MASTER's THESIS

Inference Error Reduction for Parking Occupancy System via Batch-norm Statistics and Confidence Boosting

배치 정규화 통계 및 신뢰도 부스팅을 이용한 주차 점유 감지 시스템의 추론 오류 감소

BY

DUONG TUNG LAM

FEBRUARY 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

MASTER's THESIS

Inference Error Reduction for Parking Occupancy System via Batch-norm Statistics and Confidence Boosting

배치 정규화 통계 및 신뢰도 부스팅을 이용한 주차 점유 감지 시스템의 추론 오류 감소

BY

DUONG TUNG LAM

FEBRUARY 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Inference Error Reduction for Parking Occupancy System via Batch-norm Statistics and Confidence Boosting

배치 정규화 통계 및 신뢰도 부스팅을 이용한 주차 점유 감지 시스템의 추론 오류 감소

지도교수 차 상 균

이 논문을 공학석사 학위논문으로 제출함

2022 년 12 월

서울대학교 대학원

전기 컴퓨터 공학부

덩퉁람

2023 년 1 월

위 원 장	WEN-SYAN LI 교수
부위원장	차상균 교수
위 원	김영민 교수

Abstract

Although a smart camera parking system concept has existed for decades, a few approaches have fully addressed the system's scalability and reliability. Because the cornerstone of a smart parking system is the ability to detect occupancy, most current systems have used sensors buried under parking spots for this task. However, this is extremely costly when expanding the solution to a large-scale capacity as the price will go up with the number of parking places. Moreover, as CCTV has been installed in various parking sectors nowadays, it would be beneficial to exploit occupancy detection through the computer vision approach. Nevertheless, traditional methods use the classification backbone to predict spots from a manually labeled grid. This process is time-consuming and loses the system's scalability in production. In addition, when considering Deep Learning approaches, solutions will only partially generalize for some situations and can potentially cause numerous errors during inference, which massively reduces the benefits of using computer vision in a smart camera parking system. These drawbacks demand a fast and low inference error smart camera parking system which is the scope of this thesis. In this thesis, the system boosts the inferencing time by replacing traditional classification methods with a CNN detector called OcpDet and operates the detection at edge devices for a scalable and load balancing structure. Therefore, the OcpDet backbone is powered by Mobilenet for fastest and lightweight inference, which is friendly for edge devices. Information from the training error module and the spatial estimation module were injected into the model to tolerate the false detections that can occur. The training information is extracted from the OcpDet's Batch-norm

statistics to tell whether the captured scene matches the training knowledge. If the scene is out of domain knowledge, its capture is collected for model improvement through active learning iterations and conducting further inspections. Meanwhile, the spatial knowledge averts false detections and suppresses wrong-located bounding boxes during inference through a confidence boosting technique. Based on the enhanced results, it can be treated as a spatial error for a scene and combined with its training error to decide whether to skip it in the outcome. In the experiment, the system was benchmarked on the existing PKLot dataset and reached competitive results compared to slow classification solutions. To measure the scalability and reliability of the system, an additional SNU-SPS dataset was created, in which the system performance is challenged from various views and conduct system evaluation in parking assignment tasks. For these tasks, a simulation from multi-edge cameras of different parking lots surrounding the SNU campus was conducted, and gathered the parking detection information through a message-broker protocol. The result from the SNU-SPS dataset shows that the thesis's approach is promising for a real-world application with a small error trade-off.

Keywords: Smart Parking System, Occupancy Detection, Inference Error, Edge Device

Student Number: 2020-25413

Contents

List of Figures

List of Tables

Chapter 1

Introduction

According to the 2018 UN media, 68% of the world's population will move to urban areas by 2050[28]. This dense population in towns and cities directly leads to an increase in the number of cars and other vehicles, which raised a major concern about parking management on its efficiency. Letting drivers wander in the city to find an appropriate parking slot in a tight city space causes significant air pollution and wastes drivers' time and energy. It also leaves empty spaces in parking lots and varies statistical measurements on parking occupancy rate, which trouble operators to exploit their facility for revenue. In addition, these factors may worsen, particularly during peak hours when the flow density is at its maximum. For real concrete evidence, a recent report by INRIX [14] shows that on average, a typical American driver spends 17 hours a year looking for a parking space, which can go up to 107 hours when addressing a dense population city like New York. From [32] analysis, the exceeding of CO2 emissions can rise nearly three times due to this problem. Therefore, a stable future city needs a Smart Parking System (SPS) that can link drivers and parking operators

and benefit both sides. By suggesting optimal parking places to drivers and managing their destination, a future SPS not only minimizes vehicle emissions (via decremented delays in finding the vacant parking spot [1]), but also provides operators a reliable number of customers to boost revenues through e.g. dynamic pricing [33].

Regardless of potential promises, most SPS functionalities are strictly bounded by the performance of correctly determining the occupancy of a parking lot. Hence, the current parking system relies heavily on sensors as the first layer of the system [1]. However, despite its high precision, this turns out to be expensive when scaling up the parking lot size for future perspective, as each sensor (magnetometer/ultrasonic sensor)[33] is designed to operate solely on a single parking spot. An effective solution for this drawback is applying computer vision (CV) to occupancy detection. A single camera can cover multiple parking locations and eliminate the need for a sensor per parking spot[2]. Furthermore, because most parking lots nowadays have security cameras, it reduces installation and maintenance costs and supports multiple additional tasks for better parking management, such as wrong parking placement, abnormal behavior, and theft detection [39] with which sensors fail to cope.

Although computer vision is promising, no datasets exist for a full CV-SPS intention. Most popular datasets PKLot[15], and CNRPark-Ext[3] and their solutions [39, 4, 38, 30, 3] are constrained to a small number of parking lots and treat each parking spot as a binary classification image. Regarding performance, there are three main drawbacks to this type of dataset and their following research. First, it limits solutions to operate in the classification scheme solely. Second, when the number of slots increases, reliable deep-learning classification solutions [38, 30] require multiple forward passes and need to be faster to run in real-time feedback to drivers or to stack more additional tasks. Lastly,

a parking operator using solutions from this dataset must reannotation every parking spot for new installation. For example, an operator is in charge of three parking lots with at least 300 parking spots in each facility. He must perform 900 annotations to use the solution, and this procedure will be conducted again when the positions of the cameras change. Therefore, an automatic parking space localization and classification solution for a scalable CV-SPS is needed to deal with the future urban population. Moreover, when considering the functionalities of SPS in these datasets, there is no information on the parking location or surrounding traffic in this dataset as they only focus on occupancy results. It creates a big gap between existing data and the SPS's scope in the CV paradigm.

Aware of those flaws in current CV approaches and the importance of CV in future SPS, the thesis poses a complete CV-SPS with a new SPS-based dataset called SNU-SPS. For the CV-SPS, it is separated into four layers and demonstrated in Fig 1.1 and extended to distributed edge services to provide a flexible and scalable architecture.

The first layer is the occupancy detection layer which aims to capture parking spot statuses at the edge level by a deep learning object-detector called OcpDet. Changing the scope to object detection instead of classification and providing results at the edge services lifts the system's performance to real-time operation, produces results frame by frame and avoid delay when solving all inferences at the central server. To serve for the edge devices which requires light weight model, the OcpDet inherits the Mobilenet [35] as the backbone. However, most modern object detectors are not error-free and can be potentially wrong in a real-world inference due to the variant of the environment (i.e the different views of CCTV installation, the color of the parking spot, the brightness and the contrast of the observations throughout the day, etc). Hence, to maintain

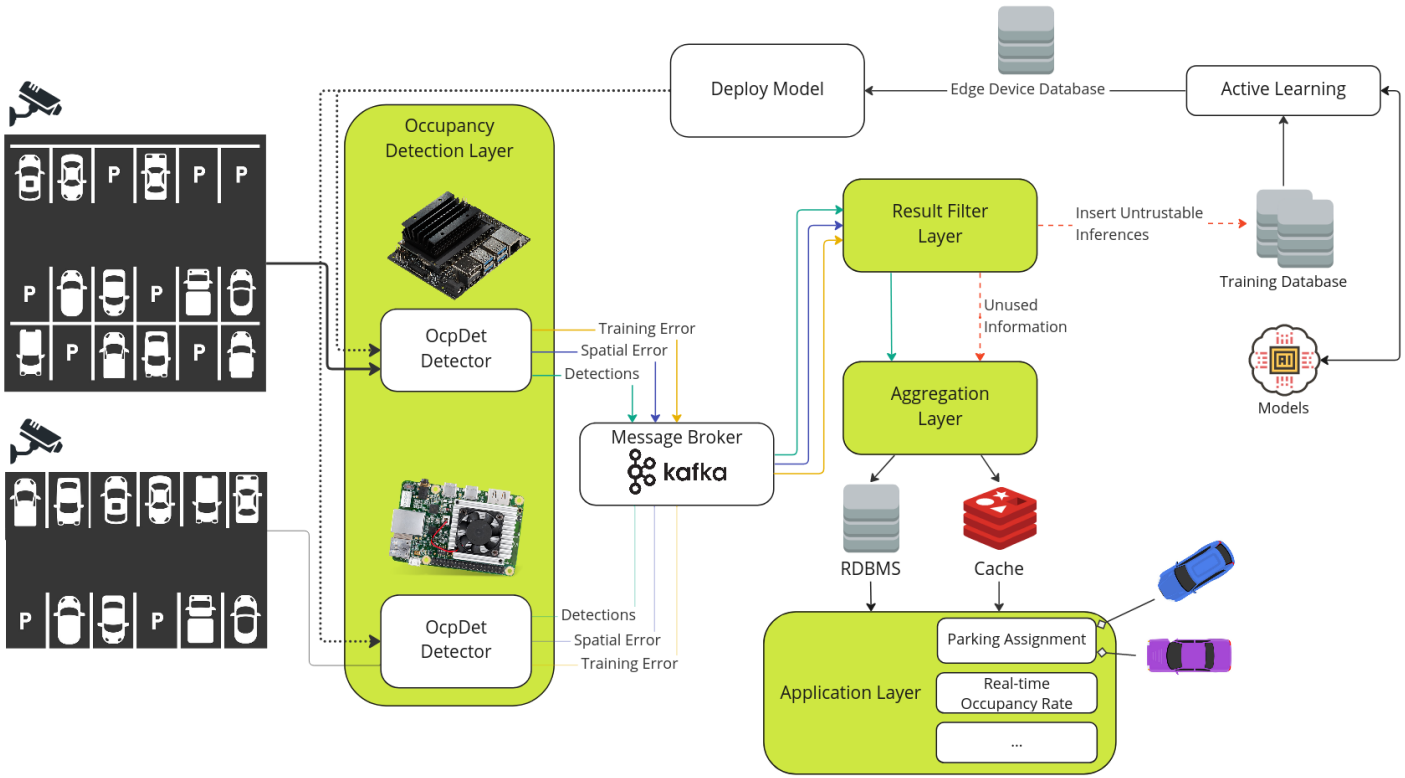

Figure 1.1 CV-SPS overall architecture: an occupancy detection Layer, a result filter layer, an aggregation layer and an application Layer

a reliable SPS, a result filter follows by the detection layer as the second layer. A wrong detection or inference error result can be relayed on two main factors: spatial error or incorrect determination in the visual observation and an out-of-domain capture input. Under this awareness, the filter layer addresses two additional results from two modules of OcpDet: the spatial estimator module and the training error module, to create a complete contradiction for the results in both visual information and training knowledge perspective.

The training error module catches the training difference of an inference frame by injecting domain self-awareness without substantial delay for model interpretation. Instead of creating disagreement from multi-model inference or multi-forward passes, which causes a massive delay in the system judgment, the module emphasizes the likelihood of a sample falling into the training and target domains. As Batch Normalization layers [21] appear in every current SOTA model, the training error module takes them as reference statistics for training data [10, 11] to extract domain samples. Hence, the model can continuously measure the difference between observed data samples and its previously extracted domain information, which assures the model performance interpretation is stable in the training knowledge. In the meantime, the spatial error can only be replicated by a reference groundtruth, which is unavailable during inferencing. During inspectation, the spatial error can be formalized as low confidence activation in final detections. Therefore, the spatial estimator module aims to create the groundtruth or the precise detections by averting false detections by enhancing low confidence activation and suppress the incorrect high confidence. This procedure is considered as a confidence boosting method. As most modern detectors rely on anchor boxes for detection, the module is created to predict a separated head of active anchors in the scene as a supplement measurement to boost and suppress the main detection results. Based on the

final consequence change in predictions, the spatial error module analysis it as the spatial error during interpretation.

These errors are then aggregated as a single score. Inferences with high error accumulation will be marked as unusable information and collected for fine-tuning and retraining the detection model in an active learning manner. As the system emphasizes on improving model through active learning. The new and untrusted samples are ranked in the most impact sample to the model improvement knowledge. These samples are then added to the training pipeline for improving the OcpDet. In the meantime, only correct/believable detection results are stored and analyzed in the third layer as the aggregation layer and transparent to operators and drivers. From this layer, the last layer can support applications such as optimal routing for drivers or alert operators about upcoming occupied parking slots. SNU-SPS dataset is created to support this idea of the system. It contains parking slots captured from multiple parking lots at various angles, ranges, and positions with different light and contrast settings to train the detectors. Furthermore, it is also attached with parking lot GPS and surrounding traffic information for system performance analysis.

The proposed system is extensively tested on the SNU-SPS dataset for efficiency evaluation and conduct detection measurements with the popular parking datasets PKLot[15] for a detailed benchmark. The results from the experiment raise a competitive performance compared to exhaustive classification methods and promise a small error trade-off for applications. In addition, to prove the robustness of the capturing the inference error for model improvement in active learning manner, additional experiment is tested extensively on the SNU-SPS dataset for improving model accuracy and hinted a potential of this approach when the data expansion is provided.

Chapter 2

Related Work

2.1 Deep Learning Automatic Parking Occupancy Detection

Because most previous work focuses on solving the Smart Parking System (SPS) occupancy task as image classification from datasets [15, 3] with manual label mask/grids location of a parking lot, none of the available datasets could be found for automatic parking space detection. It leads to a small amount of effort on this topic. It was found that there are currently two main approaches to this topic: a mask-based method and a detector-based method.

A mask-based method aims to provide parking patches directly from captures and perform binary classification for the occupancy. The perspective transformation method [7, 9, 29] is usually used in this scheme to bring the parking lot to a 2D grid presentation. Therefore, it can save time for self-annotating parking locations and exploit the classification machine-learning and deep-learning backbones. However, since the perspective projection process is highly depen-

dent on the camera setting to the parking lot, classification models need to be retrained for different camera settings, which questions the scalability of those methods. Notice this behavior, [26] has introduced a GAN approach that generates the parking place's masks from a team of drones, but there is no comprehensive measurement of the correctness of these masks. In addition, this method requires a top-view capture of the lot, making it unrealistic for indoor parking facilities.

In contrast to mask-based solutions, detector-based approaches perform detection and classification tasks in a single process by a CNN architecture instead of separating them into two processes, which maintains flexibility and fast inference for CV-SPS infrastructure. The CNN architecture in this realm regresses a parking slot as a foreground or a region of interest and optimizes its classification score. This procedure can be classified into two-stage and one-stage detectors. While the two-stage detector, such as Faster RCNN [34] focuses on the first stage to propose the regions of interest and performs classification on those regions in the second stage, the one-stage detector combines both tasks by grid-anchor regressions. However, because an empty or small parking slot is easily confused as a part of the image's background, both of these architectures face a lot of flaws [31]. Most recent works [24] only used detectors to find a parked car in the parking lot and determine the occupancy rate by a preowned parking lot's capacity and location. This approach relaxes the problems into the well-known car detection, but it limits the extension of the SPS for letting drivers know the location of the parking slot. Recent developments in new architectures such as YOLO [6], and RetinaNet [27] have opened some flexibilities in the small object capture. The idea of using a drone's captures is also used by [20]. The author performs the car detection at the top-view by Faster RCNN and YOLO and combines it with the layout proposal. This

method faces the same drawback as [26] and is restricted for car detections. For a complete occupancy detection from a detector, only [31] has been conducted on a RetinaNet on PKLot [15] dataset. However, the results show much confusion between moving cars and occupied parking slots. While the main reason for this inefficiency is the nature of the PKLot dataset itself (partial area of the parking lot is annotated), the method's performance can be improved if there is an attention mechanism on the parking lot region. In addition, there is a potential non-optimized model design as there is no information on the grid-anchor feature selection provided. In constrast to this, the OcpDet is well-built with spatial error awareness, which emphasize the important of capturing parked vehicles and blank spots.

2.2 Deep Model Uncertainty

As a model performance is a reflection of the coverage of the training dataset, recent research tends to capture the model error or inference error in the wild by measuring its uncertainty/stability or contradiction. This type of model error capture has usually been integrated with the active learning method, which helps to select the most contributed samples from an additional growing set to improve the model overall performance. However, most works focused on image classification [5, 12], and little attention has been drawn to the general task-agnostic such as object detection. Extending the scope for smart parking system, a model determination should be constrained to a single-forward pass, which limits the available work. It was found that recent work on catching deep model uncertainty divided into two main branches: training-domain base and non-training-domain base.

2.2.1 Non-training domain base

The methods aim to produce a score for an inference sample for selection. The non-training domain splits into single-forward and multi-forwards to determine this score.

For single-forward approaches, [8, 40] proposed a simple solution by combining the marginal score of classification scores in one image and can be treated as a pixel scores aggregation in instance segmentation. Despite this simple implementation, according to Gal [18], a model can be uncertain in its predictions even with a high softmax output, which means using the output interpretation is not a solid factor in determining model uncertainty. Instead of relying on the margin or the entropy of the output prediction like previous approaches, [41] tackled the problem by estimating the model stability from its training behavior. This estimation comes from additional regression layers that capture a training pair difference. Even so, it requires a model to increase the number of parameters and may affect the model convergence during training.

Different from the single-forward methods, multi-forward solutions aim to create a contrast in the inference. [22, 37] represent the model uncertainty as *contextual disparity* of the image. An image's *contextual disparity* is extracted by the stability of a category prediction from an image and its similar version (a next frame [37], or its augmentation [22]). However, these approaches increase the computational time twice as each score result can only be obtained after two inferences. Meanwhile, [19, 5, 12, 17] share the same path of scoring unlabeled images by gathering inferences from an ensemble of N neural networks. This procedure leads to N times slower inference in the worst case and causes a huge computation burden. Especially, when solving this at the edge device, it causes a burden on computation due to the hardware limitation.

2.2.2 Training domain base

Solutions from this branch compare the new inference samples with the training information representation, which allows them to execute in a single forward manner. From this concept, Sener [36] follows a distribution manner to select new image samples through their intermediate feature distance in terms of Core-Set representation. The intuition is learning over a representative subset from the whole pool samples. Following this idea, [37] replaces the feature distance with the *contextual disparity*. However, both methods are strictly limited by the number of training samples. Each new addressing image sample has to be compared with M training feature information, which is heavily computational when M reaches millions of samples. Furthermore, the determination of domain difference from both methods is not robust. It selects an inference sample by the error difference to a training sample instead of an inference sample to the whole training dataset knowledge. Choi [13] approached the problem differently by replacing the original localization and classification output heads of a detector with their density estimation to evaluate the uncertainty of the results, which is implicitly a statistical comparison module to the training domain. However, to implement this method on edge is difficult due to a whole structure modification and can not exploit the existed edge inference framework.

Compared with these methods, the system training error module can be categorized as another training-domain base approach as it replicates the comparison through Batch-norm information. By extracting the training domain from Batch Normalization layers and using it as a reference statistic for new data sample selection, none of the architecture modifications like [13] or increasing model parameters [41]. Moreover, it preserves the single forward data selection without multi-forward passes, or ensembles, significantly reducing the

computation, which is tremendously important for the edge device inference.

Chapter 3

SNU-SPS Dataset Description

Figure 3.1 SNU-SPS dataset images representation from various indoor and outdoor views. Annotation colors (Red: Occupied, Green: Available, Blue: Restricted and Yellow: Illegal)

The SNU-SPS dataset contains nearly 3500 images to support the full scope of CV-SPS. Those images are captured from various views, heights (1-3m), and

Set Type	Total Images	Total Labels	Classes			
			Available	Occupied	Illegal	Restricted
Train	2848	18263	7229	10596	396	42
Test	574	2747	1291	1336	36	84

Table 3.1 Training and Testing Sets

light conditions in indoor and outdoor parking lots at a full HD resolution. Each parking lot has different spot background colors, light conditions, and capture alignments. The total images were manually checked, labeled, and attached by GPS to the corresponding parking slot.

3.1 Image Acquisition

All images are captured with a full-HD resolution. For the training set, it is captured randomly for one month in 15 parking lots. Meanwhile, the test set is captured consecutively in 6 parking lots from 3-6 pm through 5 working days. Those parking lots have been selected randomly and have different sizes and scales as well as background parking spots' colors. It should also be noted that none of the six parking lots are in the training set. Moreover, test samples contain various weather conditions (sun/rain/cloudy) and have corresponding surrounding traffic measurements from the open Korean government website *http://www.utic.go.kr*. This is to serve for further future SPS tasks and benchmarks.

3.2 Labeling

For each parking sector, parking spots were labeled as *available/ occupied/ illegal/ restricted* and hid vehicle license plates for privacy concerns. Each annotation is covered by four key points that specify the localization of a parking lot. The wrapping bounding boxes were formulated from these key points. Especially, optional image masks for the test set are provided to filter out overlapping areas and non-important localization among capture among parking lots. The intention is to maintain the system's constraints and preserve a better parking assignment benchmark.

Chapter 4

Occupancy Parking System

As shown in Fig 1.1, a parking lot is divided into sectors to create a scalable and efficient CV-SPS. Each sector is controlled and well-observed by a camera and has non-overlapped observing areas among cameras. This constraint reduces the complexity of the problem by duplicating observation or high occlusion. Assuming the parking lot can be set up with this requirement, the overall system architecture consists of four layers: an occupancy detection layer, a result filter layer, an aggregation layer, and an application layer. The collection layer is responsible for gathering the detection results from distributed cameras in the parking lot as well as their potential error info during inference. Then, these results are propagated to the filter layer to cleanse for reliable results. Non-trusted results are masked out as non-usable spaces. This filtered information is stored in the aggregation layer that acts as the system's middleware. From this layer, the application layer can receive reliable SPS support. Users can have a transparent measurement of the current occupancy capacity of a parking lot, while model engineers can access and inspect poor performance behavior

in specific sectors. Especially, optimal routings and parking assignments can execute with high precision by correctly capturing vacant spots.

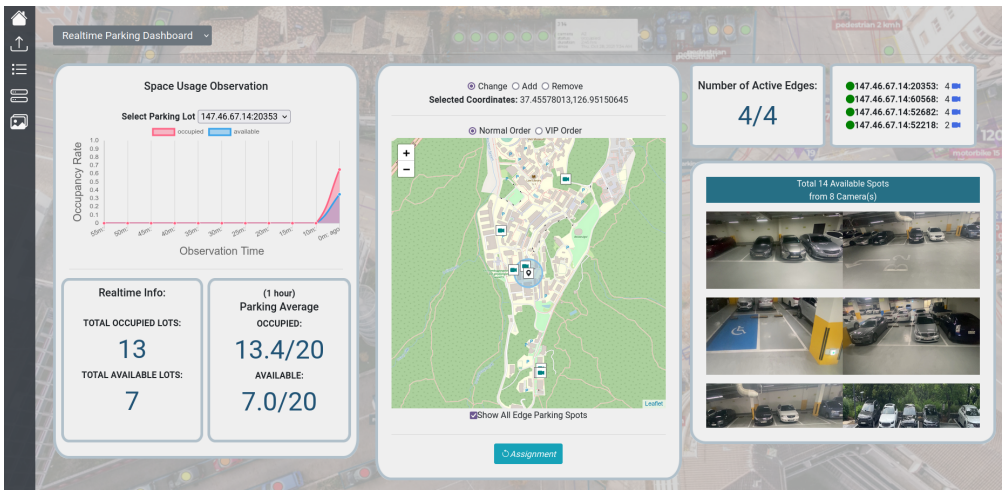

Figure 4.1 Smart Parking Management Dashboard Website Visualization

4.1 Occupancy Detection Layer

When addressing parking occupancy as an object detector, the most arousing problem is the confusion of the parking slots with the image background information, such as moving cars or blank spots. The only meaningful visual information is the thin lines separating spots. However, it is usually missed at the lower level of deep neural networks. Thus, RetinaNet[27] is a promised solution due to its feature pyramid network (FPN). In short, the FPN backbone combines standard convolutional network lower features with lateral connections of early-level features. This simple characteristic allows the network to construct rich, multi-scale object features, which maintain the impact of the line features in the network. However, from [31] results, despite capturing good center localization, traditional RetinaNet could not expand the parking space

tightly when the mAP dropped dramatically from 63.64 to 4.75 when raising from 0.5 to 0.75 precision. This behavior can be caused by non-optimized anchor grid features and a lack of a location attention mechanism. Moreover, the Resnet backbone is quite heavy for computation and may not be able to scale up with other additional SPS tasks, limiting the scope of CV-SPS.

$$L_{size} = \sum_{i=0}^{N} D^i_{p,k} / D^i_{p,c} \tag{4.1}$$

Noticing this limitation, the heavy Resnet backbone is replaced with a lightweight Mobilenet[35] backbone for faster inference and to build up a new model from this architecture called OcpDet. To make the model focus more on the line attribution information, instead of solely detecting the bounding boxes by their centers and sizes, OcpDet was trained with four additional points as the four key points described in the SNU-SPS dataset. Therefore, OcpDet also predicts the key points of parking slots in the localization head. A new loss function L_{size}, which aims to maximize the predicted coverage boxes to their corresponding predicted key points, has been designed from this scope. These N key points are treated as anchors which pull the box corners closer to them. In equation 4.1, the distance between a box corner p to its corresponding keypoint k is denoted as $D_{p,k}$ and the distance between a box corner p to the center c is denoted as $D_{p,c}$. However, as the goal is detecting the localization of the parking spots, the loss function L_{size} is treated as a regularizer during the training stage, which makes the key points regression omitted in the inference phase.

4.2 Result Filter

The scope of the system is not only focusing on providing detection results of the parking spots but also estimating the error of its interpretation in terms of

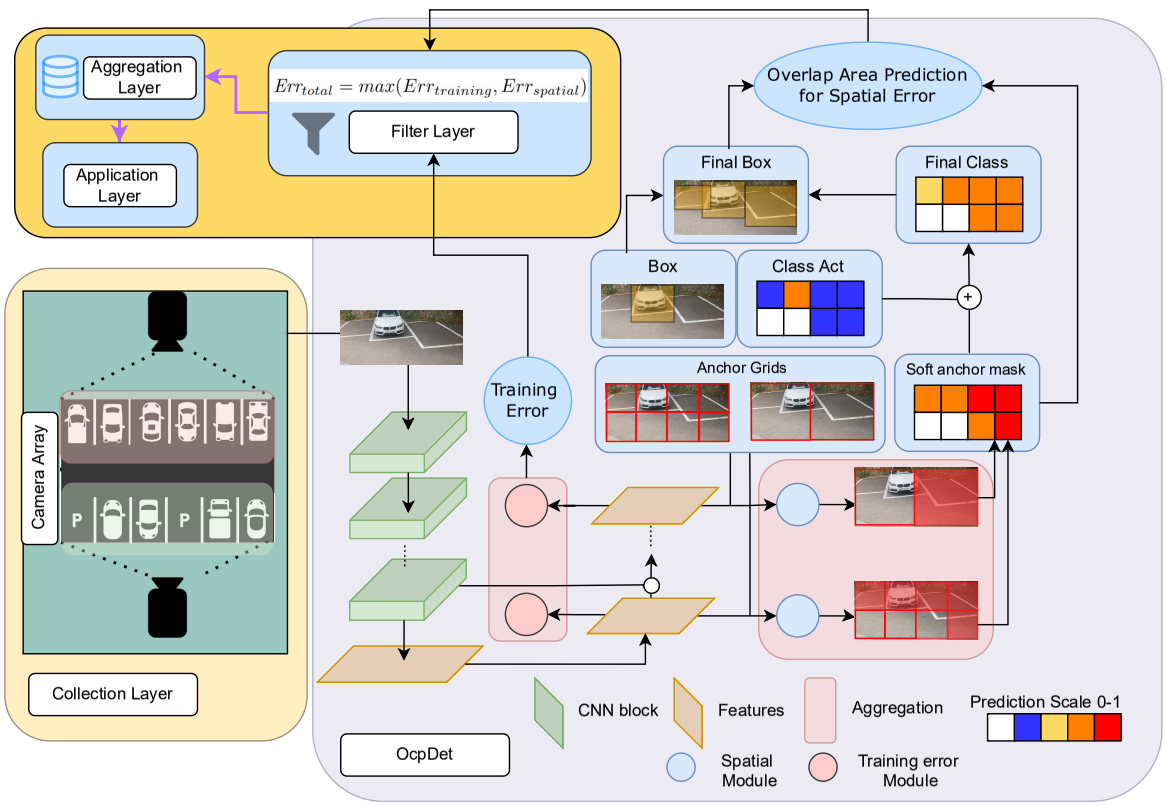

Figure 4.2 Overall Result Filter Approach from OcpDet Modules: the training error module (red) and the spatial estimator module (blue)

spatial information and training knowledge awareness. Hence, these tasks are responsible by two corresponding modules: the spatial estimator module and the training error module, which have been injected into OcpDet as additional features as shown in Fig 4.2.

4.2.1 Spatial Estimator Module

Because OcpDet is a deep learning object detector, it uses anchor boxes to provide detection results. Anchor boxes are formed by dividing an image into patches and represent as different scale levels for classification prediction and localization regression. From these predesigned anchors, the parking spots can be determined by their corresponding activation score and filter out low-confident locations by a threshold. Because of this mechanism, well-determined locations may be missed out, which leads to incorrect visual information. However, to capture this error without human supervision is difficult as there is no existed groundtruth during inferencing.

The spatial estimator module overcomes this problem by providing additional heads that predict anchors that can be active in the scene. As the parking lot layout is usually aligned, equally separated, and non-overlapped among spots, it is very convenient to wrap a parking slot in one single anchor patch representation (i.e non overlapping object will occur). Based on this characteristic, the spatial module can create an anchor activation mask from each level feature generator of FPN as demonstrated in Fig 4.3 and rely on this representation to enhance or suppress the confidence of anchors i.e confidence boosting. Through this confidence boosting, the final outcome can be treated as the groundtruth of the inference and provide spatial error estimation.

In detail, a residual convolution block is attached for each of the N feature levels. The last channel is averaged to get a 2D map of anchor patches to avoid

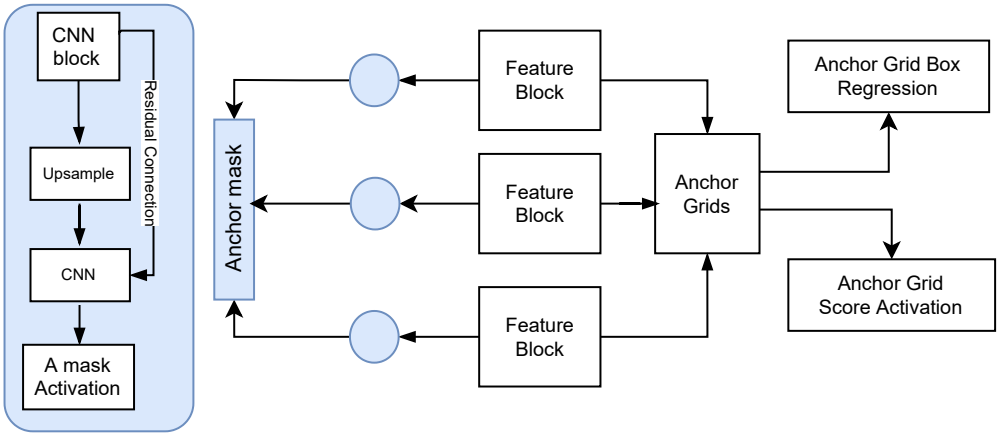

Figure 4.3 The Module Architecture for the Spatial Estimatior Module

insufficient computations. A flattening map, such as the fully connected layer, can reach huge connections for early-level features. For example, a 256x256 anchor can cost 65k connection and additional 65 parameters for a dense layer. With that intuition, as parking slots are in the foreground, this 2D prediction is triggered by a sigmoid activation and is treated as a binary classification (Fig 4.3). As demonstrated in Fig 4.2, from this activation map, an anchor mask, which acts as a confidence boosting, is created for parking slot locations as a reference for the model spatial outputs. To train this activation map, each map M from N feature levels is compared with its corresponding classification target map C in the classification head by a ϵ difference as a loss function. Therefore, this loss can act as a regularizer for the model to provide attention to the foreground region of the model. Moreover, as the anchor mask head is not directly connected to the localization head prediction, it gives the model another degree of freedom to operate while implicitly improving the localization through top-level features.

4.2.2 Training Error Module

The key novelty of the training error module is extracting and combining the training knowledge of the neural network into the model inference phase. Thus, more factual information can be relied on than just model output uncertainties (bounding boxes, classification scores, etc). Thus, the system aims to extract this knowledge through Batch Normalization layers [21], which is famous in architecture nowadays for reducing the covariance error during training.

A model Φ is assumed to consist Batch Normalization layers [21]. An input x is generally transformed into x' by a layer f, such as a convolutional layer or a fully-connected layer. The transformed information x' is then fed into a Batch Normalization layer. This layer analyzes the mean (μ) and the variance (σ^2) of the total N representation of x in a single batch and re-center and re-scale them by these mean and variance i.e normalization. This procedure extends to repeat for lower layers of the model, making each Batch Normalization layer a recording statistic of a fraction of the model knowledge.

The idea of resembling Batch Normalization layers as the training knowledge of the model has been introduced in data distillation [10, 11]. Distillation re-constructs the training dataset by minimizing the difference of each convolutional layer's output with the Batch Normalization statistic from a batch of gaussian images. This procedure forces the transformation x' to regress to the high-population information model received during the training. Therefore, it indicates that input will fall into the training knowledge if the distillation loss is low and quickly converges, while non-familiar information will raise this loss. Our method inherits this motivation, taking the Batch Normalization layers to link information to the training knowledge. However, directly applying distillation in inferencing is ineffective. First, creating distillation loss for every

sample is expensive because each inference must be detached, and the model must be trained to tune a captured input's pixels repeatedly. Second, as the output information at a layer j is not in the corresponding statistics μ_j and σ_j, it does not guarantee its following output transformation at a layer k will not near the μ_k due to some non-linear activations. Therefore, it makes a model mistake information in final task decision layers such as softmax [18]. Lastly, because a model can have hundreds of Batch Normalization layers, addressing every Batch-norm layer is insufficient in terms of memory and computation, as a model must store every previous layer's output for comparison before finalizing the training error outcome.

Hence, to cut the edge with the distillation scope, information should only be obtained from specific/sensitive Batch Normalization layers that can potentially get information out of their statistics. By addressing these layers, untrustable layer transformations are focused and used to conduct a direct statistic comparison and reduce the computational overhead from $\mathcal{O}(M)$ to $\mathcal{O}(M')$ as M' is the sensitive subset of M layers. The process of selecting those Batch Normalization layers is summarized in Algorithm 1 and is demonstrated below.

$$L_{dist} = \sum_{m=1}^{M} \parallel \mu_m - \mu_{\Phi,m} \parallel_2 + \parallel \sigma_m + \sigma_{\Phi,m} \parallel_2 \tag{4.2}$$

At first, the distillation training process of [10] by the loss function Eq 4.2 is mimicked to choose these sensitive candidates by feeding a random gaussian image GI for N epochs. Then, each μ_m and σ_m of each batch norm layer in the last n epochs is stored in N where distillation nearly reaches the plateau. This procedure results in two arrays for mean record BN_{mean} and standard deviation record BN_{std} with the same size $M \times n$. Finally, a z-test formula is applied to find the maximum distribution difference among records of each layer

as their expected sensitivity values and select the top M' layers. It is assumed that insensitive layer distribution should be stable at the end of the distillation. Hence, these sensitive Batch Normalization layers are determined and attached to the training error module.

$$E(M', I) = \max_{m \in M'} \| \frac{\mu_m - \mu_{\Phi,m}}{\sigma_m + \sigma_{\Phi,m}} \|_2 \tag{4.3}$$

Because the statistic size of each Batch-norm layer is different, a training error module is created to estimate the information error during the inference. A z-test function conducts the estimation as a normalization function from the convolutional layers' outputs of each Batch-norm layer. Then, a *max* aggregation is set to compare the final value of each module and represent the maximum value as the image's global error value E as demonstrated in Eq 4.3.

However, from the training perspective, it is difficult for the training process to transform every data sample to match closely the mean and variance of each Batch Normalization, especially among different batches. Therefore, the model normally will bring the transformation x to a proportional difference of these statistics and optimize learning parameters θ from this gap. For example, for a quick benchmark in Fig 4.4(left), none of the transformation convolutional outputs match the corresponding batch norm statistic layer through z-test comparison. Therefore, when the training dataset is available, those training samples are passed through E Eq 4.3 as the training reference instead of using a Batch-norm layer's mean and variance as the reference. As shown in the module design Fig 4.5 below, the global error E is compared with a training reference error for new coming samples to determine the final value error. This comparison is made by normalizing E with the corresponding batch norm error mean μ and its variance σ.

Algorithm 1: Sensitive BN layers selection:

input: epochs N, random gaussian image GI, sensitivity S, number of selected sensitive layers M'

$i \leftarrow 0$;

$n \leftarrow 0$;

while $i \leq N$ **do**

  optimize(L_{dist},GI)

  if *start plateau* **then**

    $n \leftarrow N - i$;

  end

  if $n \neq 0$ **then**

    for $m \leftarrow 1$ **to** M 1 **do**

      $BN_{mean}[m][i-n] \leftarrow \mu_m$;

      $BN_{std}[m][i-n] \leftarrow \sigma_m$;

    end

  end

end

; **while** $m \leq M$ **do**

  $j \leftarrow argmin(BN_{mean}[m])$;

  $S[m] = max(\frac{BN_{mean}[m]-BN_{mean}[m][j]}{BN_{std}[m]^2+BN_{std}[m][j]^2})$;

end

; **return** $rank(S, M')$

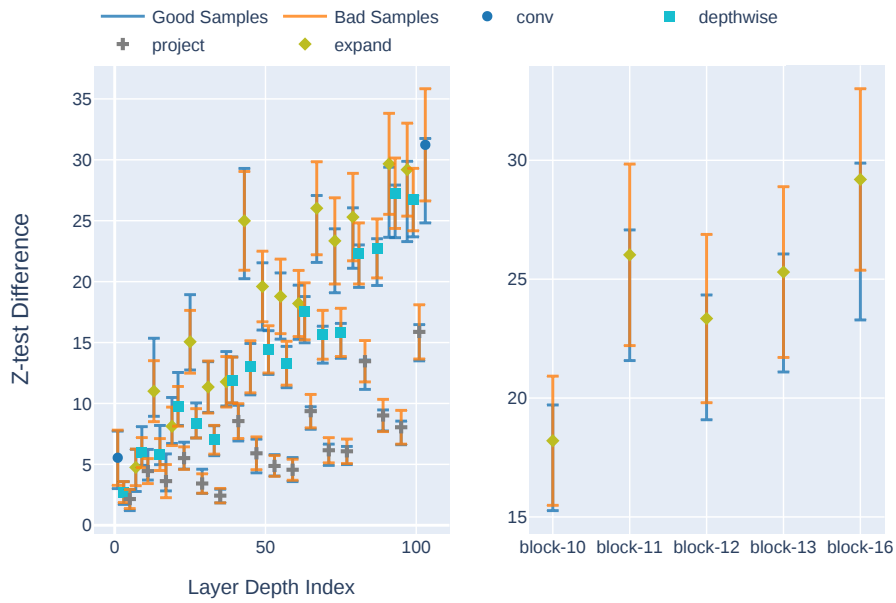

Figure 4.4 Convolutional output in z-test comparison with corresponding Batch-norm layer statistics from Good Samples (mAP > 0.5) and Bad Samples (mAP < 0.1). **Left:** Overall difference with incremental layer index, **Right:** Difference on on selected sensitive layer

4.2.3 Filter Information

After obtaining the detection results with two additional predictions from the spatial and training error modules, the occupancy detections are analyzed and filtered out. To check the correctness in terms of spatial information, the active anchor mask S is passed through a tanh activation, which guides low activation values to below zero and hightlights the rest. Then, instead of immediately applying on the N' bounding boxes that have higher activation confidence than the object detection threshold, the prediction P from the foreground N detec-

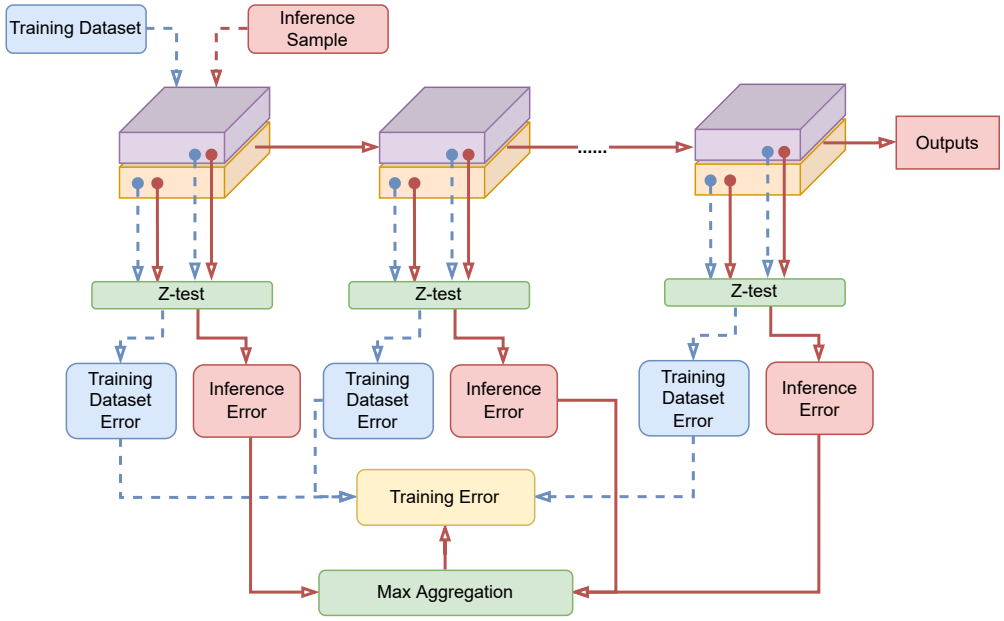

Figure 4.5 The Module Architecture for the Training Error Module

tions are added with this activated information as the final outcome P' (Eq. 4.4).

$$P' = P + tanh(S) \tag{4.4}$$

$$Err_{spatial} = 1 - \frac{1}{M}\sum_{i=0}^{M} 1(\frac{D_i \bigcap R}{D_i}) \quad \text{s.t} \quad 1(\nu) = \begin{cases} 1, & \text{if } \nu > \gamma \\ \nu, & \text{otherwise} \end{cases} \tag{4.5}$$

$$Err_{total} = max(Err_{training}, Err_{spatial}) \tag{4.6}$$

Based on the new bounding boxes having been retrieved, suppressed, and filtered by a traditional object detection threshold, the new strong-belief M candidate boxes are formed as demonstrated in Fig 4.2. They are compared with N' bounding boxes by overlapping. To reduce the complexity $M \times N'$, N'

is treated as a unified region R. This region is compared with D_i area coverage from M bounding boxes. From an array of overlapping ratios, the spatial error $Err_{spatial}$ is estimated for each frame capture detection as demonstrated in Eq. 4.5 by an overlapping threshold γ.

However, because the spatial information can still be affected by uncovered knowledge input, this spatial error is combined with the training error $Err_{training}$ as the outcome of the detection failure estimation. To keep the error regardlessly robust, the maximum aggregation is selected as a final error score Err_{total} for the inference (in Eq. 4.6). From this score, the aggregation can filter out untrustable information during detection, mark them as a failure in the corresponding categories and collect for active learning iterations.

4.3 Aggregation Layer

As the SPS gathers information from multiple devices for application layers, it has to cope with failures in both occupancy determination and concurrency of information from each device (node of the system). While the previous result filter layer has handled the first obstacle, the latter can cause a huge drawback in terms of application as the outcome of each observation can mismatch in time stamp and provide unstable state predictions for parking spots.

To tolerate this problem, the system attaches the timestamp attribute of each observation to its model inference output before transmitting back to the aggregation layer. Due to the network or the crashing situation that could happen at each node, the system has different protocols to handle delay and non-response nodes. At first, a timeout counter is set to wait for all devices outcome messages. During this process, each timestamp is normalized to 'seconds'. Outputs containing the same timestamp will be grouped as the final observation

of the system. In each group, multiple outcomes from the same camera will be accumulated as one final result (i.e only overlapped information among these results will be kept). This approach reduces unstable predictions during the run of a device and provides consistent information. When the timeout counter reaches its limit, any group missing a device's output information can borrow its closest neighbor within 5 seconds as recovery information. If the information can not be retrieved or the node fails to deliver to the system continuously, it will be marked as crashed or unavailable sectors for future inspections.

The data management layer can be formed from this aggregation behavior and adopt a heterogeneous structure with multiple databases. NOSQL and SQL databases are used to achieve high performance in both real-time applications and analysis queries. Specifically, the most frequently accessed data are stored in the in-memory caching subsystem so that the data can be accessed and updated frequently. This type of subsystem can achieve a milliseconds latency and a high throughput in data updating and can exchange by using an in-memory key-value data structure. Furthermore, using caching information reduces the pressure of the database as it acts like the fast-accessing media component in the system. Therefore, a real-time application like the parking assignment can keep providing surrounding parking statuses while the users check these statuses simultaneously through the web browser without waiting for the final aggregation information. Then, only the structured and fully-aggregated data are stored in RDBMS and accessed with SQL. Meanwhile, all the data generated from the previous layer are archived in the NOSQL big data storage subsystem, which supports only the non-realtime offline data processing applications such as the duration of a vehicle parked, the average occupant space throughout a day, etc.

4.4 Optimal Routing Parking Assignment

At the system's application layer, optimal routing and parking assignments are chosen as a modern CV-SPS fundamental task. A conventional way to solve the association between the predicted vacant spot and new parking requirements is to treat it as an assignment problem that can be solved using the Hungarian algorithm. N available parking slots stored in the middleware layer will be assigned as the targets for M parking requirements, forming a $M \times N$ matrix. As the Hungarian algorithm only accepts a square matrix, a standard approach is to add to M dummy requests or N targets depending on which size is smaller. However, this can lead to redundant optimization when the gap between M and N is diverse. Therefore, a prioritized protocol based on the number of available spots with the number of requests is provided. If M is much bigger than N, the assignment's top N early requests will be selected. Meanwhile, if M is much smaller than N, M slots will be registered from parking lots as having low occupancy rates. In addition, to conduct a sufficient assignment from $M \times N$ matrix, each element is scored by cost assignment for ranking and minimizing the assignment budget. Therefore, taking into account the intention of reducing CO2 emission and maximizing the parking profit for the operator, a cost function C is formulated which takes three main factors: time traveling, parking distance, and parking price with γ as the weight controller. To reduce the complexity of the matrix, only the optimal routing distance has been addressed.

$$C = \gamma C_{price} + (1 - \gamma)(C_{travel} + C_{distance}) \tag{4.7}$$

Chapter 5

Experiments

In this chapter, experiments have been conducted to estimate the performance of the system and its efficiency in a real-world scenario. As the main scope of the thesis is inference error reduction and provide precise predictions, the aggregation layer benchmark is excluded and the following experiments are conducted. First, the overall performance of the system and the impact of using the spatial module and reference key points for improving parking localization are studied. Second, the effectiveness of filtering out unstable inference to the model's actual performance is investigated. Third, based on the filtered out samples, the impact of adding these samples to the training pipeline is also investigate. Finally, the application impact of the system on two primary SPS tasks: optimal routing and parking assignment tasks is analyzed.

5.1 Experimental Setup

5.1.1 Dataset settings

PKLot dataset [15] and proposed SNU-SPS dataset is used for the solutions benchmark. CNRPark-Ext[3] is not addressed because this dataset's parking spot border lines are faded and not consistently visible. PKLot dataset contains three sub-datasets captured from a high view: PUCPR, UFPR05, and UFPR04, which leads to a small scale of parking spots. All of these sub-dataset parking locations are partially annotated. Thus, masks for each sub-dataset to clip out non-annotated parking regions are provided to avoid false positives during training and make it suitable for the detection benchmark. The dataset from each sub-dataset is split in half for training and testing, as the PKLot authors suggested. For the SNU-SPS, the test set is separated as described in the previous chapter. However, due to a small label of *illegal* and *restricted* classes in the dataset, the test set is only addressed with two classes: *occupied* and *available*. In addition, because the scope of the SNU-SPS dataset is to detect correctly in a sector, only medium and large ground truths are addressed in the test set. During the assignment application test, the detection will be filtered out of overlapping areas by our provided masks.

5.1.2 Training settings

For experiments, with the scope of an efficient and quick response in the first system layer, the single-stage object detector: SSD-Mobilenet (denoted MBN), Mobilenet-FPN (our model backbone, denoted MBN-FPN), and OcpDet, is addressed. The training engine for these models is Tensorflow Object Detection API, a public open source and friendly for the edge model deployment. The training protocol for those detectors is 25000 iterations with a batch size of 48

by SGD optimizer. To keep the detection robust to the small parking spots, high-resolution 896-pixel input is used instead of the traditional 300-pixel or 640-pixel. This setting is applied for both datasets, PKLot and SNU-SPS.

5.2 Detection Layer Performance

Method	Test Set	Recall	mAP			Classification Score	
		(0.5:0.95)	0.5	0.75	(0.5:0.95)	Occupied	Available
OcpDet	PUCPR	0.88	0.98	0.98	0.84	0.98	0.98
MBN-FPN		0.76	0.86	0.85	0.72	0.83	0.90
MBN[35]		0.41	0.48	0.35	0.31	0.49	0.46
Classifier[38]		-	-	-	-	0.99	0.99
OcpDet	UFPR05	0.98	0.99	0.99	0.97	0.99	0.99
MBN-FPN		0.84	0.93	0.90	0.82	0.93	0.94
MBN[35]		0.42	0.51	0.42	0.37	0.50	0.53
Classifier[38]		-	-	-	-	0.99	0.99
OcpDet	UFPR04	0.96	0.99	0.99	0.93	0.99	0.99
MBN-FPN		0.83	0.95	0.90	0.79	0.95	0.96
MBN[35]		0.43	0.52	0.44	0.36	0.51	0.53
Classifier[38]		-	-	-	-	0.99	0.99
OcpDet	SNU-SPS	0.56	0.81	0.48	0.47	0.83	0.80
MBN-FPN		0.54	0.77	0.46	0.45	0.80	0.74
MBN[35]		0.51	0.71	0.48	0.44	0.73	0.69
Classifier[38]		-	-	-	-	0.86	0.85

Table 5.1 PKLot and SNU-SPS Detection Benchmark

In this part, the efficiency of the spatial module and the localization improvement of additional reference key points is studied. The main metric for the detection evaluation is mAP(mean average of precision) and recall ranging from 0.5 to 0.95 IoU(intersection over union). Meanwhile, as the classification task is the side task to benchmark with a classification approach, the mAP(0.5) is addressed from each class for the comparison. As results are summarized in Table 5.1, the performance of the localization attention leads in all data settings. [31] solution is not compared because of its insufficient model's performance.

In the PKLot dataset, MBN struggles to learn the features because it lacks top feature generation from the grid and lines to capture the small objects. Meanwhile, thank the FPN, both MBN-FPN and OcpDet outperform MBN in this dataset. Due to the fixed location of the parking lot's captures, OcpDet can strongly overfit the position of each parking space and turn it into a grid classifier. As demonstrated in Fig 5.1, the localization guidance from the anchor mask generations helps the anchor patches avoid negative samples that do not belong to the parking area. The model can boost the performance to near perfection using the anchor mask head during training. In addition, by using additional key points, the approach improves localization detection and preserves its tightness among scales.

In contrast to the unchanged parking layout of the PKLot dataset, the SNU-SPS creates more challenges for the model to select the correct anchor patch due to its various capture positions. This various view-points make not only OcpDet but also other detectors struggle. In this dataset, both the reference key points and the anchor mask generator are inefficient as the model grid is not stationary. Despite the challenge of adaptability, when looking into the activation of the anchor mask through different scales of the spatial module in Fig 5.1, the obtained mask on the parking lot still has denser attention

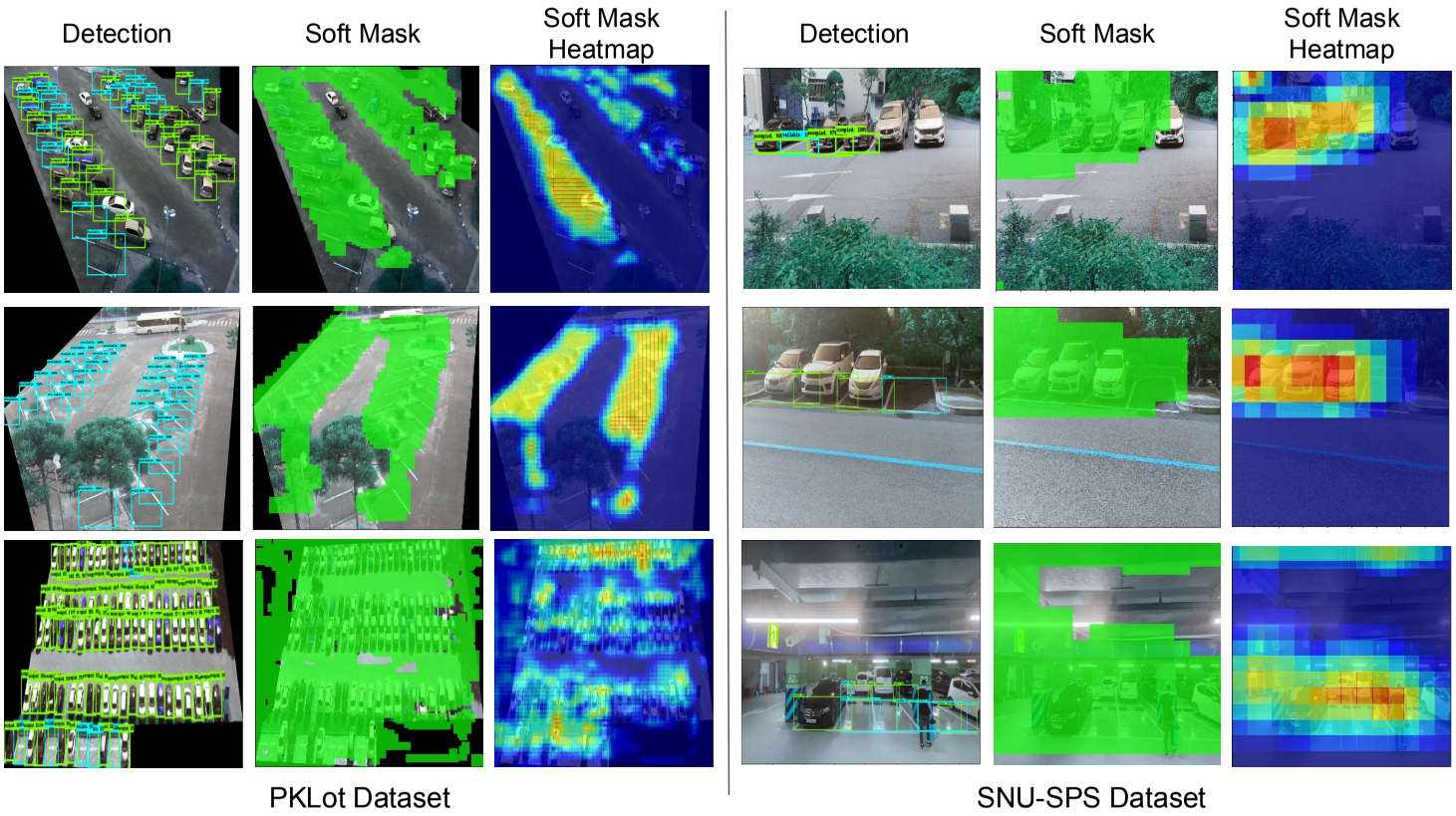

Figure 5.1 Visualization occupancy detections, anchor mask predictions coverage and anchor mask predictions activation on PKLot dataset and SNU-SPS dataset

than other foreground predictions. Combined with the impact of reference key points, OcpDet leaves a gap of nearly 5% on mAP(0.5) to original MBN-FPN and 10% to MBN. Moreover, in the first row of Fig 5.1, it shows that OcpDet is not sensitive to the car appearance. The prediction only activates inside in the parking zone where the lines are visible.

To deeply understand the impact of the spatial estimator module, the way of predicting the foreground from hor-grid features through the FPN is investigated. It has been shown that regardless of the grid's anchor box size, it will be considered a foreground if the classification activation is switched on. As the pixel activation map through the class activation map (HiRes-GradCam) [16] can help researchers understand which pixel contributes to the final decision. This method is implemented in this experiment. As demonstrated in Fig 5.2, there is a high density of pixel activations at each middle of parking slots whenever the model predicts the location class, which is reasonable as a class determination belongs to what has inside a parking border. The class confidence is reduced when the number of surrounding activation maps is dimmed Fig 5.2-a and disappears in Fig 5.2-b. This visualization once again shows that the activation map only produces a reliable decision when pixel activation falls inside a parking spot, which implies the spatial module's activation can evade this problem by adding its activation of the anchor mask prediction.

5.3 Result Filter Layer Performance

To evaluate the result filter efficiency, OcpDet is addressed solely on SNU-SPS as there is still room for model improvement. The error of an inference sample is calculated from the formula Eq 4.6. As the original performance mAP is 0.81, there is roughly 19% of the data is incorrectly determined. A maximum

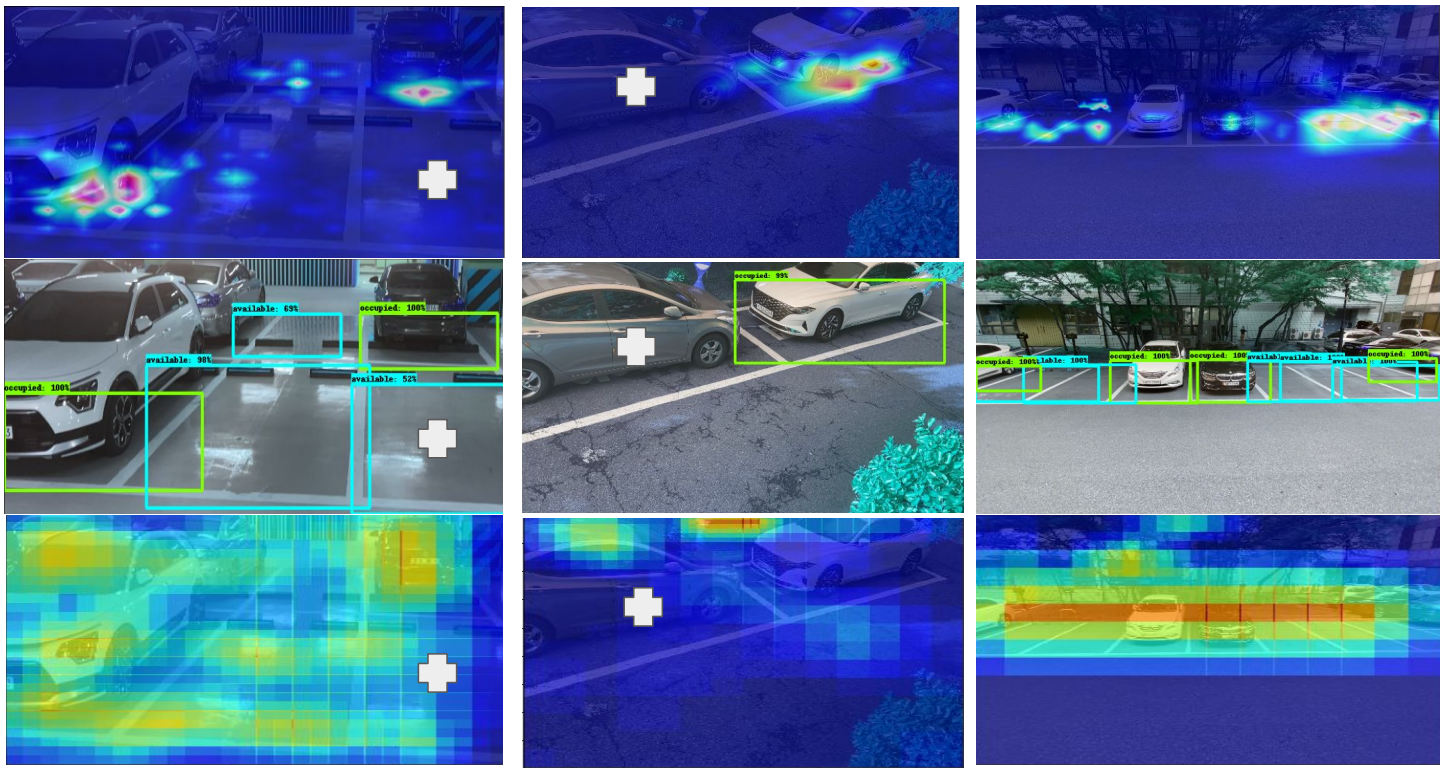

Figure 5.2 HiRes-GradCam activations following by Image Detections (green: "occupied", blue: "available") and their coresponding anchor mask activations. White arrows stand for weak detection area: (a) low confidence score, (b) zero detection. c) is an exmaple of our model inference

20% of the test set is allowed from the test set to emphasize the effectiveness of filtering out poor results from the pool. By limiting the number of samples, the experiment avoids removing low inference error sample and gives a fair benchmark with other approaches. For the purpose of testing out the benefit of the spatial module and the training error module, the overlapping γ of the spatial module is locked at 0.7, while the number of sensitive batch-norm layers of the training error module is 5. In Table 5.2, the results are summarized on result filter using: only the spatial module $OcpDet_{spatial}$, only the training error module $OcpDet_{te}$, both modules with the max aggregation $OcpDet_{max}$. To keep the experiment robust, the results from using the average aggregation $OcpDet_{avg}$ and applying the elegant solution Learning Loss [41] $OcpDet_{ll}$ and Contextual Disparity [37] $OcpDet_{cdcs}$ are also used. The benchmark on [36] and [13] have been omitted from the experiment due to their low-scalability [36] and mon-frendly edge deployment [13]. For a fair performance at the edge level, Jetson-Nano has been chosen as the edge benchmark device for FPS inference. The final FPS is calculated after passing all the images from the test set through the OcpDet model.

According to the experiment, the result filter layer has boosted the model's overall accuracy, proving that the filter can ensure a better quality for the detector regardless of which module dominates. Moreover, the max aggregation also shows a better factor balance than the average aggregation method. For a close inspection, although the spatial filter can score and remove unreliable results, its behavior is still affected by the impact of the input on the model knowledge. Thus, its results are slightly lower than the training error module's approach. In addition, the experiment has also shown that using the training error can strongly boost the model performance and avoid mistakes. It proves that using the batch norm layers as a reference can be better instead of predicting the

Method	FPS	Recall	mAP			Classification Score	
		(0.5:0.95)	0.5	0.75	(0.5:0.95)	Occupied	Available
OcpDet	6.6±1.3	0.56	0.81	0.48	0.47	0.83	0.80
$OcpDet_{spatial}$	6.6 ± 1.3	0.56	0.83	0.50	0.47	0.85	0.82
$OcpDet_{te}$	5.2±0.9	0.58	0.85	0.50	0.48	0.86	0.83
$OcpDet_{ll}$[41]	6.1±0.5	0.55	0.82	0.49	0.47	0.83	0.81
$OcpDet_{cdcs}$[22]	3.5±1.5	0.56	0.84	0.49	0.48	0.84	0.84
$OcpDet_{max}$	4.8±0.9	0.57	0.84	0.49	0.48	0.85	0.83
$OcpDet_{avg}$	4.8±0.9	0.56	0.83	0.48	0.47	0.84	0.81

Table 5.2 Result Filter Performance on the OcpDet by SNU-SPS Detection Benchmark

model error directly like [41] approach or two-times inference comparison like [22]. However, it also shows that using the training error module is expensive for the inference. Due to the heavy tensor acquisition during the computation for the error, it takes away nearly 2 FPS performance. This drawback can temporarily be avoided by omitting the training-error module and using the solely the spatial module, which is at the third of the ranking performance in the benchmark.

5.4 Active Learning Performance

In this section, the active learning performance is benchmarked by randomly splitting the training dataset into three parts. The first part contains 2000 images. Meanwhile, the second part and the last part is equally divided 424 images. The scope of this experiment to the impact of adding high inference error on the model improvement through active learning steps. Under this intention, the test set is treated as the evaluation of the model after active learning steps. At first, the OcpDet is trained with the first part of the training set and evaluated as the initial performance (step #0). The second and third parts of the training set are treated as the inference environment. Under this setting, a trained OcpDet inferences on the two inference environment and high inference error samples are chosen to append to the training pool, which is an active learning step. This procedure has been introduced in various active learning benchmark [37, 13, 23]. However, the number of added samples is limited based on the performance of the previous step, which is the procedure of the result filter benchmark. This experiment is conducted five times for a well-defined evaluation on the same methods having been introduced at part 5.3. The result is summarized in Table 5.3.

Method	Active Learning Step (0.5 mAP)		
	#0	#1	#2
$OcpDet_{spatial}$	0.72±0.23	0.78±0.18	0.83±0.21
$OcpDet_{te}$	0.72±0.23	0.79±0.12	0.85±0.15
$OcpDet_{ll}$[41]	0.72±0.23	0.77±0.16	0.83±0.09
$OcpDet_{cdcs}$[37]	0.72±0.23	0.78±0.11	0.84±0.16
$OcpDet_{max}$	0.72±0.23	0.79±0.17	0.85±0.12
$OcpDet_{avg}$	0.72±0.23	0.78±0.15	0.83±0.16

Table 5.3 Two Steps of Active Learning Performance on the OcpDet by SNU-SPS Detection Benchmark

As demonstrated in Table 5.3, using training error information is highlighted as the best approach for active learning. By leading at every step, it consist less data than other approaches to reach the best knowledge for the model. Moreover, the method preserves a stable and low variance for the model learning, which is contrasted to the spatial error. Other results from the spatial error through aggregation also hints a lower performance in addressing active learning performance. This outcome shows the spatial error information is not a straight way for improving model knowledge and should stay as a way for reducing the inference error. In the mean time, [22] once again shows better performance than [41] method and closely matches the training error information approach. However, it costs more data and infenrences, which is a big drawback for future active learning steps. Hence, further expansion on dataset can draw the gap between two approaches.

5.5 Optimal Routing and Parking Assignment

To make a comprehensive benchmark on the impact of detection results on the assignment application layer, the traffic information over two months from the Korean government website *http://www.utic.go.kr* is collected. This traffic information is associated with the test set to form a close-loop simulation. In the simulation, each day from 3 to 6 pm, there will be fixed 100 requirements for booking a vacant spot to 6 available parking lots in the test set. The suggested optimal road for each request will be assigned from the MapQuest API. A correct spot assignment is considered by the Hungarian assignment [25] from the masked-out test label. This assignment is treated as the ground truth for the benchmark.

Figure 5.3 The cost errors and the assignment errors: averaging simulation for 5 days at 6 parking lots

Then, two evaluation metrics are computed: cost error and assignment error. The cost error is computed by the absolute error of the ground truth assignment

cost $C_{g,i}$ and the vacancy detection assignment cost $C_{p,i}$, which is designed by Eq. 5.1. The assignment cost is computed after getting the assignments. From each assigned parking lot in 6 parking lots, the total number of booked slots N is computed and compared between the ground truth $N_{g,i,j}$ and the detection $N_{p,i,j}$ by an absolute error. To keep a fair comparison among days of simulation, the two values are normalized from 0 to 1 and γ is set to 0.5.

Each day simulation will be performed from the same corresponding weekday in the traffic simulation. Thus, each simulation for a day is repeated at least eight times to capture the model's average performance due to different traffic statuses across two months.

$$C = \gamma C_{price} + (1-\gamma)(C_{travel} + C_{distance}) \tag{5.1}$$

$$Err_{cost} = \sum_{i=3}^{6} \frac{\mid C_{g,i} - C_{p,i} \mid}{C_{g,i}} \qquad Err_{assign} = \sum_{i=3}^{6}\sum_{j=1}^{6} \frac{\mid N_{g,i,j} - N_{g,i,j} \mid}{N_{g,i}} \tag{5.2}$$

From Fig. 5.3, the system allows the system to operate at most 40% error for the cost-minimizing budget while maintaining at least 70% correct on assignment. Because the cost error is proportional to the traveled distance, wrong assignments on far-distance drives can cause a huge gap in the optimal cost. Meanwhile, the error is not much in terms of matching the number of assignments. From these metrics, operators will benefit the most as their vacant spaces will automatically be assigned with a minimal error. In contrast, some drivers may get some disadvantages from the system assignment. Further improvement needs to be done on the driver metric has to be addressed for a throughout value benchmark.

Chapter 6

Conclusion & Future Work

This thesis proposes a novel end-to-end CV-SPS with a detailed benchmark on both old and new datasets and stress test simulations. Even though the proposed dataset is small, it shows challenging factors, and it is the first dataset for computer vision with full CV-SPS scope. Moreover, the system has proved its efficiency for the CV-SPS scope and can close the gap to the classifier approach when addressing stationary views. The system also provides a novel filtering method that preserves better model interpretation performance by injecting two new modules: the training error module and the spatial estimator module. Especially, it provides better performance than uncertainty capture methods without requiring multiple passes for determination. The system also proves to operate with reliable performance when the connection is stable, and the capture results do not dynamically change between seconds.

However, there are still plenty of drawbacks for the system. Due to the limited size of the dataset, the performance of the approach has not been addressed to illegal parking and restricted parking classes. Moreover, active learn-

ing performance can only been conducted for limited steps. Therefore, in the future, further expansion will be made on the dataset to increase the size of the dataset and assist a better and more realistic measurement of the aggregation layer performance. The dataset will also be improved with additional functional information for SPS, such as vehicle reidentification or parking type selection for a complete intelligent application. Moreover, the OcpDet will also be addressed and optimized to improve edge devices' performance and make it robust to speed-sensitive speed inference applications. Lastly, the system should move from the centralized system to the distributed system to maximize the benefit and the scalability of system. In the current design, if the central server fails or crashes, it is impossible to maintain the service and the applications.

Bibliography

[1] Al-Turjman, F., Malekloo, A.: Smart parking in iot-enabled cities: A survey. Sustainable Cities and Society **49**, 101608 (2019)

[2] Lisboa de Almeida, P.R., Honório Alves, J., Stubs Parpinelli, R., Barddal, J.P.: A systematic review on computer vision-based parking lot management applied on public datasets. arXiv e-prints pp. arXiv–2203 (2022)

[3] Amato, G., Bolettieri, P., Moroni, D., Carrara, F., Ciampi, L., Pieri, G., Gennaro, C., Leone, G.R., Vairo, C.: A wireless smart camera network for parking monitoring. In: 2018 IEEE Globecom Workshops (GC Wkshps). pp. 1–6. IEEE (2018)

[4] Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., Vairo, C.: Deep learning for decentralized parking lot occupancy detection. Expert Systems with Applications **72**, 327–334 (2017)

[5] Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M.: The power of ensembles for active learning in image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9368–9377 (2018)

[6] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)

[7] Bohush, R., Yarashevich, P., Ablameyko, S., Kalganova, T.: Extraction of image parking spaces in intelligent video surveillance systems (2019)

[8] Brust, C.A., Käding, C., Denzler, J.: Active learning for deep object detection. arXiv preprint arXiv:1809.09875 (2018)

[9] Bura, H., Lin, N., Kumar, N., Malekar, S., Nagaraj, S., Liu, K.: An edge based smart parking solution using camera networks and deep learning. In: 2018 IEEE International Conference on Cognitive Computing (ICCC). pp. 17–24. IEEE (2018)

[10] Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Zeroq: A novel zero shot quantization framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13169–13178 (2020)

[11] Chawla, A., Yin, H., Molchanov, P., Alvarez, J.: Data-free knowledge distillation for object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3289–3298 (2021)

[12] Chitta, K., Alvarez, J.M., Lesnikowski, A.: Large-scale visual active learning with deep probabilistic ensembles. arXiv preprint arXiv:1811.03575 (2018)

[13] Choi, J., Elezi, I., Lee, H.J., Farabet, C., Alvarez, J.M.: Active learning for deep object detection via probabilistic modeling. arXiv preprint arXiv:2103.16130 (2021)

[14] Cookson, G.: Parking pain—inrix offers a silver bullet. INRIX—INRIX. Online. Available: http://inrix.com/blog/2017/07/parkingsurvey. Accessed: November **21** (2017)

[15] De Almeida, P.R., Oliveira, L.S., Britto Jr, A.S., Silva Jr, E.J., Koerich, A.L.: Pklot–a robust dataset for parking lot classification. Expert Systems with Applications **42**(11), 4937–4949 (2015)

[16] Draelos, R.L., Carin, L.: Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks. arXiv e-prints pp. arXiv–2011 (2020)

[17] Feng, D., Wei, X., Rosenbaum, L., Maki, A., Dietmayer, K.: Deep active learning for efficient training of a lidar 3d object detector. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 667–674. IEEE (2019)

[18] Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059. PMLR (2016)

[19] Haussmann, E., Fenzi, M., Chitta, K., Ivanecky, J., Xu, H., Roy, D., Mittel, A., Koumchatzky, N., Farabet, C., Alvarez, J.M.: Scalable active learning for object detection. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 1430–1435. IEEE (2020)

[20] Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal network. In: Proceedings of the IEEE international conference on computer vision. pp. 4145–4153 (2017)

[21] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)

[22] Ismail, Zhiding, Anima, Laura, Alvarez, J.M.: Rationalizing the labeling costs for training object detection. CVPR (2021)

[23] Kao, C.C., Lee, T.Y., Sen, P., Liu, M.Y.: Localization-aware active learning for object detection. In: Asian Conference on Computer Vision. pp. 506–522. Springer (2018)

[24] Kirtibhai Patel, R., Meduri, P.: Faster r-cnn based automatic parking space detection. In: 2020 The 3rd International Conference on Machine Learning and Machine Intelligence. pp. 105–109 (2020)

[25] Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)

[26] Li, X., Chuah, M.C., Bhattacharya, S.: Uav assisted smart parking solution. In: 2017 international conference on unmanned aircraft systems (ICUAS). pp. 1006–1013. IEEE (2017)

[27] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)

[28] Media, U.N.: 68% of the world population projected to live in urban areas by 2050, says un. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html (2018), last Accessed: 2022-06-08

[29] Nieto, R.M., García-Martín, Á., Hauptmann, A.G., Martínez, J.M.: Automatic vacant parking places management system using multicamera vehicle detection. IEEE Transactions on Intelligent Transportation Systems **20**(3), 1069–1080 (2018)

[30] Nyambal, J., Klein, R.: Automated parking space detection using convolutional neural networks. In: 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech). pp. 1–6. IEEE (2017)

[31] Padmasiri, H., Madurawe, R., Abeysinghe, C., Meedeniya, D.: Automated vehicle parking occupancy detection in real-time. In: 2020 Moratuwa Engineering Research Conference (MERCon). pp. 1–6. IEEE (2020)

[32] Paidi, V., Håkansson, J., Fleyeh, H., Nyberg, R.G.: Co2 emissions induced by vehicles cruising for empty parking spaces in an open parking lot. Sustainability **14**(7), 3742 (2022)

[33] Polycarpou, E., Lambrinos, L., Protopapadakis, E.: Smart parking solutions for urban areas. In: 2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM). pp. 1–6. IEEE (2013)

[34] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)

[35] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)

[36] Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489 (2017)

[37] Sharat, Himanshu, Saket, Chetan: Contextual diversity for active learning. In: ECCV. pp. 137–153 (2020)

[38] Valipour, S., Siam, M., Stroulia, E., Jagersand, M.: Parking-stall vacancy indicator system, based on deep convolutional neural networks. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). pp. 655–660. IEEE (2016)

[39] Varghese, A., Sreelekha, G.: An efficient algorithm for detection of vacant spaces in delimited and non-delimited parking lots. IEEE Transactions on Intelligent Transportation Systems **21**(10), 4052–4062 (2019)

[40] Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Transactions on Circuits and Systems for Video Technology **27**(12), 2591–2600 (2016)

[41] Yoo, D., Kweon, I.S.: Learning loss for active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 93–102 (2019)

Korean Abstract

스마트 카메라 주차 시스템의 개념은 수십 년 동안 존재했지만, 시스템의 확장성과 신뢰성을 완전히 다룬 접근 방식은 적었다. 스마트 주차 시스템의 초석은 점유 감지이기 때문에, 현재 대부분의 시스템은 주차 지점 아래에 매설된 센서를 사용해왔다. 하지만, 이는 주차 공간의 수에 따라 가격이 상승하기 때문에 대규모로 솔루션을 확장할 경우 매우 많은 비용이 들게 된다. 또한, 최근 다양한 주차 공간에 CCTV가 설치되고 있는 만큼, 컴퓨터 비전 방식을 통한 점유 감지가 더욱 유리해졌다. 그럼에도 불구하고, 전통적인 방식은 분류 백본을 사용하여 수동으로 레이블이 지정된 그리드에서 지점을 예측한다. 이 과정은 시간이 많이 소요되고 제품화 단계에서 시스템의 확장성을 잃게 된다. 또한 딥 러닝 접근법을 고려할 때, 솔루션이 일부 상황에 대해서만 부분적으로 일반화되어 추론 중에 잠재적인 오류를 많이 발생시킬 수 있다. 이는 스마트 카메라 주차 시스템에서 컴퓨터 비전을 사용하는 이점을 크게 감소시킨다. 이러한 단점들은 이 논문의 주제인 빠르고 추론 오류가 적은 스마트 카메라 주차 시스템을 요구한다. 본 논문에서는 기존 분류 방법을 OcpDet이라는 CNN 감지기로 대체하여 추론 시간을 향상시키고, 엣지 장치들에서 감지를 수행시켜 확장 및 로드 밸런싱이 가능하도록 한다. 따라서 OctDet 백본은 엣지 장치에 적합한 빠르고 가벼운 추론을 위해 Mobilenet을 기반으로 구동된다. 훈련 오류 모듈과 공간 추정 모듈의 정보를 모델에 주입하여 발생 가능한 탐지 오류에 대응하였다. 훈련 정보는 OcpDet의 배치 정규화 통계에서 추출되어 해당 장면이 훈련 지식과 일치하는지 여부를 알려준다. 만약 장면이 도메인 지식을 벗어난다면, 능동 학습 반복 및 추가 검사를 통해 장면을 수집하여 모델을 개선한다. 한편, 공간적 지식은 신뢰도 향상 기법을 통해 추론 중 탐지 오류를 피하고 잘못된 위치의 bounding box를 억제한다. 향상된 결과를 기반으로 장면에 대한 공간 오류는 훈련 오류와 결합되어 결과물에서 포함되는지 여부를 결

정할 수 있도록 한다. 구현된 시스템은 기존 PKLot 데이터셋을 기반으로 벤치마크 되었으며 느린 분류 솔루션들과 비교하여 경쟁력 있는 결과를 내었다. 시스템의 확장성과 신뢰성을 측정하기 위해 추가적인 SNU-SPS 데이터셋을 생성해 다양한 시점 및 주차 할당 작업에서 시스템을 평가를 수행하였다. 이러한 작업들을 위해, 서울대 캠퍼스를 둘러싼 다양한 주차장의 다중 엣지 카메라로부터 실험을 수행하였고, message-broker 프로토콜을 통해 주차 감지 정보를 수집하였다. SNU-SPS 데이터셋의 결과는 논문의 접근 방식이 작은 오류 trade-off을 가지고 실제 응용 프로그램에 적용 가능하다는 것을 보여준다.

주요어: 스마트 주차 시스템, 점유 감지, 추론 오류, 엣지 장치
학번: 2020-25413

Acknowledgements

First and foremost I am extremely grateful to my advisor, Professor Cha Sang-Kyun for his invaluable advices, continuous support, and patience during my Master study. His immense knowledge and plentiful experience have guided me throughoutly to complete this topic and encouraged me on my academic research and daily life. I can not imagine how I can complete this topic without him. I also give my appreciation to all Professors of Department of Electrical & Computer Engineering, who have taught me valuable lectures and provided me advanced knowledge in my Master course to form a concrete foundation for my thesis.

Secondly, I would like to thank all the members in the PIDLab (Peta-scale In-memory Database Laboratory). It is their kind help and support that have made my life in the Korea an easy and wonderful time. Especially, I want to thank individually Le Van Duc, Bui Tien Cuong and To Hai Thien for their technical support and suggestion on my research work and life adaptation. Besides, I want to give my special thanks to Nguyen Gia Quan, Vo Thi Chinh, Huynh Thanh Hau and other members of Vietnamese Community in Seoul National University for always sharing their experience and helping me with my academic life in Korea.

Furthermore, I would like to express my huge gratitude to Zhang Yena, Dewandra Bagus, Aregay Mulugeta, Khaing Khaing Htun, Renke, Batbayar Enkhbaatar, Pawee Chiranothai, Marianne Chang, Soumyakanti Bose, Sambit Ghosh, Lart Souy, Bai Shen Feng, Hakim, Mathieu Touly, Tahar Kalem and all foreign friends for believing in me and encouraging me to archive my Master degree.

Finally, I want to convey my deepest gratefulness to my family. Without their tremendous understanding and support in the past three years, it would be impossible for me to complete my study.