records the settings actually used.

Within the trainer, optimization uses Dice loss together with the RAdam optimizer, initialized with a learning rate of 10^{-3}. Training is allowed to continue for as many as 300 epochs, but the schedule is regulated by `ReduceLROnPlateau`, which monitors validation mean IoU, reduces the learning rate by a factor of 0.5 after five non-improving validation epochs, and does not decrease below 10^{-7}. Early stopping provides

an additional termination criterion by halting the run after 50 consecutive validation epochs without improvement. At the reporting level, the trainer computes epoch-wise training and validation summaries for loss, pixel accuracy, mean IoU, frequency-weighted IoU, Dice score, class-wise IoU, and the current learning rate. For a class c, the monitored overlap terms follow the standard dense prediction definitions

$$\mathrm{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{17}$$

and, if $n_c = TP_c + FN_c$ denotes the number of ground-truth pixels for class c and $N = \sum_c n_c$ is the total number of labeled pixels in the evaluated set, the frequency-weighted IoU is

$$\mathrm{FWIoU} = \sum_c \frac{n_c}{N} \cdot \frac{TP_c}{TP_c + FP_c + FN_c}. \tag{18}$$

This weighting gives greater influence to classes that occupy more ground-truth pixels, which is useful in the present dataset because road, building, vegetation, and other large regions dominate the pixel distribution. Unlike plain mean IoU, which treats all classes equally, frequency-weighted IoU provides a complementary view of segmentation quality that remains sensitive to overall scene structure under class imbalance. The corresponding class-wise Dice score is

$$\mathrm{Dice}_c = \frac{2TP_c + \epsilon}{2TP_c + FP_c + FN_c + \epsilon}. \tag{19}$$

The best checkpoint is selected by the highest validation mean IoU, which is also the quantity used by the scheduler. The trainer writes one CSV row per epoch so the full learning history can be visualized later, and the same metric definitions are reused on the held-out test split after training. The evaluation module then converts the logged history into metric charts and produces qualitative prediction grids for test samples.

The primary models emphasized in this paper are U-Net and U-Net plus CBAM. The same training environment also includes FCN, DeepLabV3, LightSeg, and SwinV2B [1]–[6]. The source tree also contains a `YOLOv11` placeholder path, and `train.py` still lists it in the set of allowed model choices. In the present draft, that line is best treated as a secondary comparison entry rather than as part of the paper's primary implementation story or deployed application workflow.

For inference, the unified application in `app.py` loads checkpoints, preprocesses an uploaded image, predicts a full semantic mask at the selected model resolution, and restores that mask to the original image geometry with nearest-neighbor resizing. The same full mask is then used in two ways. First, the interface renders a user-selectable overlay for qualitative inspection. Second, the mask is passed to the downstream analysis layer so that later scene interpretation remains tied directly to the segmentation output. The interface can also run secondary comparison models on the same image, but those paths are best understood as qualitative reference points rather than the core technical focus of the paper.

B. Implementation Boundaries

One of the most important implementation choices in this work is that the scene-analysis layer remains separate from the learned segmentation model. The analysis module does not backpropagate into the segmentation backbone, and it does not require any additional training data or learned supervision. This choice keeps the architecture modular and makes the later analysis interpretable, but it also means the scene-analysis quality is bounded by the quality of the input mask and the strength of the chosen heuristics.

VII. Results

Table I summarizes the reported segmentation results from the local training pipeline. Among the compared models, the proposed U-Net plus CBAM variant gives the strongest overall performance, achieving the lowest loss (0.238) together with the highest pixel accuracy (0.903), mean IoU (0.686), and frequency-weighted IoU (0.824). Relative to the plain U-Net baseline, this corresponds to a loss reduction of 0.018, a pixel-accuracy gain of 0.006, a mean-IoU gain of 0.019, and a frequency-weighted-IoU gain of 0.007, while increasing the parameter count only slightly from 49,925,268 to 49,936,314. These numbers support the central claim of the paper: lightweight attention on selected skip pathways improves the baseline without materially changing its computational scale. The broader comparison is also informative. FCN remains competitive in pixel accuracy (0.897) and frequency-weighted IoU (0.818), but it still trails the CBAM-enhanced model on the principal overlap metrics. DeepLabV3 is similarly competitive in mean IoU (0.641), yet does not match the best overall balance of overlap and loss. By contrast, SwinV2B underperforms substantially in this local setup despite having the largest parameter count, while the lightweight models `YoLoV11` and `MobileNet_V3` reduce parameter count dramatically but also finish with lower IoU and frequency-weighted IoU. These results should therefore be read as local experimental findings that support a model-centered comparison rather than as an official benchmark reproduction.

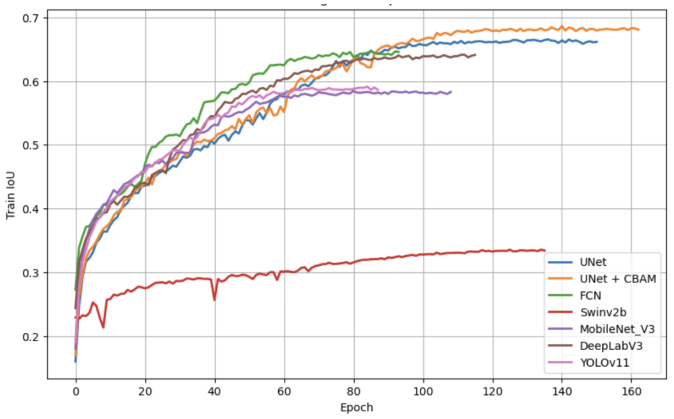

Fig. 5. Training IOU Comparison

TABLE I
SEGMENTATION MODEL PERFORMANCE COMPARISON

Model	Params	Loss ↓	Pixel Acc ↑	IoU ↑	FWIoU ↑
UNET	49,925,268	0.256	0.897	0.667	0.817
UNET_CBAM (OURS)	**49,936,314**	**0.238**	**0.903**	**0.686**	**0.824**
SWINV2B	87,159,308	0.445	0.828	0.429	0.716
FCN	32,956,500	0.278	0.897	0.646	0.818
YoLoV11	3,166,964	0.321	0.879	0.591	0.791
MobileNet_V3	3,221,368	0.331	0.878	0.584	0.791
DeepLabV3	60,995,688	0.283	0.895	0.641	0.751

Figure 5 shows that the summary statistics in Table I are consistent with the observed training dynamics. The U-Net and U-Net plus CBAM runs converge into the strongest IoU range among the compared models, with the CBAM-enhanced variant finishing above the plain baseline. This pattern suggests that the improvement is not merely an isolated endpoint fluctuation, but is reflected across the training trajectory. FCN and DeepLabV3 remain reasonably competitive during much of training, yet their curves level off below the final IoU reached by the proposed model. SwinV2B exhibits the weakest trajectory in the present experiment, reinforcing that a larger backbone does not automatically guarantee stronger segmentation performance under the training configuration used here. The lightweight `YoLoV11` and `MobileNet_V3` models offer far smaller parameter counts, but the training curves indicate a corresponding reduction in final overlap quality.

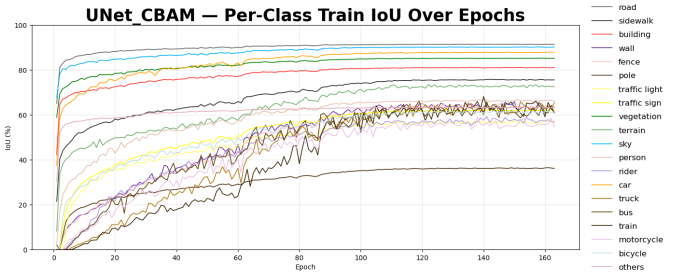

Fig. 6. Overall train IoU for each objects over epochs

Figure 6 provides a class-wise view of train IoU over epochs for the CBAM-enhanced U-Net. The dominant urban classes stabilize earlier and at higher IoU values, especially broad semantic regions such as road, building, vegetation, and sky. In contrast, thinner or less frequent classes show slower growth and noisier trajectories across training. This behavior is consistent with the class-imbalance pattern described in the Dataset section: rare classes such as riders, bicycles, motorcycles, traffic lights, and traffic signs contribute fewer pixels and are therefore more difficult to learn consistently. The per-class trends therefore support the interpretation that the proposed model is strongest at recovering the large structural layout of urban scenes, while smaller and rarer categories remain the more challenging part of the segmentation problem.

After training all, we evaluated the model by using some images in 100 test images that were not involved in the training process.

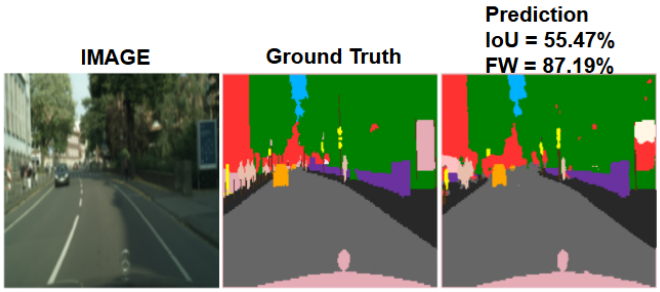

Fig. 7. Segmentation for a sample

Figure 7 shows the segmentation result for one real test image. The left image is the input, the middle image represents the ground truth, and the right image is the predicted segmentation produced by our trained model. Visually, the predicted output closely resembles the ground truth, which explains the high FWIoU score of 87.19%, reflecting strong overall similarity. However, a more detailed comparison reveals that many small or infrequent objects are not accurately captured in the predicted mask. As a result, the mean IoU is only 55.47%. This metric more precisely reflects the model's performance in capturing fine-grained details between the ground truth and the predicted segmentation.

The downstream scene-analysis results are directly coupled to this mask quality. When the segmentation output preserves boundaries and region-level structure more reliably, the derived class coverage estimates, connected-component counts, layout priors, and scene tags are correspondingly more stable and interpretable. When masks are noisier or semantically inconsistent, the later scene-analysis outputs degrade in the same direction. The most defensible conclusion from the present Results section is therefore twofold: first, the CBAM-enhanced U-Net provides the strongest reported segmentation performance in this study; second, improvements at the segmentation

stage translate into more coherent explainable scene-level interpretation.

VIII. Application

The repository is not only a training and evaluation environment, but also an integrated application for live semantic segmentation and explainable urban-scene interpretation. Whereas the earlier sections focus on model design, dataset structure, and training outcomes, the application layer shows how those components operate together in a runnable system. In that sense, the application is the point where the paper's model-centered contribution becomes operational: a trained segmentation network produces the mask, and the mask is immediately reused for visualization and deterministic scene-level reasoning.

At runtime, the main execution surface is a unified Flask application in `app.py`. That application maintains a model registry that stores the user-facing model label, checkpoint source, loader function, and expected input resolution for each available architecture. When a model is requested for the first time, its checkpoint is downloaded, restored, moved onto the configured device, and cached in memory so later requests can reuse it without repeating the setup cost. The runtime also reuses the shared semantic label map, color configuration, and device definitions provided by `application.py`, so training, visualization, and analysis remain synchronized around the same 20-class representation.

The user-facing interface in `templates/index.html` exposes this runtime through an upload-and-analysis workflow. A user can upload an image, choose one or more models, and select which semantic classes should be highlighted in the overlay view. The uploaded image is decoded once, converted into the model's expected RGB tensor representation, resized to the selected architecture's input resolution, and passed through the network. The predicted logits are then reduced by $\arg\max$ to a discrete class mask and expanded back to the original image geometry with nearest-neighbor interpolation so that later visualization and measurement remain aligned with the source image. Importantly, the overlay selection changes only which classes are visually emphasized; it does not alter the underlying full-mask analysis.

That full-resolution mask is then passed directly to the downstream analysis pipeline in `utils/urban_scene_analysis.py`. The analysis stage extracts class coverage, grouped planning categories, layout structure, region statistics, approximate object counts, planning scores, spatial flags, relation flags, warnings, scene tags, and a final summary paragraph. In the application interface, these outputs are rendered not only as a colored overlay, but also as structured interpretation panels and a reasoning box. The reasoning panel is especially important because it combines the higher-level planning summary with class-level count and coverage lines, allowing a user to trace the displayed interpretation back to the predicted semantic content rather than treating the application as a black-box labeler.

The same interface also supports multi-model comparison, which is a practical extension of the paper's broader model-centered framing. One selected model is treated as the primary visualization model, while additional selected models run on the same uploaded image through the same inference and analysis path. Instead of comparing architectures only through raw overlay images, the application compares them at the scene-analysis level. The comparison payload can therefore report shared scene tags, disagreement notes, and large class-level or group-level spread differences across models. This makes the application useful not only for producing one segmentation output, but also for inspecting how different segmentation backbones support or alter the resulting urban-scene interpretation.

Fig. 8. Segmentation for a sample

Taken together, the application in figure 8 is the practical integration point for the paper's full contribution. It links trained segmentation models, semantic overlays, structured reasoning, and deterministic urban-scene interpretation in one operational workflow. As a result, the repository is more than a set of training scripts or benchmark comparisons: it is a runnable system that makes the segmentation output inspectable, comparable, and directly usable for the explainable scene-intelligence layer emphasized throughout this study. We are pleased to invite you to explore our application at the following link: https://huggingface.co/spaces/Azure2212/CityScapeSegmentationWebsite.

IX. Discussion

Relative to the core segmentation papers cited in this work, the present study is best understood as a model-centered analysis built around a known baseline, a lightweight attention refinement, and an operational application layer rather than as a benchmark-reproduction study. The Results section now provides a concrete local comparison across multiple architectures, and within that comparison the CBAM-enhanced U-Net is the strongest reported model in terms of the combined loss and overlap metrics. At the same time, the implementation does not claim to re-create every original training detail, dataset protocol, or experimental ablation from U-Net, FCN, DeepLabv3, MobileNetV3 with LR-ASPP, or Swin Transformer V2 [1], [3]–[6]. Its value lies instead in showing why a U-Net-style encoder-decoder remains a strong urban baseline, how a compact attention mechanism can refine that baseline,

and how those models behave under a shared local training pipeline.

Compared with scene-parsing papers such as PSPNet and SPGNet, the paper takes a different route to scene-level reasoning [9], [10]. Those papers modify the learned network itself so that global context or staged semantic guidance improves the segmentation process. The present system does not insert a learned scene-parsing module into the segmentation backbone. Instead, it treats the predicted mask as the final output of the segmentation model and then performs a second-stage, deterministic interpretation pass. This makes the approach less ambitious as a learned scene model, but more transparent as an explainable analysis pipeline. The scene-analysis outputs can be traced directly back to mask coverage, connected regions, band-wise distributions, and explicit thresholds rather than to implicitly learned internal representations. The new Application section strengthens this contribution by showing that the same segmentation-plus-analysis pipeline is not only a paper abstraction, but also a runnable interface for live inference, overlay rendering, structured planning signals, and cross-model comparison.

HANet offers an especially relevant point of comparison because it exploits structured vertical priors in urban scenes [11]. The current study does not implement height-driven attention or learned urban priors. However, it does rely on the same fundamental insight that urban street scenes have regular structure. The layout-prior logic in the scene-analysis layer checks whether sky is concentrated near the top, road near the bottom, and built or green framing along the edges. In that sense, the project adopts an urban context prior similar in spirit to HANet, but shifts it from a learned segmentation mechanism to a downstream analytic heuristic.

The work also occupies an intermediate position between plain semantic segmentation and fuller scene-understanding formulations. Mottaghi et al. argued for viewing segmentation inside a broader contextual reasoning pipeline [12]. Panoptic Segmentation later formalized a unified representation that combines stuff and thing understanding [13]. The present project does not achieve either a joint probabilistic scene-understanding model or a true panoptic representation. What it does provide is a modest, interpretable step in that direction. Connected components are used as approximate object-region counts, and semantic groups are used to derive broader scene indicators. This is weaker than instance-aware or panoptic reasoning, but richer than a plain semantic color map.

Even with the reported local experimental results, this comparison remains implementation grounded rather than leaderboard oriented. That limitation should be understood as a boundary on generality, not as a claim that evaluation is unimportant. The most important strengths that can already be defended are interpretability, architectural clarity, and operational integration. First, the U-Net baseline and the CBAM enhancement are easy to describe in terms of feature preservation and feature recalibration. Second, the downstream urban-scene outputs can be traced back to semantic coverage, connected regions, band-wise layout, and explicit rules rather than opaque post hoc judgments. Third, the application layer demonstrates that these pieces can be deployed together in one coherent workflow rather than existing only as isolated scripts.

X. Future Work and Limitations

Cityscape segmentation has proven to be essential in many applications, particularly in autonomous driving. However, semantic segmentation alone is not sufficient to enable fully functional self-driving systems. In the future, it is necessary to incorporate depth estimation to determine the distance of each object from the camera. By combining scene understanding with depth information, AI agents—potentially enhanced by strong large language models (LLMs)—can make more informed and reliable decisions for autonomous driving based on comprehensive scene analysis. The whole self-driving car decision in future will be illustrated in figure **??**

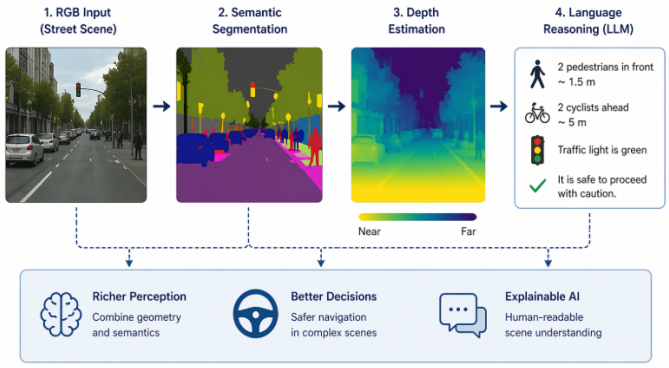

Fig. 9. Segmentation for a sample

The study also has clear limitations. First, the scene-analysis outputs are heuristic. They are explainable, but they are not learned from human-labeled planning judgments, and they are not formal urban measurements. Second, connected-component counts are only an approximation to object instances. Separate blobs in a semantic mask do not provide the same guarantees as true instance or panoptic segmentation [13]. Third, the downstream reasoning assumes that the segmentation mask is sufficiently accurate. Misclassified pixels can affect class coverage, region extraction, adjacency shells, and all later scene tags. Fourth, although the paper now reports local experimental comparisons, it does not yet provide a fully archived benchmark package, an official benchmark protocol, or an exhaustive ablation suite between U-Net and U-Net plus CBAM. The reported comparisons should therefore be interpreted as implementation-grounded results rather than leaderboard claims. Fifth, the CBAM enhancement is architecturally motivated and implemented, but it is not yet accompanied by an exhaustive class-wise error analysis showing exactly which urban categories benefit most. Sixth, the application layer is useful as an interpretable runtime surface, but it still depends on fixed heuristics and single-image inference rather than uncertainty-aware or temporally consistent reasoning.

These limitations also suggest clear directions for future work. The most immediate next step is to archive the full experiment package more rigorously so that the paper's reported results can be reproduced through a cleaner benchmark record and more controlled comparisons between the baseline U-Net and the CBAM-enhanced variant. Beyond that, deeper CBAM ablation, stronger class-wise error analysis, and more explicit reporting of checkpoint provenance would improve the experimental story. On the application side, the scene-analysis layer could be extended with uncertainty-aware reasoning, richer instance-level perception, temporal consistency across video, or learned context modules that preserve some degree of interpretability. The present work is therefore best viewed as a strong starting point: it demonstrates how a classical segmentation model, a lightweight attention refinement, and a deterministic application-facing analysis layer can support richer urban-scene interpretation, while leaving substantial room for more rigorous evaluation and more advanced scene-understanding mechanisms.

XI. Conclusion

This paper presented a model-centered view of urban scene segmentation and interpretation built around U-Net and CBAM-enhanced U-Net. The baseline model uses an encoder-decoder architecture with skip connections to recover detailed pixel labels, while the attention-augmented variant refines selected skip features through sequential channel and spatial recalibration. In the reported local comparison, the CBAM-enhanced variant delivered the strongest overall combination of loss, pixel accuracy, mean IoU, and frequency-weighted IoU. Together, these models provide the learned perception layer for a Cityscapes-based urban-scene workflow.

The paper also showed that the predicted mask can support more than visualization. By treating the mask as an intermediate semantic representation, the downstream analysis layer derives class coverage, semantic groups, approximate object counts, layout priors, relation flags, scene tags, and summary text in an explainable manner, while the integrated application layer operationalizes that workflow through model selection, overlay rendering, and structured runtime interpretation. The resulting contribution is modest but meaningful: it links a classical segmentation baseline, a lightweight attention refinement, a deterministic scene-intelligence module, and a runnable application into a coherent urban AI system. Future work can strengthen this foundation with more rigorous experiment archiving, deeper ablation of the CBAM enhancement, and richer scene-understanding mechanisms.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv preprint arXiv:1505.04597*, 2015.

[2] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proc. Eur. Conf. Comput. Vis.*, 2018.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[5] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[6] Z. Liu *et al.*, "Swin Transformer V2: Scaling Up Capacity and Resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12009–12019.

[7] M. Cordts *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.

[8] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A Review of Semantic Segmentation Using Deep Neural Networks," *Int. J. Multimedia Inf. Retrieval*, vol. 7, no. 2, pp. 87–93, 2018.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[10] B. Cheng *et al.*, "SPGNet: Semantic Prediction Guidance for Scene Parsing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5218–5228.

[11] S. Choi, J. T. Kim, and J. Choo, "Cars Can't Fly Up in the Sky: Improving Urban-Scene Segmentation via Height-Driven Attention Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9373–9383.

[12] R. Mottaghi, S. Fidler, J. Yao, R. Urtasun, and D. Parikh, "Analyzing Semantic Segmentation Using Hybrid Human-Machine CRFs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3143–3150.

[13] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic Segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9404–9413.