Annexure3b- Complete filing

INVENTION DISCLOSURE FORM

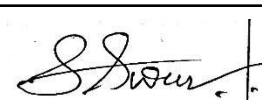
Details of Invention for better understanding:

1. TITLE: A Vision-Based Intelligent Parking Slot Monitoring and Occupancy Duration Prediction System Using Deep Learning and Web-Based Visualization

2.INTERNAL INVENTOR(S)/ STUDENT(S): All fields in this column are mandatory to be filled

A. Full name	Saragada Thrinath
Mobile Number	6305015647
Email (personal)	saragadathrinath@gmail.com
UID/Registration number	2318051
Address of Internal Inventors	Lovely Professional University, Punjab-144411, India
Signature (Mandatory)	

INTERNAL INVENTOR(S)/ STUDENT(S): All fields in this column are mandatory to be filled

A. Full name	Sangani Swamy
Mobile Number	6302934271
Email (personal)	swamysangani21@gmail.com
UID/Registration number	2326532
Address of Internal Inventors	Lovely Professional University, Punjab-144411, India
Signature (Mandatory)	

INTERNAL INVENTOR(S)/ STUDENT(S): All fields in this column are mandatory to be filled

A. Full name	Shaik Irshad
Mobile Number	125622399
Email (personal)	irshadshaik695@gmail.com
UID/Registration number	2317390
Address of Internal Inventors	Lovely Professional University, Punjab-144411, India
Signature (Mandatory)	Irshad

3.DESCRIPTION OF THE INVENTION: This invention is about an intelligent, vision-based system to manage parking that automatically monitors and analyzes the use of parking spaces through computer vision and deep learning techniques. The system gets the real-time video feed from one or more cameras and uses the predefined regions of interest (ROIs) for the individual parking slots. Each ROI is separately analyzed by a trained convolutional neural network (CNN) to determine if the slot is occupied or empty in real time.
The system goes beyond just availability detection as it keeps track of the changes in the occupancy state of each parking slot and calculates the time for which a slot is occupied. These time-based events are automatically recorded with slot identifiers and timestamps, thus facilitating the analysis of parking behavior, usage patterns, and peak demand periods over time.

The invention comprises a lightweight full-stack web-based dashboard that shows live camera views, real-time parking slot status, confidence-aware classification outputs, occupancy timers, and historical usage records. The dashboard lets administrators and users visually check parking availability and get data-driven insights without the need for manual intervention or physical sensor infrastructure.

Since the system is based solely on camera vision and does not require embedded hardware sensors, it is less costly to deploy, easier to maintain, and can be scaled to different-sized parking areas without any problem. The combination of spatial intelligence (slot-wise classification) with temporal intelligence (occupancy duration tracking) helps the system to be used for advanced parking analytics, operational optimization, and smart parking facilities as well as smart city applications for informed decision-making.

4.Technical Implementation:

The technology has a modular design, full-stack architecture that includes a backend service that handles live video processing and data management and a frontend application that handles visualization and user interaction. To ensure that processing is done with minimal delay and that API handling can be scaled up without any problem, the backend is implemented using Flask or FastAPI. At the same time, the frontend is built using React to deliver an engaging and responsive UI.

1.System components
• Frontend (React)
The frontend is implemented as a web-based graphical dashboard that provides real-time visibility into parking activity. It displays live camera feeds streamed from the backend, current availability of each parking slot, timers showing how long each slot has been occupied, and historical parking usage logs. The frontend communicates with the backend through REST APIs and streaming endpoints to periodically retrieve updated data, which is rendered in a structured, interactive, and user-friendly manner.

• Backend (Flask / FastAPI)
The backend is like a brain that controls the whole system with the help of the other parts. It takes

the live video from the cameras and uses computer vision techniques to enhance the images. By using CNN models, the backend identifies which parking slots are occupied and which are empty. To know for how long, the backend also keeps some timers to track the duration of the slot occupancy and hence it has a record of all parking events. In addition, the backend sends the video with overlays to the frontend via MJPEG or other similar protocols. Also, it has APIs to give the real-time updates and historical data to the users.

• Inference & Processing Modules

The system design is very modular with inference and processing scripts like collect_data.py, train_model.py, realtime_inference.py which do the following:

- Point out ROIs in the image showing one parking slot
- Take pictures of slots in the image and create a labeled data set
- Teach CNN models to recognize parking slots
- Perform batched real-time inference on video frames
- Detect the transition from empty to occupied and vice versa
- Compute occupancy time and create structured logs

In effect, these modules are the main idea that the system uses to realize the parking situation automatically and provide subsequent analysis.

• Key Dependencies

The innovative work cannot be done without the system dependencies that the authors chose to be efficient and popular libraries:

- Pytorch and Torchvision for CNN training and inference
- OpenCV for the operations like grabs, video preprocessing, ROI extraction, and overlay drawing
- NumPy for numerical operations
- Flask or FastAPI for API services and video streaming

These dependencies make operation possible and dependable in real-time on a wide variety of devices.

• Data Storage

The system is designed to organize data in a structured manner:

- /data/raw/ is a place where the system stores the cropped images of the parking slots that are the input data for training. These images are categorized into empty and occupied ones.
- /data/processed/rois.json is a file where the parking slot coordinates are stored.
- /logs/occupancy_logs.csv is a file in which the timestamps, slot identifiers, occupancy durations, and historical usage patterns are stored.

This design provides traceability and convenience when doing historical analysis.

2. **System Workflow:**

Live Camera Feed Initialization: The backend initiates a camera input and then keeps on capturing video frames. The predefined ROIs are fetched from the configuration files and superimposed on each frame so that the pixel areas can be linked with the respective parking slots.

Frontend-Backend Interaction: The frontend is in charge of sending API requests to the backend for fetching slot status, occupancy duration, and historical data in real-time. The live camera previews are being streamed through MJPEG endpoints in order to achieve low-latency visualization.

Frame Processing & Inference: On getting the frame, the backend will do the cropping based on ROI definitions for each parking slot shown in the frame, and then it will do the necessary preprocessing. The batch mode inference using the CNN model goes through the frame to label each parking slot as either occupied or empty. Also, it calculates the confidence levels for each prediction, and if there is a change in the state of the slot, it will notify the update of its internal timers.

Status Tracking and Logging: A slot-state monitoring short-term inference loop is always going on. When a slot is at the point of being changed from empty to occupied, the start timestamp is recorded. When the moment occurs that such a slot goes from occupied to empty, the whole period is then calculated and saved to the occupancy log file. Thus, it provides a safe way to store the records on how parking spaces have been used over time.

Data Delivery to Frontend: The backend collects and dispatches combined data to the frontend. Among other things, these data include the status of the current slots, the confidence scores, the active occupancy duration, and the historical usage patterns. The dashboard is being updated at the moment to represent the latest situation and to show the trends.

5.Novel Features of the Invention

Real-Time Dual-Level Intelligence: The implemented system merges spatial intelligence (using CNNs for slot-wise occupancy classification) and temporal intelligence (non-stop occupancy duration tracking). So, the system analyses on two levels at the same time, thus, it yields far more detailed and deeper insights than those of mere sensor-based or object detection-only systems.

Robust and Safe Video & Data Handling Pipeline: The patented method utilizes predefined ROIs for exact slot extraction, batched inference for operational efficiency, structured logging of occupancy events, and the proper handling of video frames and data files. The system's stability is not compromised even at high loads or in case of partial failures.

Efficient Deep Learning Integration for Real-Time Deployment: The back-end is combined with PyTorch-based CNN models that are FastAI-based and efficient for deployment in real-time with batch processing, resolution control, and a fallback on CPU or Apple MPS mode. This, in turn, makes possible the deployment of a unit on a laptop, an edge device, or a cloud platform that is suitable for a smart city infrastructure.

Lightweight Full-Stack Web Dashboard for Instant Visualization: The solution comes with one integrated web dashboard that constitutes a tool for displaying the things that are going on at the moment in front of the camera, showing slot availability, confidence values, occupancy timers, and historical analytics all in one single interface, thereby making both the monitoring process and decision-making much easier.

6.Potential Applications

Parking Efficiency Optimization: Real-time spotting of free parking spaces is made possible through camera-based computer vision and a trained CNN.Users or administrators can, therefore, with minimum effort locate the most free spaces. Parking searching time is, thus, shortened, co2 emissions due to traffic are decreased, and manual monitoring is totally avoided.

Automated Slot Detection & Management: This approach relies on an AI system that only uses visual data from predefined areas of parking slots to perform automatic classification of each slot as occupied or empty. The elimination of physical sensors or human checking is the main advantage of the system while the ability to easily deploy across small parking areas as well as large smart city environments are its main features.

Usage & Occupancy Analytics: Moreover, it keeps the record of the duration of occupancy of each parking slot and stores the historical usage data. Parking managers are thus able to identify the behavior patterns, optimize pricing scenarios, make the best use of the available spaces, and take decisions based on data which leads to more efficient resource allocation.

7.PROBLEM ADDRESSED BY THE INVENTION:

The system is designed to overcome the major problems that come with parking management such as the inefficient use of parking spaces, the increased time of searching for available spaces, and the reliance on manual monitoring or sensor-based detection. Most of the time, traditional parking solutions depend on the installation of physical sensors in every parking space, which demand a lot of money to be deployed, need regular maintenance, and are difficult to expand, or they rely on human supervision, which is not always stable and effective. The invention that is being proposed gets rid of these drawbacks by using a vision-based artificial intelligence method that involves standard cameras and trained convolutional neural networks to automatically identify the occupancy status of individual parking slots in real time. The system monitors each predefined parking slot as occupied or empty and tracks the occupancy duration for every slot; thus, the system produces in-depth information about the parking behavior, such as the arrival times, the departure times, the average stay durations, and even the peak usage periods. This automated approach, which is free from sensors, is more accurate and scalable and at the same time, it lowers the operational costs. The availability of both real-time and historical data on occupancy makes it possible for parking management to be done in a more intelligent manner, the overall utilization of the space gets better, and it also leads to the easing of the traffic flow and the more efficient parking operations in urban and smart city environments.

8.OBJECTIVE OF THE INVENTION (Provide minimum two)

• The ultimate objective is to offer a precise and real-time estimation of the vacant parking spaces by implementing a vision-based deep learning model. In this case, a trained convolutional neural network is the one that, with great confidence, identifies each parking space as an occupied one or an empty one among the ones previously defined.

• By regularly monitoring occupancy duration through continuous tracking and keeping historical records of slot utilization, the system can provide intervention-ready, automated insights about parking usage. As a result, efficient parking management, better space allocation, less congestion, and smoother traffic flow will be possible.

9.STATE OF THE ART/ RESEARCH GAP/NOVELTY:

Sr. No.	Existing Art / Patent / Product	Abstract (Condensed)	Research Gap	Novelty vs Proposed System
1	**CN118675328A (CN, 2024)** Intelligent park parking space prediction method	Camera-based CNN detects slot occupancy; RNN analyzes historical occupancy to predict future slot status; outputs availability trends in real time.	No explicit user intent conditioning; no calibrated confidence or prediction intervals; no explainability; no ANPR-based audit logging; continual learning and drift handling not specified.	Proposed system introduces intent-conditioned per-slot remaining-time prediction with calibrated confidence, explainability, auditability, and edge-level continual learning.
2	**ESWA 2023 – APSD-OC** Automatic vision-based parking slot detection and occupancy classification	Vision-only detection of parking slots and their occupancy using deep learning; focuses on accuracy and automation.	No prediction of duration or vacancy time; no historical learning; no intent, uncertainty modeling, explainability, or learning loop.	Proposed system extends beyond detection into predictive, intent-aware, confidence-calibrated decision making.

3	**EP1895486A1 (EU)** Automated parking system with ANPR	Uses CCTV and ANPR to log vehicle entry/exit, duration, and enforcement evidence.	No deep-learning occupancy modeling; no per-slot prediction; no confidence metrics; no explainability or adaptive learning.	Proposed system combines modern CNN occupancy, predictive analytics, and confidence estimation while retaining audit capability.
4	**Embedded vision-based parking detectio** (IJIRT, ~2021)	Lightweight CNN performs real-time occupancy detection on edge devices.	Detection only; no duration modeling, no intent, no confidence intervals, no explainability, limited adaptation.	Proposed system adds temporal prediction, intent conditioning, uncertainty calibration, and continual learning at the edge.
5	**US10121172B2** Parking lot monitoring system	Camera-based monitoring of parking spaces and events for management purposes.	No predictive remaining-time estimation; no intent awareness; no uncertainty or explainability; no learning loop.	Proposed system introduces per-slot temporal intelligence and predictive guidance absent here.
6	**Uncertainty-Aware Parking Prediction** (BNN, 2025)	Bayesian neural networks estimate parking availability with uncertainty and calibration; uses contextual features.	Not vision-only; mostly area-level; no per-slot remaining-time; no intent input; no audit linkage.	Proposed system applies calibrated uncertainty at per-slot level using camera-only data and intent conditioning.
7	**US 11,928,964 B1 / INRIX family** Predicting parking availability	Uses historical occupancy and vacancy durations to estimate availability or wait time.	Data-source agnostic; typically zone-level; no camera-only constraint; no intent semantics; no explainability.	Proposed system narrows scope to vision-only, per-slot, intent-aware, confidence-calibrated prediction.

8	**IEEE / ArXiv camera-only parking guidance systems**	Vision-based occupancy detection with user guidance based on current availability.	No predictive time-to-vacancy; no uncertainty; no intent; no learning loop.	Proposed system enables forward-looking, confidence-aware guidance.
9	**Commercial ANPR parking systems** (SmartParking, Tattile)	Vision-based license plate recognition for access control, duration tracking, and enforcement.	No occupancy prediction; no per-slot analytics; no confidence or explainability.	Proposed system fuses ANPR auditability with predictive intelligence and learning.
10	**US20090202105A1** Automatic license plate recognition	Foundational ALPR for vehicle identification.	No parking prediction or learning; no occupancy intelligence.	Proposed system uses ALPR as a supporting audit layer within a predictive framework.

10.RESULTS AND ADVANTAGES:

1. Results

The system successfully performs real-time video processing and AI-based slot assessment, producing two primary outputs:

Real-time occupancy classification for each predefined parking slot
(e.g., { "A1": "occupied", "A2": "empty", ... }) powered by a trained CNN model.

Continuous occupancy duration tracking and historical logs, generating
(e.g., { "slot": "A1", "duration_seconds": <number>, "timestamp": <ISO> })
which allow deeper insights into parking usage, overstays, and peak-time patterns.

These outputs collectively form a robust analytics pipeline enabling intelligent parking management.

2. Advantages

• Automated, Accurate Parking Management
Instead of manual checking and hardware-based sensors, the system employs AI-driven computer vision that is able to give real-time parking availability detection that is trustworthy and exact.
• Operational Efficiency & Data-Driven Insights

Nonstop occupancy tracking and historical data collection give local administrators a chance to grasp demand trends, arrange parking layouts in an optimal way, enforce policies, and manage traffic by making data-driven decisions.

• Scalable and Flexible Architecture
The modular backend architecture which is realized through Flask or FastAPI and exposes standardized APIs and live video streams that are easily integrative with modern web frontends thus allow you to be deployed in different places without the need for specialized infrastructures.

• Robust Resource Management
On top of that, the system assures consistent performance by means of well-tuned video handling, usage of predefined ROI, and batched inference procedures. During prolonged operation and high-traffic conditions, the system retains its reliability while still being able to minimize computational overhead.

11.EXPANSION:
Variables and Functional Aspects Covered in the Patent Claims

1. Parking Slot ROI Definitions
Set of predefined Regions of Interest (ROIs) for individual parking slots comprising spatial coordinate boundaries, slot identifiers, and mapping logic for the application of these ROIs to each incoming video frame for independent slot-wise analysis.
2. AI-Based Slot Classification Models
A trained convolutional neural network (CNN) model together with an associated preprocessing pipeline, including image normalization, resizing, and batch inference, configurable to the classification of a parking slot as occupied or empty. The system also provides for confidence scoring and a mapping mechanism that associates the classification outputs with specific parking slot identifiers.
3. Occupancy Duration Tracking Logic
A time-tracking mechanism that identifies slot state changes (empty to occupied and occupied to empty), records corresponding timestamps, computes total occupancy duration, and creates structured historical logs representing parking behavior and usage patterns.
4. Live Video Processing and Error Handling
An ensemble of video processing techniques that manage camera startup, frame grabbing, and the usage of different input sources, e.g., USB cameras, IP cameras, or integrated webcams. The system comprises fallback and recovery provisions to cope with frame corruption, camera disconnection, invalid ROIs, resolution mismatches, and inference-related errors while at the same time, the performance of continuous real-time operation is not affected.
5. Frontend–Backend Synchronization Protocols
The protocol dictating the communications of the inference backend with the web-based frontend, specifying the data exchange frequency, live video streaming formats like MJPEG, uniform JSON data structures for slot status and occupancy metrics, as well as low-latency, resource-efficient update mechanisms that do not overload system components.
6. Scalability and Deployment Parameters
The features of the system design allow for scalability in deployment, such as the execution of batched inference, optimized frame processing, the modular replacement of different AI models or hardware backends (CPU, GPU, MPS), and the ability to monitor multiple camera feeds or a large parking area with minimal changes to the architecture

12.USE AND DISCLOSURE (IMPORTANT): Please answer the following questions:

A. Have you described or shown your invention/ design to anyone or in any conference?	YES ( )	NO ( )✓
B. Have you made any attempts to commercialize your invention (for example, have you approached any companies about purchasing or manufacturing your invention)?	YES ( )	NO ( )✓
C. Has your invention been described in any printed publication, or any other form of media, such as the Internet?	YES ( )	NO ( )✓
D. Do you have any collaboration with any other institute or organization on the same? Provide name and other details.	YES ( )	NO ( )✓
E. Name of Regulatory body or any other approvals if required.	YES ( )	NO ( )✓

13.KEYWORDS: Please provide the right keywords for searching for your invention.

1. Vision-Based Smart Parking System
2. AI-Driven Parking Slot Detection
3. Computer Vision–Based Parking Monitoring
4. Parking Slot Occupancy Classification
5. CNN-Based Parking Analysis
6. Real-Time Video Inference
7. Region of Interest (ROI) Mapping
8. Parking Occupancy Duration Tracking
9. Automated Parking Management System
10. Video-Based Parking Analytics
11. OpenCV-Based Video Processing
12. Deep Learning Inference Engine
13. Web-Based Parking Dashboard
14. MJPEG Video Streaming
15. Historical Parking Usage Logging
16. Edge AI Deployment for Smart Cities
17. Scalable Camera-Based Parking Infrastructure

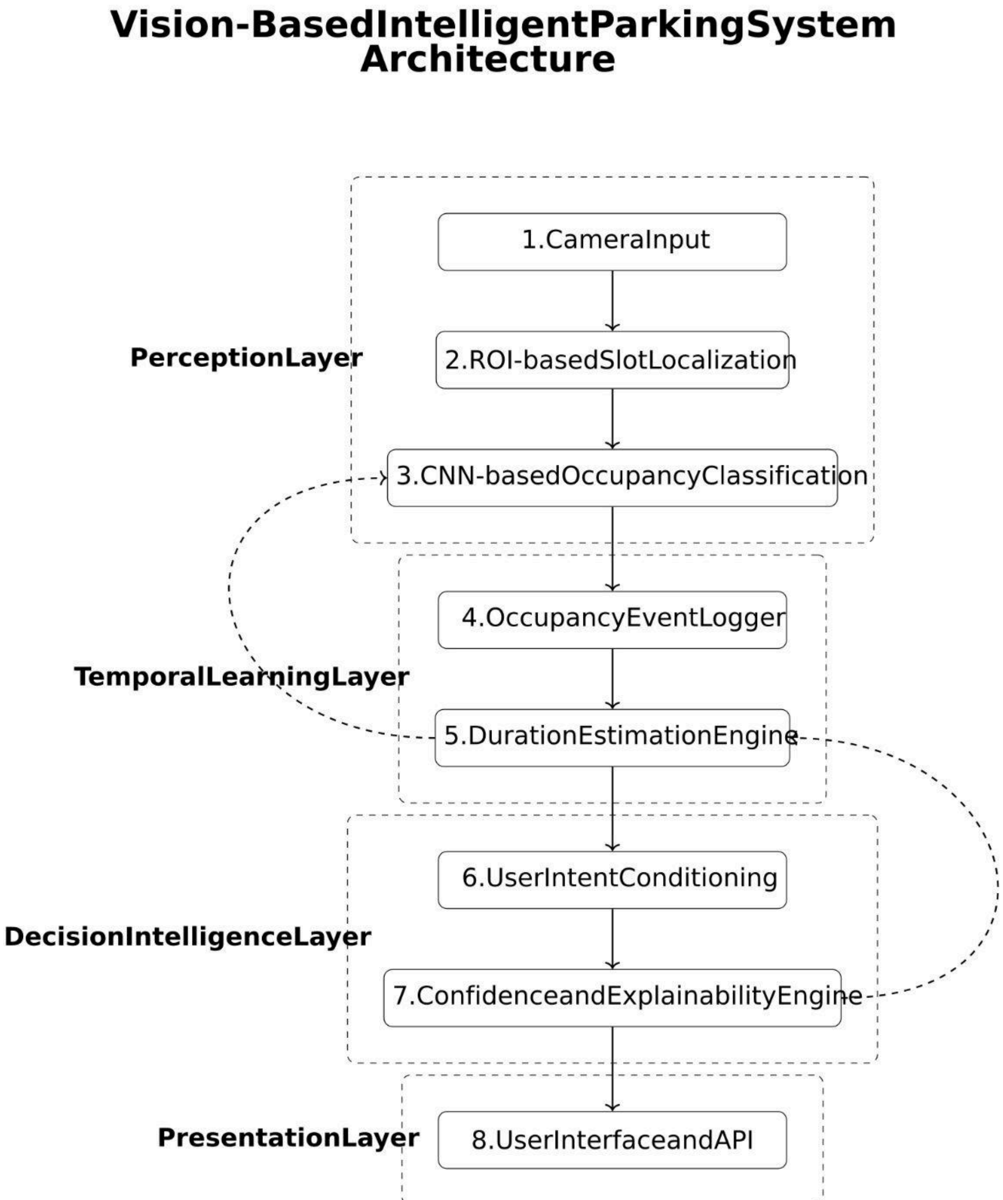
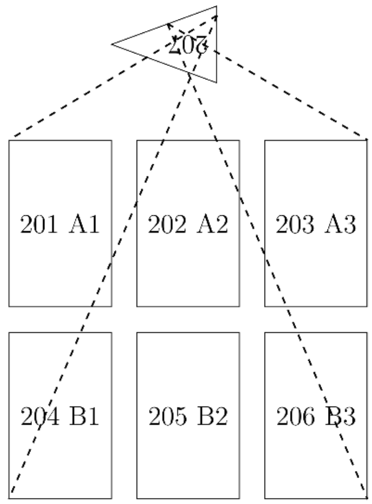


Figure1:Layeredarchitectureofthevision-onlyintelligentparkingsystemwithlearning feedback



208 Parking Area Coverage

201–206 (Parking Slots)

Labels 201 through 206 correspond to the individual parking spaces that have been organized visually. From each slot, the system obtains unique identifiers and further treats them as independent analytical units. The geometry and the relative distances of the slots are a part of the system calibration which is the stage where the physical layout is fixed. The slots then become a reference to which the monitoring system can compare across different periods of time.

207 (Camera Unit)

207 is a single monocular camera of fixed position, which is looking down from a parking area, and is mounted somewhere above and around the area. It has an arrangement such that a top-down or an angled view would get recorded where up to multiple parking slots can be seen at once. No other sensing device is necessary. All the subsequent inferences that the system makes, such as the detection of occupancy and the temporal analysis, are solely dependent on the camera feed.

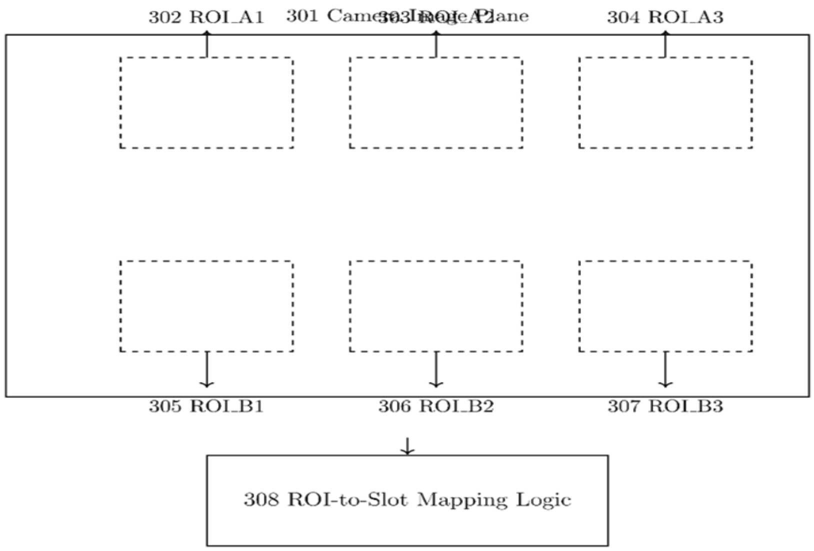
Field of View Boundaries (Dashed Lines)

The dashed lines are the actual camera boundaries of the real field of view. Within these boundaries lie the spatial regions where the system observes parking slots. This system dynamically verifies that it is only those slots that are fully or partially within the region that are considered for inference purposes.

208 (Parking Area Coverage)

208 is a label that signifies the total area of the monitored parking lot. This place is a combination of multiple parking slots that are under the coverage of a single camera. The system has the capability of scalable deployment in such a way that there could be several such zones each with its independent monitoring by a corresponding camera unit.

Figure 3: ROI Mapping and Slot Localization



301 Camera Image Plane

301 visualizes the live video frame from a single monocular camera, which is fixed and aimed to cover the entire parking area. The camera is constantly fed with the visual data that contains the multiple parking spaces all in one field of view.

302–307 Regions of Interests (ROIs)

Each dashed rectangular region (302 to 307) indicates a different Region of Interest that has been mapped to a physical parking space. These ROIs are either manually or semi-automatically defined during the establishing stage and they remain spatially fixed with respect to the image plane of the camera.

302–304 denote parking spaces in the first row (Row A)

305–307 denote parking spaces in the second row (Row B)

Every ROI focuses on the visual area of a single parking space and thus, there is no interference from nearby parking spaces, vehicles, or any other background objects.

308 ROI-to-Slot Mapping Logic

The area 308 stands for a logical mapping layer that connects each ROI with a unique identifier of a parking slot. This association allows the consistent tracking of the occupancy state, duration, and historical behavior at each slot locally and over time.

Mapping logic is like a visual passport control - by analyzing clips in each ROI it "checks" what slot the data belong to, hence the info about that slot can be logged, predicted and reported to the user.

Purpose and Technical Advantage

Figure 3 explains the initial concept that helps to convert the unprocessed images taken by the camera into detailed and slot-specific visual data. The system accomplishes:

Deterministic per-slot monitoring using a single camera

Scalability to multiple slots without any additional sensors

Reliable downstream learning of occupancy duration and behavior

This figure is the enabler of the entire pipeline that follows: occupancy classification, time-to-vacancy prediction, intent conditioning, confidence estimation.

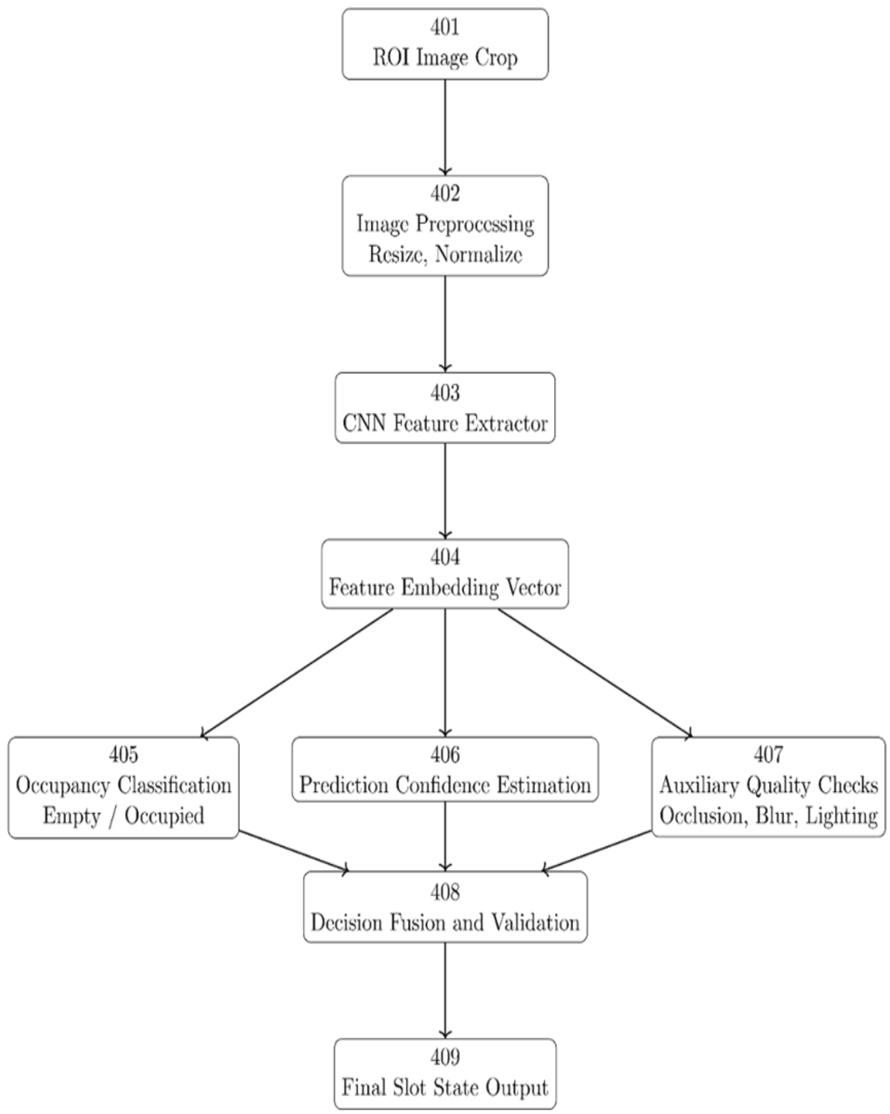


Figure 1: Advanced CNN-Based Parking Slot Occupancy Classification Architecture

401 ROI Image Crop

Region of Interest coordinates that were predefined and obtained from camera calibration are used to isolate each parking slot individually. Slot-specific visual data is the only thing that is processed, thus background noise is completely removed and less computational power is needed.

402 Image Preprocessing (Resize, Normalize)

The local image of the cropped ROI is resized to a certain input resolution and is normalized by standard mean and variance values. The step taken here is to establish the consistency of the different factors such as lighting condition, camera resolution, and weather condition.

403 CNN Feature Extractor

To generate both high-level spatial and semantic features from the input, a lightweight convolutional neural network is employed over the preprocessed image. The network is trained specifically for parking environments, enabling robust detection under partial occlusion, shadows, and perspective distortion.

404 Feature Embedding Vector

The compact numerical embedding which visually represents the parking slot is generated by the CNN. This embedding is the common representation for many downstream decision branches.

405 Occupancy Classification (Empty / Occupied)

The embedding vector is localised by a classification head which identifies the binary occupancy state of the slot. The principal machine decision that consists of the most up-to- date availability changes is produced by this section of the module.

406 Prediction Confidence Estimation

Besides classification, a confidence estimation head also estimates the confidence of the predicted state. The confidence score produced reflects the level of certainty of the model and it is later utilized for the user display, filtering, and other risk-aware decisions that happen downstream.

407 Auxiliary Quality Checks (Occlusion, Blur, Lighting)

One such auxiliary branch looks at various factors that determine image quality: occlusion level, motion blur, and illumination anomalies. It serves as a protective layer against the presence of unreliable visual conditions, which in turn should result in lowered prediction accuracy.

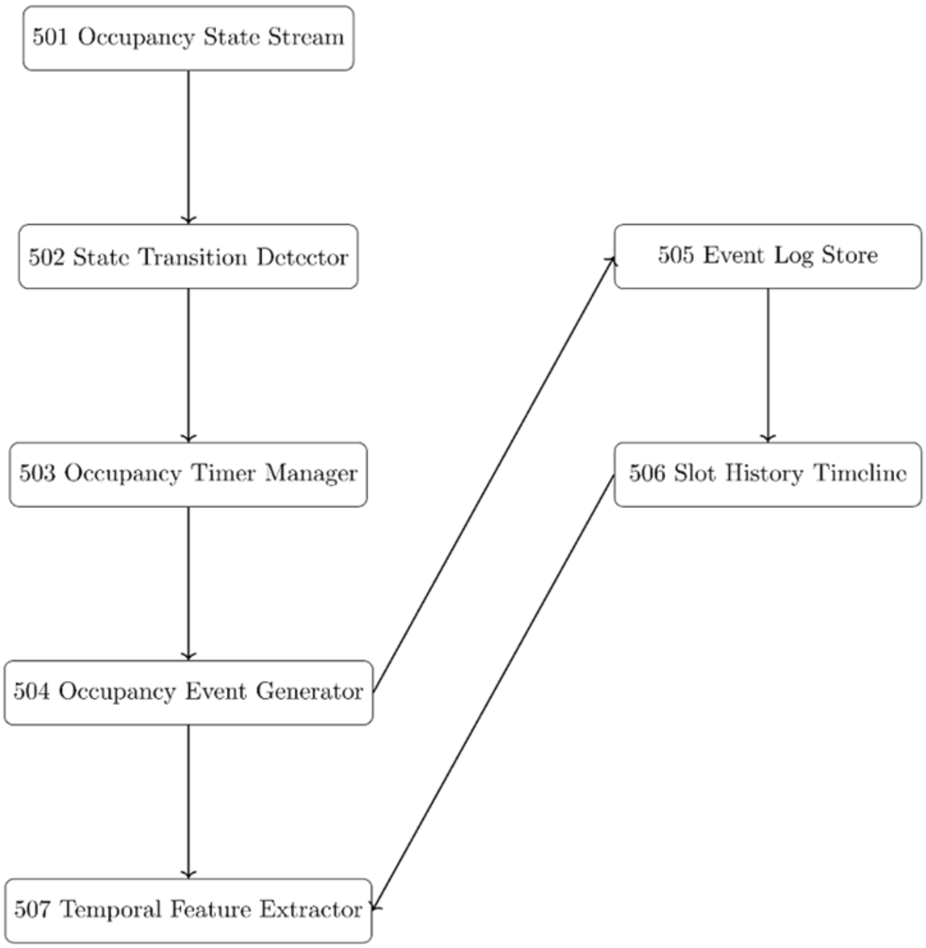
408 Decision Fusion and Validation

The fusion logic combines together the results of classification, confidence estimation, and quality check modules. Across this level, decisions are not only confirmed, but the suppression of unreliable predictions and the assurance of stable system running are also performed.

409 Final Slot State Output

Emitted as the final output is the slot state which had been validated together with its confidence metadata. The result is then sent to the occupancy logger, prediction engine, and user interface for real-time visualization and analytics.

Figure 5: Occupancy Timeline and Event Logging



501 is a depiction of a continuous stream showing the different states of slot-level occupancy that have been inferred by the vision-based classification module. Each of these signals stands for a specific parking slot and shows the slot's detected state over time.

502 is a location where frame-level occupancies from successive instances get temporarily stored. This storage is done to smooth the short-term noise and provide a stable input for state transition analysis.

503 scans through the occupancy signals that are buffered to figure out the changes in the state. It identifies those state changes that result to a significant increase or decrease of the parking spaces, such as from empty to occupied or from occupied to empty and also changes that include wrongly classified transient states, which it filters out.

504 is a device that keeps a model of states for each parking slot and ensures that transitions that only make sense are the ones accepted and that occupancy events are the ones that are tracked continuously over time.

505 is the one that starts and updates the time counters when a slot is in an occupied state. The timer is kept going until an exit event is registered.

506 is responsible for the creation of the discrete entry and exit events that indicate the arrival and the departure of a vehicle respectively. The events thus created are the duration calculation and historical logging triggers.

507 keeps a record of all the entry and exit events in the form of a log, which is stored in the persistent storage system, thereby providing a parking activity log that can be referred to without the need for manual camera footage review.

508 keeps comprehensive historical data that is organized for each parking slot and includes such things as occupancy durations, timestamps, and event sequences as well as being able to provide long-term behavioral analysis.

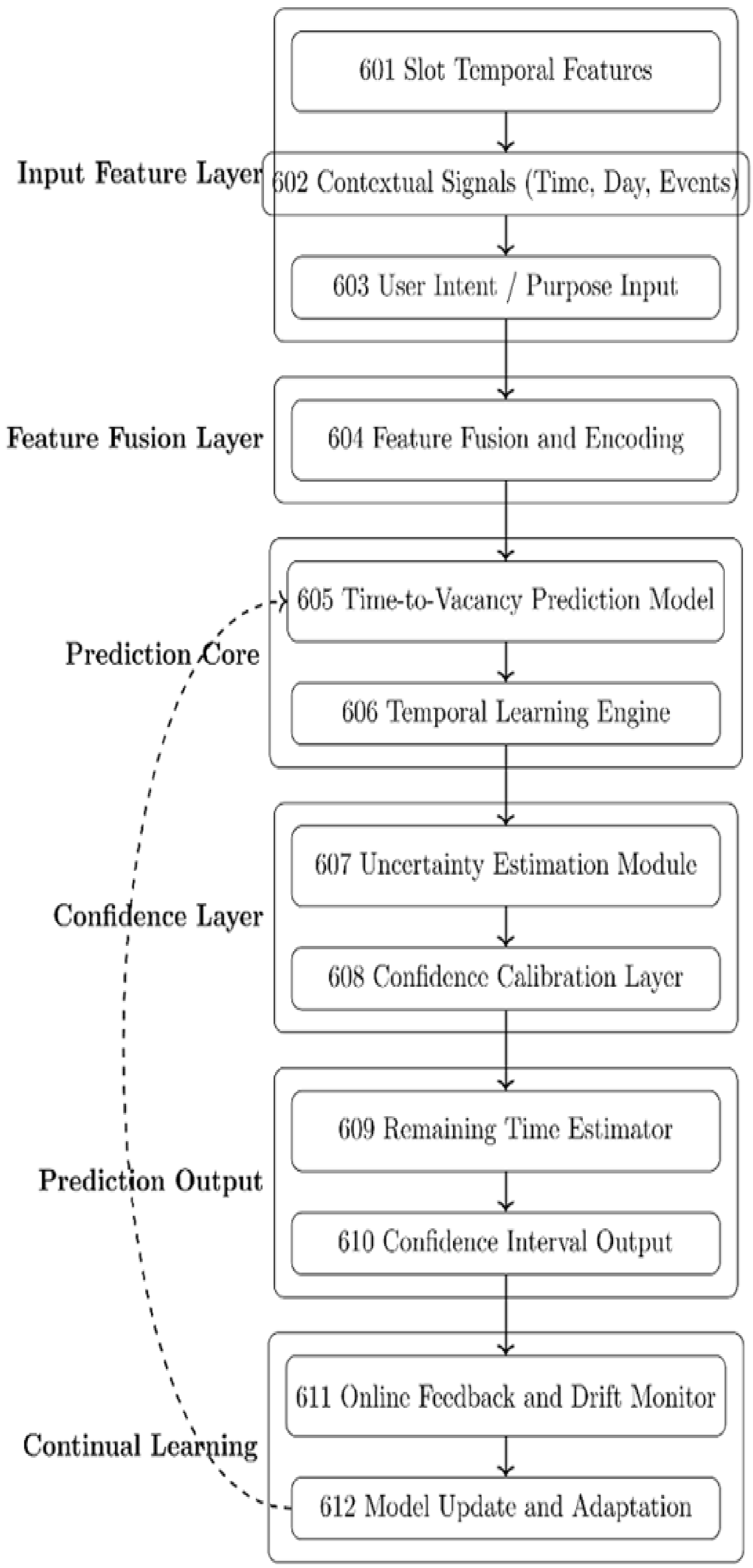
509 is responsible for the extraction of statistical and temporal features from the historical occupancy data. Such features could be the average dwell time, variance, time-of-day patterns, and usage frequency.

510 releases aggregated duration metrics as well as learned temporal patterns, which are then used by predictive models for the estimation of the remaining occupancy time as well as the probability of vacancy.

Technical Significance

Figure 5 outlines the changes that the raw occupancy classes go through before they become well-structured, time-aware behavioral data. The pipeline makes possible accurate duration tracking, historical learning, and downstream prediction, all of which are done without the need for any camera or sensor-based device.

Figure 6: Time-to-Vacancy Prediction with Intent Conditioning and Confidence Estimation



601 represents slot-level temporal features derived from historical occupancy events, including dwell durations, entry-exit patterns, and recurrence statistics for a given parking space.

602 denotes the contextual signals related to the occupancy event, such as the time of day, the day of the week, calendar effects, and localized traffic or facility activity patterns.

603 corresponds to an explicit user-provided intent or purpose indicator, which semantically classifies the expected parking behavior, for example shopping, dining, work, or other activity types.

604 performs feature fusion and encoding, combining temporal, contextual, and intention-based inputs into a single representation suitable for downstream prediction.

605 Is the core time-to-vacancy prediction model that estimates the total expected occupancy duration for the active parking event at the slot level.

606 represents the temporal learning engine, which captures sequential dependencies and evolving behavior patterns across repeated parking events.

607 is the uncertainty estimation module, which evaluates the prediction variability arising from behavioral diversity, limited data, or environmental changes.

608 applies confidence calibration to convert raw uncertainty estimates into reliable confidence scores or prediction bounds.

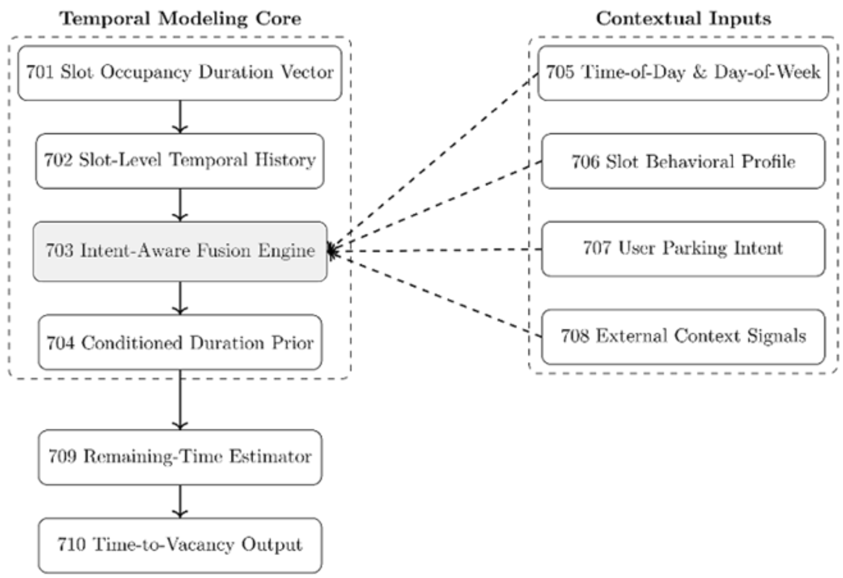
609 computes the remaining time-to-vacancy by adjusting the predicted duration using the elapsed occupancy time.

610 outputs a confidence-aware remaining time estimate including calibrated intervals suitable for user display or system decision making.

611 monitors online feedback and concept drift by comparing predicted outcomes with observed exit events over time.

612 performs adaptive model updating, allowing continual learning and recalibration of slot-specific behavior without the need for manual retraining.

Figure 7: User Intent Conditioning and Context Fusion



701 Slot Occupancy Duration Vector

The continuously updated numerical vector capturing the duration for which each parking slot has remained occupied or vacant is this component of the system. The vector is generated only from camera-based occupancy events.

702 Slot-Level Temporal History

This block is storing a detailed historical record of the occupancies of each individual slot, which includes repeated patterns of use that were different times and days.

703 Intent-Aware Fusion Engine

This core decision module is where the temporal slot behavior is merged with the contextual and user-provided intent signals. It combines the historical occupancy patterns with the real-time contextual inputs to change the duration of the expectations dynamically.

704 Conditioned Duration Prior

This output is a more accurate probabilistic prior of the expected parking duration for a slot that is dependent on both the behavior being historical and the intent/context signals being current.

705 Time-of-Day and Day-of-Week Context

This input is the temporal context that includes the hour of the day and the weekday or weekend indicators, which have an impact on the typical parking behavior patterns.

706 Slot Behavioral Profile

This component models the long-term behavioral characteristics of each parking slot such as average dwell time, peak usage windows, and variability over time.

707 User Parking Intent

This input is an explicit declaration of the parking purpose, for instance, shopping, dining, cinema, or work, which is utilized to select and weigh the relevant duration priors.

708 External Context Signals

This block is the source of optional contextual modifiers such as holidays, events, or abnormal traffic conditions, which can have an impact on the parking duration distributions.

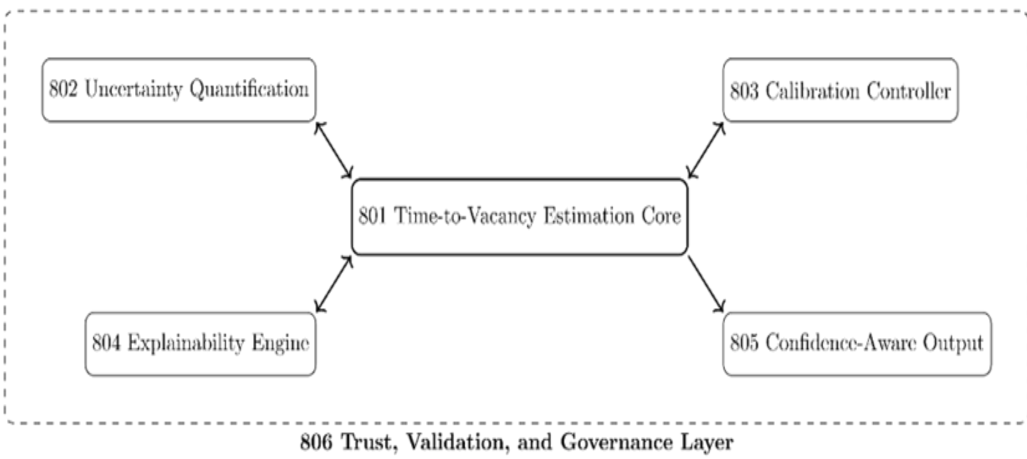
709 Remaining-Time Estimator

By combining the conditioned duration prior with the time elapsed since the start of the occupancy, this module calculates the expected remaining occupancy time for a slot.

710 Time-to-Vacancy Output

The final output at this level of granularity is a local parking space prediction of the time left before vacancy, which can be shown to users or be available for downstream decision systems.

Figure 8: Confidence and Explainability Control Subsystem



801 Time-to-Vacancy Estimation Core

This element is the central decision-making unit that figures out the time left at a slot of parking based on occupancy history obtained from the vision and other contextual inputs. It gives the initial duration estimates that are later regulated by the control subsystems.

802 Uncertainty Quantification Module

The role of this module is to gauge the uncertainty of the time-to-vacancy point prediction. It gets uncertainty sources from model variability, historical error patterns, or probabilistic estimators and then sends these signals back to the estimation core for the performance to be changed accordingly.

803 Calibration Controller

The calibration controller is always on the lookout for the alignment between predicted durations and the observed ground truth outcomes. To maintain statistically reliable coverage over time, it modifies confidence scaling and interval bounds quite constantly thus ensuring that the reported estimates are still legitimate under the changing conditions.

804 Explainability Engine

The component through which understandable explanations for each prediction are generated by going through the contributing temporal, contextual, and behavioral factors. The explanations provided by the Explainability component help the operator, users, and regulators.

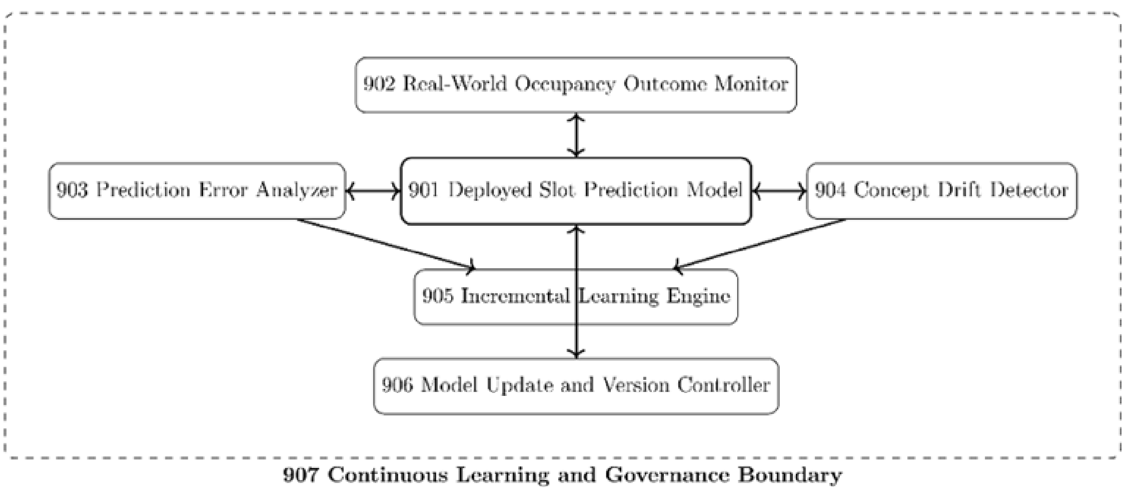
805 Confidence-Aware Output Interface

The final prediction results formatted by this interface for external consumption together with calibrated confidence indicators and explanatory summaries are the remaining-time estimates for the display through user interfaces or programmatic APIs.

806 Trust, Validation, and Governance Layer

This boundary is the estimation core layer and all control modules altogether that are held accountable for the system-wide validation policies, confidence thresholds, and governance rules. The layer ensures the consistent functioning of the estimation, calibration, uncertainty, and explainability functions as well as upholding operational integrity.

Figure 9: Learning Loop and Continual Adaptation Architecture



901 Deployed Slot Prediction Model

A model that has been deployed uses real-time, per-slot inference from locally available, edge, or low-latency environments by employing camera-derived historical data on occupancy and other contextual input variables. While this model is running, it yields occupancy state predictions as well as remaining time estimates.

902 Real-World Occupancy Outcome Monitor

The main function of the system or device is to constantly keep track of the actual happenings of the parking spaces (the real-world occupancy) after each prediction, that is, the true vacancy time, the occupancy duration, and the state transitions. Subsequently, the recorded results represent the ground truth feedback that is used for the assessment of the model in real-life conditions.

903 Prediction Error Analyzer

The prediction error analyzer identifies the differences between the predicted and the actually observed data and calculates the residuals, bias, and systematic deviation pattern accordingly. Furthermore, the module pinpoints the decrease in the accuracy of the residual time at the slot or at the group level.

904 Concept Drift Detector

The drift detector checks for the changes in the statistical input sources, for instance, the occupancy behavior or the environment, such as the change of the time of day, and seasonal usage, changes in camera viewpoint, and user behavior that is continuously evolving etc. The drift detection mechanism is completely independent of explicit prediction errors and, therefore, it can identify silent model decay.

905 Incremental Learning Engine

The incremental learning engine through its selection process utilizes new data, error signals, and detected drift indicators to retrain or adapt model parameters only if the data have been validated. The engine adapts to new patterns by only partially retraining since learning is restricted so as to not forget the previously learned knowledge thereby it can continually improve without a full retraining.

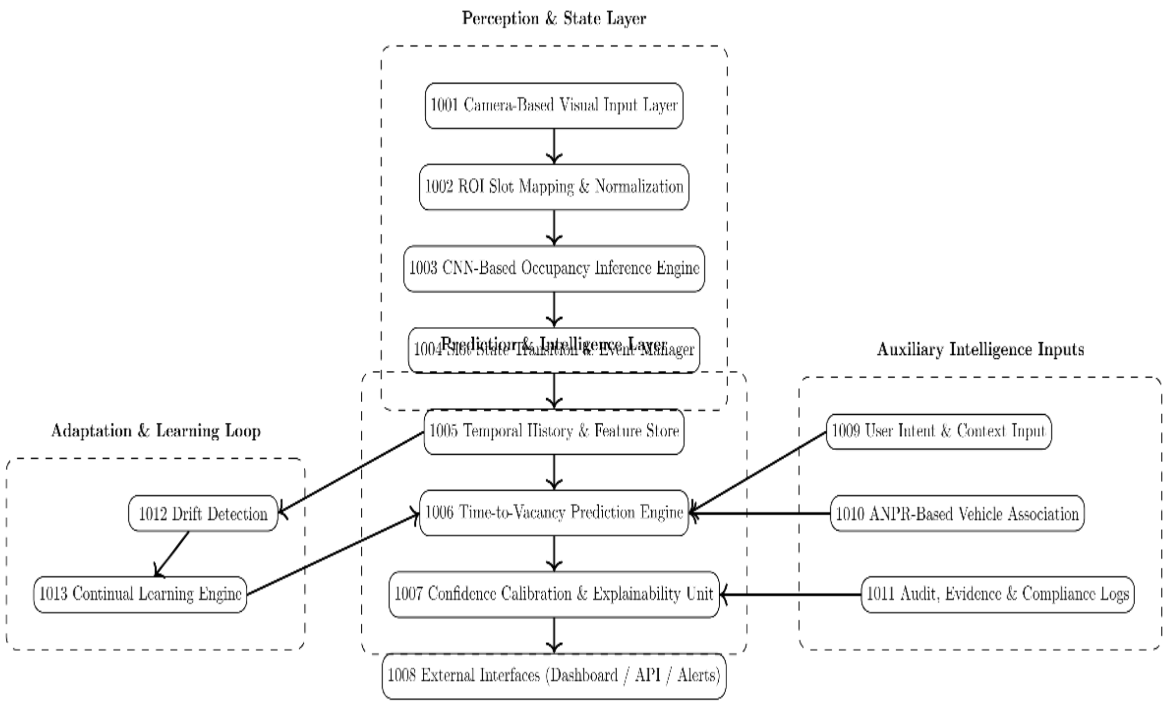
906 Model Update and Version Controller

This unit controls the implementation of revised model versions thus it is in charge of rollback, version tracking, and controlled activation. By the use of confidence and validation thresholds, changes might be local to a particular camera, a zone, or the whole system.

907 Continuous Learning and Governance Boundary

The governance boundary provides a framework for the operational constraints, auditability, and safety measures that cover the whole learning loop. It guarantees that adaptation, updating, and inference are conducted in accordance with the set accuracy targets, stability requirements, and deployment policies.

Figure 10: End-to-End Vision-Based Parking Intelligence System



1001 – Camera-Based Visual Input Layer

The visual data with one or more fixed imaging devices are continuously captured from a parking area. The camera input serves as the main sensing modality for the system, which does not depend on embedded ground sensors, thus allowing for a cheap and easily scalable deployment.

1002 – ROI Slot Mapping and Normalization

The system processes the captured frames to locate and normalize the predefined regions of interest corresponding to individual parking slots. Geometric consistency, lighting normalization, and slot-level isolation for the further inference process are the things this module guarantees.

1003 – CNN-Based Occupancy Inference Engine

The occupancy state of the slot is the output of the evaluation of each normalized slot image via a trained convolutional neural network. Along with the discrete state, the confidence score is also given.

1004 – Slot State Transition and Event Manager

The company tracks occupancy outputs over time for the purpose of detecting state changes between occupied and empty ones. The finite-state transition model is used to generate entry and exit events, thus ensuring temporal stability and noise suppression.

1005 – Temporal History and Feature Store

Occupancy timelines on a slot level are the main idea that the company keeps in mind while the latter layer is implemented. The feature layer learns different kinds of features that include dwell duration, frequency patterns, time-of-day behavior, and slot-specific usage statistics from the usage data collected.

1006 – Time-to-Vacancy Prediction Engine

The parking system using past temporal features and current slot states can tell how much time will pass before the slot that is occupied now will be free. The predictions are made for each particular parking space, not for a whole zone or a group's level.

1007 – Confidence Calibration and Explainability Unit

The unit that also calibrates confidence measures accompanies predicted remaining time values. This department also generates the easily digestible explanations that show which temporal, contextual, or behavioral factors contributed to each prediction.

1008 – External Interfaces (Dashboard, API, Alerts)

Prediction results, confidence indicators, and slot status are made available to the users through the programmatic APIs and the user interfaces. The system is capable of supporting the real-time visualization, alerts, and third-party integration.

1009 – User Intent and Context Input

Contextual explicit inputs such as user-parking intent or user purpose are the factors that are being considered in the prediction pipeline. These inputs help in estimating the time-to-vacancy by either selecting or weighting the learned behavioral cohorts.

1010 – ANPR-Based Vehicle Association

In order to associate vehicles with specific slots and durations, license plate recognition is applied as an optional method. This association is the source of auditability, accountability, and post-event analysis without the need for manual video reviewing.

1011 – Audit, Evidence, and Compliance Logs

All the events related to the occupancy, as well as the predictions and vehicle associations, have been logged in a secure manner to be later used as verified records for operational review, dispute resolution, or regulatory compliance.

1012 – Drift Detection

The prediction error and data distribution shifts are being monitored by the company all the time. The system is supposed to be able to recognize environmental drift or behavioral drift such as lighting changes and seasonal usage variations.

1013 – Continual Learning Engine

Once drifting or accumulating new data causing the system to change, the system changes its model parameters by incremental or periodical retraining, thus keeping the prediction accuracy and confidence calibration intact over time.