**Computational intelligence based point of interest detection by video surveillance implementations**

Emre Tercan1\*, Serkan Tapkın2, Furkan Küçük3, Ali Murat Demirtaş3, Ahmet Murat Özbayoğlu4, Abdussamet Türker5

1 Department of Traffic Safety, 13th Region, General Directorate of Highways, 07090, Antalya, Turkey

2 Department of Civil Engineering, Faculty of Engineering, Bayburt University, 69010, Bayburt, Turkey

3 Department of Electrical Engineering, Faculty of Engineering, TOBB University of Economics and Technology, 06560, Ankara, Turkey

4 Department of Computer Engineering, Faculty of Engineering, TOBB University of Economics and Technology, 06560, Ankara, Turkey

5 Department of Computer Engineering, Faculty of Engineering, TOBB University of Economics and Technology, 06560, Ankara, Turkey

**Abstract**

Latest advancement of the computer vision literature and Convolutional Neural Networks reveal lots of opportunities that are being actively used in various research areas. One of the most important examples for these areas is autonomous vehicles and mapping systems. Point of interest (POI) detection is a rising field within automated video tracking and autonomous mapping systems. Within the last few years, the number of implementations and research papers started rising due to the advancements in the new deep learning systems. In this paper, our aim is to survey the existing studies implemented on POI detection systems, in particular designed to be used for automated vehicles and automatic mapping systems. It has been tried to address this problem from a transportation and vehicle industry perspective. At the same time, a deep learning based POI detection model will be introduced that can be embedded into automated cars such that real-time video analysis and POI detection might be achievable in actual usage for future implementations.

**Keywords:** point of interest detection, YOLOv2 algorithm, faster R-CNN, deep learning, video surveillance

1. Introduction

Automated image analysis has always attracted researchers and industry professionals due to the wide implementation areas ranging from text-to-speech systems to automated vehicles. With the advancements in the computational power, data collection capabilities through IoT/edge computing devices, smart sensors, faster communication through high speed wireless networks and the rise of the artificial intelligence and machine learning models, these automated image analysis systems became widely available and more feasible than ever. Within the field, video surveillance and automated object detection draws more attention, mostly from a security viewpoint. However, these systems are also valuable for autonomous vehicle advancements [1-4].

Driverless cars and intelligent transportation systems have become an important research field [5,6]. Lots of industries and manufacturers are trying to introduce (or already introduced) their fully automated driverless systems into the market. Object recognition is an essential part of such systems, and their flawless performance is crucial for the successful future of driverless car technology. Point of interest (POI) detection is gaining importance within the automated object recognition area. Even though there have been some prior studies for accurate detection of POIs like restaurants, gas stations, historical sites, etc., most of these studies are based on location estimation systems like Global Positioning System (GPS). Due to the dynamic nature of driving and road conditions, the GPS based geocaching systems might not reflect the actual environmental surroundings accurately. Also, within the urban areas, the GPS signals are not always accessible in some occasions. Vision based POI detection models can be a viable solution to address such challenges [7-9].

In this paper, it is aimed to provide detailed information about the previous studies implemented on POI Detection, in particular in automated systems. Since main focus will be on the application of POI detection on automated vehicles, particular importance will be given to image and/or video based POI detection. The existing studies will not only be reviewed, but also some of the learning models that are used in such systems will be introduced. In addition, the problems and opportunities for the future will be pinpointed. In the second part of the study, a deep learning based POI detection model will be introduced that has been developed as a case study in order to provide a roadway to present how such models might be embedded into the automated cars.

The rest of the paper is structured as follows: After this brief introduction, definitions for the POI detection problem from different perspectives and problem domains will be introduced. Then, computer vision and previous POI detection studies for different domains and implementation areas will be addressed which includes some comparisons between different approaches. Later on, the machine learning algorithms used in automated POI detection models, in particular deep learning models, will be covered in Section 3. A Proof of Concept (POC) POI detection model based on some deep learning models in a real car traffic video use case will be provided in Section 4 along with the performance results and analysis. The discussions, open issues and future direction for the field will be presented in Section 5. The conclusions of this study will be presented in Section 6.

2. POI detection

A POI refers to the location of a defined point in a particular coordinate system that users may find relatively advantageous or interesting. Hotels, camping areas, tourism facilities, restaurants, fuel stations are particular examples of POIs. In the literature, various studies have been conducted about the POI detection/discovery in different application areas. Ruta et al. [10] proposed a new discovery tool for mobile devices at Augmented Reality (AR) for semantic addition of nodes in crowd-sourced OpenStreetMap (OSM) mapping. As an AR discovery facilitator, this tool provides an automatic match between the utilizer characteristics and the source definitions. Yu et al. [11] utilized machine learning algorithms and similarity metrics to estimate and classify the multi-feature resemblance measurement outcomes for multi-vendors’ POI data. They stated that the Support Vector Machine (SVM) method can find duplicate POIs more efficiently compared to the naïve Bayesian classifier and decision trees. Hao et al. [12] presented a novel system that leverages camera locations and viewing directions. This system has features that can automatically detect interesting regions and objects (POIs). Rohella and Singh [13] developed an algorithm that looks for nearby spatial objects for potential POIs such as travel and commercial centers. Shu et al. [14] presented a three-step algorithm that uses deep neural networks in order to learn and estimate POIs on 3D images by utilizing multiple feature identifiers.

Even though various methods have been proposed for POI detection, image and video based models have been of particular interest. In order to give a thorough analysis of vision based POI detection studies, it would be beneficial to briefly cover the computer vision techniques and basic models that are used in the literature.

3. Computer Vision

Computer vision (CV) is an interdisciplinary research area that mimics the human vision based on computer image understanding principles. The overall interest for research and implementations in this field has been in a steady upward increase, since not only the computers nowadays are able to do intense computations automatically, but also novel computational intelligence techniques like deep learning started dominating the industry [15,16]. Hence, this resulted in new, versatile implementations for intelligent decision-making systems in a large variety of autonomous applications like drones, autonomous cars, satellite systems etc. [17,18].

Object detection, which is one of the basic CV problems and the integral part of image analysis, can provide highly precious information for the semantic understanding of videos and images [15]. Object detection is a methodology related to CV and image processing to determine the locations of objects and the classes of each object (such as cars, buildings, roads) [19] in a particular set of images [15]. One of the most significant advancements in CV and object detection is the employment of deep learning models, in particular Convolutional Neural Networks (CNNs). CNNs not only relieved the burden of finding appropriate filters and algorithms from the academic area, it also enabled to make the most applicable and successful implementations for propriety usage.

In recent years, CNN models that are based on region-based detection algorithms (i.e. R-CNN) [20] have started demonstrating superior performances in object detection problems. However, extracting the optimum regions from the image frames requires intense computations especially in cases with complex background information. As a result, due to the high computational complexity, R-CNN and their advancement algorithms [21] are not generally preferred for real-time detection [16] implementations. R-CNN [22] provides better object detection accuracy, about 63% than conventional methods. However, there is a slow detection speed that requires 47 seconds for each image. To overcome this shortcoming, Fast R-CNN and Faster R-CNN have been proposed. Faster R-CNN increases the detection speed up to 7 frame per second (fps) [23] which might be sufficient for some applications that require real-time processing. However, many of the applications are complex and challenging, especially when the evaluation of visual information needs to be done in real time. Different lighting and meteorological conditions, complex backgrounds are parameters that make this task difficult [24]. Performing object detection at high accuracy is a challenging task due to such differences [15].

Increasing demand for real time object detection lead to development of a variety of algorithms. Some of the major proposed algorithms were R-CNN based algorithms. However, for the sake of keeping the delay of the object detection pipeline low, one-shot detection algorithms were proposed. YOLO (You Only Look Once) [25] algorithm deals with object detection as a regression issue. It separates the input image into 7x7 grids. It provides two bounding boxes for each grid cell. At the same time, each grid cell estimates a group class probability (including 20 classes) without considering the number of boxes. It processes the images at 45 fps with the average precision of 63.4% mean average precision (mAP), but performs more localization errors [23].

3.1. Computer Vision for POI Detection

Even though POI detection literature is mostly focused on static location-based models like geocaching through GPS, online approaches like real-time POI detection through offline image understanding methodologies can be viable alternatives for various automated systems. Autonomous vehicles and satellite systems are two examples for such applications of real-time POI detection using computational intelligence, in particular deep learning. One of such POI based object detection systems using deep learning is being employed in this paper and will be explained in Section 5. The proposed model is a preliminary model using YOLO network presenting a Proof of Concept use case based on a real video footage on a highway. The aim is to demonstrate the general methodology along with the environmental challenges and difficulties. The main focus on this study is roadside point of interest (POI) detection, in particular gas station detection. This paper addresses that problem, and attempts to provide a working solution on how current state-of-art convolutional networks perform on that task.

3.2. Existing Computational Intelligence / Deep Learning based POI Detection Studies in the Literature

There are a number of machine learning based POI detection studies in the literature. Ahmad et al. [7] defined the road markings and signs as a segmentation problem and studied the basic principles and algorithms of such implementations through a digital image analysis point of view. As a result, road markings detection is considered as one of the most commonly encountered POI detection problems. They also studied road markings as a POI detection problem, however, their focus was not only on detection of these markings, but also recognizing and classifying them into 10 classes using CNN with a recognition rate of 99% in their best network. Wu and Ranganathan [8] developed a template matching based road markings detection model using video input from a car camera. They used Maximally Stable External Regions (MSER) features for their template generations and training their model. Then they tested their model based on the generated templates. They trained their system based on 10 different road signs and the classification results indicate their POI detector was able to successfully detect designated POIs in real time. Bailo et al. [9] also used MSER features for their road markings detection model, however they embedded a density based clustering technique in order to provide a more robust POI detection model that would not be effected from varying road and meteorological conditions. Li et al. [26] presented an algorithm for road markings detection to be integrated with the autonomous navigation system of intelligent transportation systems. They used an Inverse Perspective Mapping (IPM) transformation through low level image processing techniques within their algorithm in order to enhance the detection performance. Greenhalg and Majid [27] and Kheyrollahi and Breckon [28] also developed automated road markings detection models in their studies. More recently, Lee et al. [29] used a Deep Learning Model called Generative Adversarial Networks (GAN) for unconstrained roadside marking recognition in a similar fashion, such that they generated a Tiny-YOLO detector for their baseline, then created samples for their training set for the generator and discriminator network for their adversarial learning model. Their results indicate a classification performance of over 95% using the same dataset as Wu and Ranganathan [8].

Careful analysis of the POI detection studies in the literature indicate that most of these studies were concentrated on the road markings, roadside signs and static or dynamic objects alongside the road. However, there are more possibilities for POI detection, such as roadside buildings, restaurants, phone boots, and gas stations. These kind of studies have not been undertaken in an intense manner throughout the vast amount of literature published till date. At this point we will introduce a POI detection model focused on gas station detection through offline model based on computer vision and deep learning.

4. Learning models for POI detection

4.1. Classical image processing models (very brief model descriptions)

Several handcrafted POI detection methods have been proposed using image processing. Shift Invariant Feature Transform (SIFT) [30] is one of the most common of these detection approaches. The SIFT algorithm attains distinguishing features from images. These features might be utilised to execute credible matching amongst dissimilar visions of the scene. The selected features do not change to scale and rotation, they are also resilient to changing perspective and brightness. The model works as follows: First, a difference-of-Gaussian (DoG) function is used to determine the key point features. Next, stable key points are found to determine the scale. After finding the scale, the orientation of key points is estimated using image gradient directions. Lastly, the key points are identified using the scale and orientation information.

Harris corner detector and the canny edge detector are popular shape detectors. Harris corner detector computes the autocorrelation of local image sections to find the edges and corners. On the other hand, canny edge detector mathematically analyses precise edge detection localization. Canny finds an analytical solution to obtain the optimal filter for edge detection.

4.2. Machine/Deep learning models

In most of the studies published in the literature, image processing/computer vision techniques are coupled with various machine/deep learning models. Some of these studies were mentioned along with the preferred techniques in the literature review enclosed in the previous section. Meanwhile, next section will introduce some of the newer models along with a case study tailored for gas station identification as a POI detection problem.

5. Case Study: POI Detection from car camera footage with Convolutional Neural Networks

So as to assess the efficacy of deep learning on the POI detection problem, a Proof of Concept model is developed and demonstrated using real video through a car surveillance camera. The developed model is clarified in depth in the subsequent sub-sections. Each sub-section will include the particular methods adopted for each process step within the model. On or after an application viewpoint, the methods in each step are employed to demonstrate how current state-of-the-art convolutional networks performs on the task of POI detection.

**5.1. Dataset Acquisition and Preparation**

Source of the data used to train the model was a video captured from a moderately busy divided Antalya-Manavgat highway in the Mediterranean region of Turkey. The acquired data was sufficient enough for providing discriminatory features, however was not large enough for training a deep learning model. Since that is the case, some augmentation methods were implemented to increase the number of data points within the dataset, as such the new dataset became adequate for training and testing.

Each image was passed from an augmentation pipeline in order to grow the dataset. The augmentation methods used were:

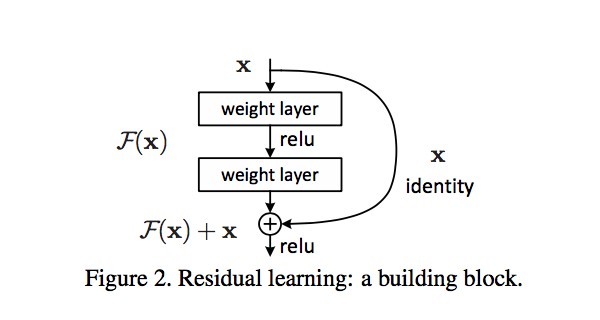
* Random Distortion: An image is distorted randomly with the probability of 0.5.
* Random Erasing: It was observed that the model over fits to undesired locations of POIs. To address that issue, some parts of the picture that are irrelevant to the region of interest are filled with Gaussian noise to prevent overfitting. As a result, during the training process, better model convergence is achieved avoiding overfitting issues.
* Random Brightness: Depending on the time of the day, road conditions and weather, lighting conditions may vary from frame to frame. In order to provide more robustness for appropriate region extraction, some form of normalization between individual frame variances might be helpful. Hence, augmenting data with random brightness is expected to make model more robust to these variances.
* Random Colour Shifting: In some tests, it is observed that the model tends to pick a colour and gives it a higher chance to classify the designated region as a positive, even though it is not the desired region of interest. To address that issue, a random colour shifting augmentation policy was employed.

5.2. State-of-the-Art Architectures and Transfer Learning

Obtaining good results from a machine learning model can be a challenging task and it might require countless iterations to achieve a satisfactory outcome. Many factors, ranging from data characteristics, feature properties, network topology to model choices and parameters, effect the overall performance during network training and it is always very difficult, if not impossible, to find the optimum machine learning model. “No free lunch theorem” states that there is no optimal architecture to fit in all data available [31]. However, we can achieve a reasonably good performance with a sub-optimal architecture. Employing state-of-the art architectures not only ensures the performance would be high enough to fulfill some expectations, it also enables to use the transfer learning technique [32].

Every year a competition is held called "ImageNet" [33]. It is a challenge to classify a given large dataset and trying to obtain best results via the developed architecture. The contributors of this competition are highly motivated researchers and largest companies in the world. Since they have enough computational power to train on a whole ImageNet dataset, it is only logical to use that architectures to ensure performance and use the pre-trained models they constructed. Transfer learning is mostly based on this paradigm [32].

In this paper, ResNet[34] was used as the backbone network. ResNet architecture employs deep residual connections between layers to address the vanishing gradient issue. Vanishing gradient issue is one of the main problems that the researchers encounter during training deeper neural networks [35]. The deeper network gets, the harder to get good gradients to train the first few layers. The gradients might get too small to have any meaningful effect during training and it becomes impossible to learn better filters. An example of the residual connection is shown in Fig. 1.



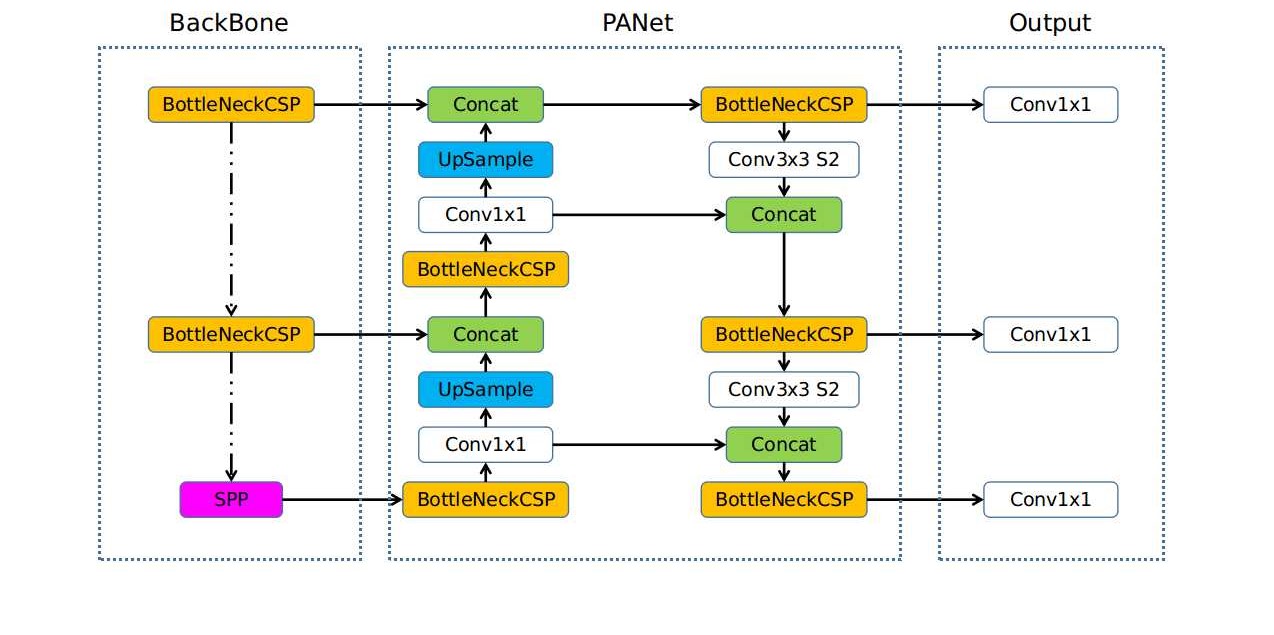
**Fig. 1** Residual Connection [36]

Transfer learning is a crucial part of the training process. The philosophy behind the idea is using the information obtained from other datasets and training experiences [32]. Considering a human interpretation of classifying images, humans tend to capture the features like edges, shapes and textures and obtain a result from this interpretation. Machine learning methods, in some level, rely on the same principals. However, since training a deep neural network takes lots of data to make the model capture good quality features from images, it is hard to train a CNN with a small dataset like this case. So, employing a state-of-the-art CNN architecture and using pre-trained model weights as a starting point for the training procedure generally achieves better performance than random weight initialization. Meanwhile, since the datasets indicate some similarities between them, the pre-trained network is expected to construct a good set of feature extractor filters with less effort [32].

5.3. Employed Model Architectures

Convolutional neural networks (CNNs) are the main components powering the most state-of-the-art methods for classifying images [37]. With the employment of the filtering techniques adapted from signal processing applications, which are also employed in classical computer vision techniques, they are able to detect hidden or visible patterns in a given image or any kind of data which can be represented by 2-D matrices. However, classifying an image is only one of many scopes within the computer vision literature. In some areas, the accurate localization of the objects in a frame may be required for some tasks. Since CNNs are capable of extracting the patterns from the images [37], as an implicit side benefit, object detection, or region of interest determination from the images also become feasible achievements that can be accomplished through the application of these versatile deep learning models, which is precisely what was aimed in this implementation.

Increasing demand for real time object detection led to development of a variety of algorithms. Some of the most commonly used algorithms were RCNN based. However, for the sake of keeping the delay of the object detection pipeline low, one-shot detection algorithms were considered. YOLO (You Only Look Once) algorithm is one of the most recognized models for object detection [25,39].

YOLO is semi-flexible in the terms of choosing backbone networks. It generally consists of 2 parts, convolutional network for feature extraction and fully connected dense network for prediction. Freedom of backbone network selection comes with some perks like employing state-of-the-art models like ResNet. This enables training by transfer learning and ensures working on a good network architecture for a computer vision task [25]. YOLOv5[41] Algorithm is basically an improvement on the previous YOLO algorithms.

**Fig. 3.** Overview of the YOLOv5 architecture taken from [41]

During training Object Detectors Intersection Over Union is used to define positives and negatives. After training keeping same IoU threshold detectors tends to produce lots of noisy detections. This mainly happens because of overfitting during training, due to exponentially vanishing positive sample and inference-time mismatch between the IoUs for which the detector is optimal and those of the input hypotheses. Cascade R-CNN is proposed to overcome these issues. Object Detection algorithms rank great number of candidates using classification scores. But, using classification scores are not reliable. In VarifocalNet proposed a novel loss function Varifocal Loss to address this problem. One stage object detectors is commonly consists of classification and regression branches. But using two branch might cause spatial alignments. To address this problem in TOOD proposes novel architecture T-Head and alignment metric TAL. The details about this model can be obtained In Localization Distillation(LD) proposed a novel Knowledge Distillation method to efficiently transfer localization knowledge from the teacher to the student. Proposed algorithm can be applied to different object detectors without sacrificing inference speed. The details about these models can be obtained in [42,43,44,45,46].

5.4. Experiments

The obtained dataset was divided into training set (80%) and test set (20%). Images in the dataset that were spared for the training set was allowed to have augmented data. Validation set only consisted of the original acquired images, and augmented versions of these images were not used in the training set. The experiments consisted of comparing YOLOV5, Cascade R-CNN, VFNet, LD, TOOD, YOLOV3 and SSD300 algorithms.

**Table 1:** Object Tracking algorithms evaluations scores on our dataset. We chose the most common evaluation metrics. We collected mAP 0.5:0.95, mAP 0.5, mAP 0.75 scores. The first and second scores are marked as red and blue, respectively. In the YoloV5 repository we used, mAP 0.75 was not defined, so we filled that cell with X.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Detector** | **Backbone** | **mAP 0.5:095** | **mAP 0.5** | **mAP 0.75** |
| YOLOv5L | CSPDarknet | 0.683 | 0.979 | X |
| YOLOv5S | CSPDarknet | 0.631 | 0.979 | X |
| Cascade R-CNN | Resnet50 | 0.612 | 0.937 | 0.731 |
| VFNet | Resnet101 | 0.556 | 0.930 | 0.541 |
| TOOD | Resnet50 | 0.541 | 0.948 | 0.536 |
| VFNet | Resnet50 | 0.540 | 0.952 | 0.511 |
| LD | Resnet34 | 0.495 | 0.951 | 0.478 |
| LD | Resnet50 | 0.477 | 0.943 | 0.418 |
| LD | Resnet18 | 0.40 | 0.948 | 0.360 |
| YOLOv3 | Darknet-53 | 0.329 | 0.67 | 0.218 |
| SSD300 | VGG16 | 0.208 | 0.362 | 0.225 |

In this study, the chosen evaluation metrics for performance comparison are the widely used mAP0.5(mean Average Precision), mAP0.75 and mAP0.95.

As it can be seen from the Table 1, relatively old algorithms YOLOv3 and SSD300 were not able to adequately extract the discriminative features to achieve higher accuracy. Most probably, a larger dataset might help getting better results. However, gas stations may include different features and the region determination can be complicated due to the fact that some specific views of gas stations may include cars, a part of sky, a big sign, etc. In most cases, it can be observed that these variations increase the number of loose bounding boxes as shown in Fig. 3. This may be the main reason why YOLOv3 and SSD300 approach were not able to perform as well as expected.



**Fig. 3.** A correct localization with loose bounding box prediction.

Meanwhile, the YOLOv5 family performed much better than other algorithms. YOLOv5L achieved a 0.683 mAP0.5:0 and YOLOv5S achieved a 0.631 mAP0.5:0.95 score. We observed that a bigger backbone does not always increase the performance of the algorithm. For instance, LD with Resnet34 achieved a slightly better mAP score than LD with Resnet50. Increasing the number of training samples may change this observation. One of the fundamental reasons behind that outperformance is probably that YOLOv5 architecture(backbone, neck, and other components) can generalize with less data than other algorithms. Moreover, fast execution performance of YOLOv5 makes it a better choice among the alternatives for object detection applications that require real-time video processing. We also trained bigger models like YOLOX-l but, the models did not converge.

6. Discussions and future recommendations

Using the proposed approach to detect POI has several advantages. Google-Map, Apple Map and most mapping applications need to update their maps periodically. This process for periodic updating can take a substantial amount of time, it might be days, or sometimes weeks. As a result, solely depending on online maps and their corresponding POI databases might result in not fully up-to-date correspondences in real-time. A lot of users might have experienced recommendations for non-existent buildings, changed roads, or the online application might not be able to properly recommend the optimum output due to the lack of identifying a new route, new building, new store, etc. It would have been very beneficial to have a system that combines the best of both worlds: An online POI database which is periodically updated by the application server and an offline application that can detect POIs dynamically in real time.

The proposed POI detection approach finds new places and adds them to the conventional map applications instantly. Moreover, distributed mapping applications which run on next-generation driverless cars automatically update their maps and adjust their moves according to a new gas station or restaurant information. For example, cars can decide whether to buy gasoline by looking at the available gasoline and nearby gas station. Cars can also work as a meal recommender system. They propose potential places according to the locations of the restaurants and time of the day.

As it can be seen from the experiments, currently available and widely adopted state-of-the-art solutions are sufficiently powerful for tackling the problem. However, obtained data is far from sufficient for a good generalization for wide-range adaptation. POI features, brands or their appearances may tend to vary for other locations in different countries. So, the dataset can be enlarged by obtaining even more visuals from different locations.

Obtaining visuals and POIs is a time consuming task if it is performed manually. The data for such mapping applications require human power, mostly through crowdsourcing to acquire the necessary POI data through appropriate tagging of the content and the coordinates. These manual efforts are subject to errors; wrong information content can be tagged for the POI, or the coordinate information might not be accurate. At the same time, there is always the possibility of GPS signal loss or lack of Wireless/4G signal preventing the car to be able to locate itself and/or access the online map and POI database. A more robust POI data acquiring might be helpful.

Hence, instead of manually associating the POI information, the bounding box of the POI can be detected automatically using corners or edges. Currently the bounding boxes around gas station and restaurant logos are determined manually, as explained above. Instead, an automatic bounding can be generated heuristically. For instance, roadside image can be divided into sections like gas station pole, pavement or predefined logos. These image sections are exploited to determine the bounding box. Motion estimation can also be applied to estimate the location of the bounding box or a spatio-temporal method can be developed to generate the bounding box. Furthermore, the proposed approach only detects a single POI. The algorithm is improved to detect multiple POIs.

The only POI class considered in this paper was gas stations. This work can be easily extended by adding more classes. Furthermore, lots of POI classes tend to have similar properties with each other like having a sign. These properties may enable a hierarchical classification through detecting a parent class first, before detecting the child class. Hence, depending on the POI resolution, dynamic clusters of POIs can be analysed (i.e. restaurants, or specific restaurants such as Chinese Restaurants).

The learning mechanism of the proposed POI detection can be enhanced by integrating the information coming from other cars, GPS, base station internal sensors like LIDAR. This additional information can be exploited to improve the detection accuracy of the learning model.

Places like gas stations or restaurants are static point of interests. The proposed approach can be extended to detect dynamic point of interests like police cars, ambulance, fire truck, suspicious vehicles (drunk drivers). To achieve this aim, the proposed approach has to work faster. Hence, novel spatio-temporal POI detection mechanism should be developed. Moreover, sequential learning approaches like Recurrent Neural Network (RNN) / LSTM can be used to estimate the location of the vehicle.

Lastly different computation models can be used to mitigate the computation complexity. In addition to the computational capabilities of cars, edge computing and cloud computing can be used to leverage the computational capabilities of centralized systems. Distributed computing can also be used. For instance, nearby cars share computation resources to distribute the complexity of the learning algorithms.

7. Conclusions

In this study, the POI detection problem within the car surveillance videos is addressed. This particular problem is a major concern for providing the best Quality of Service (QoS) for autonomous vehicles. Most applications rely on online maps for POI recommendation. However, these maps sometimes may be out-of-synch or unreachable and the recommendation system might not able to recommend the best POI under these circumstances. Various techniques used throughout the literature are analysed including traditional image processing techniques along with new state-of-the-art deep learning models. Careful analysis of these algorithms indicate the deep learning models, especially the CNN based region detection techniques handily outperform traditional methods. However, the computational complexity is also high in these deep learning models. Among the CNN based models, YOLO provides real-time execution opportunities with remarkably accurate prediction performance. For demonstrating the effectiveness of YOLO in POI detection problem, in this study, a gas station detection model based on YOLO trained and tested with a real highway video footage is adopted. 0.979 mAP0.6 score is achieved with this POC model. The results indicate that it might be possible to use such a POI detection system in real time without the explicit need for GPS access and an online POI database.

Acknowledgements

The authors would like to thank the General Directorate of Highways.

Compliance with Ethical Standards

**Conflict of interest** No potential conflict of interest was reported by the authors.

References

1. Sreenu, G., Saleem Durai, M.A. "Intelligent video surveillance: a review through deep learning techniques for crowd analysis", Journal of Big Data, 6, 48, 2019. <https://doi.org/10.1186/s40537-019-0212-5>
2. Tsakanikas, V., Dagiuklas, T. "Video surveillance systems-current status and future trends", Computers & Electrical Engineering, 70, pp. 736–753, 2018. <https://doi.org/10.1016/j.compeleceng.2017.11.011>
3. Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H., Herrera, F. "Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance", Knowledge-Based Systems, 194, 105590, 2020. <https://doi.org/10.1016/j.knosys.2020.105590>
4. Senthil Murugan, A., Suganya Devi, K., Sivaranjani, A., Srinivasan, P. "A study on various methods used for video summarization and moving object detection for video surveillance applications", Multimedia Tools and Applications, 77(18), pp. 23273–23290, 2018. <https://doi.org/10.1007/s11042-018-5671-8>
5. Chehri, A., Mouftah, H.T. "Autonomous vehicles in the sustainable cities, the beginning of a green adventure", Sustainable Cities and Society, 51, 101751, 2019. <https://doi.org/10.1016/j.scs.2019.101751>
6. Vosooghi, R., Puchinger, J., Jankovic, M., Vouillon, A. "Shared autonomous vehicle simulation and service design", Transportation Research Part C: Emerging Technologies, 107, pp. 15–33, 2019. <https://doi.org/10.1016/j.trc.2019.08.006>
7. Ahmad, T., Ilstrup, D., Emami, E. Bebis, G. "Symbolic road marking recognition using convolutional neural networks", In: 2017 IEEE Intelligent Vehicles Symposium (IV), 2017. <https://doi.org/10.1109/IVS.2017.7995910>
8. Wu, T. Ranganathan, A. "A practical system for road marking detection and recognition", In: 2012 IEEE Intelligent Vehicles Symposium, 2012. <https://doi.org/10.1109/IVS.2012.6232144>
9. Bailo, O., Lee, S., Rameau, F., Yoon, J.S., Kweon, I.S. "Robust road marking detection and recognition using density-based grouping and machine learning techniques", In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017. <https://doi.org/10.1109/WACV.2017.90>
10. Ruta, M., Scioscia, F., De Filippis, D., Ieva, S., Binetti, M., Di Sciascio, E. "A Semantic-enhanced augmented reality tool for OpenStreetMap POI discovery", Transportation Research Procedia, 3, pp. 479–488, 2014. <https://doi.org/10.1016/j.trpro.2014.10.029>
11. Yu, Q., Jiang, H., Liu, C., Wu, M. "The application of data mining in multi-supplier points of interest processing", In: 2013 Ninth International Conference on Natural Computation (ICNC), 2013. <https://doi.org/10.1109/ICNC.2013.6818119>
12. Hao, J., Wang, G., Seo, B., Zimmermann, R. "Point of interest detection and visual distance estimation for sensor-rich video", IEEE Transactions on Multimedia, 16(7), pp. 1929–1941, 2014. <https://doi.org/10.1109/TMM.2014.2330802>
13. Rohella, A. Singh, S. "Path independent real time points of interest detection in road networks", In: 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016. <https://doi.org/10.1109/IC3I.2016.7918040>
14. Shu, Z., Xin, S., Xu, X., Liu, L., Kavan, L. "Detecting 3D points of interest using multiple features and stacked auto-encoder", IEEE Transactions on Visualization and Computer Graphics, 25(8), pp. 2583–2596, 2019. <https://doi.org/10.1109/TVCG.2018.2848628>
15. Zhao, Z., Zheng, P., Xu, S., Wu, X. "Object detection with deep learning: A review", IEEE Transactions on Neural Networks and Learning Systems, 30(11), pp. 3212–3232, 2019. <https://doi.org/10.1109/TNNLS.2018.2876865>
16. Hendry, Chen, R.-C. "Automatic license plate recognition via sliding-window darknet-YOLO deep learning", Image and Vision Computing, 87, pp. 47–56, 2019. <https://doi.org/10.1016/j.imavis.2019.04.007>.
17. Baca, T., Stepan, P., Spurny, V., Hert, D., Penicka, R., Saska, M., Thomas, J., Loianno, G., Kumar, V. "Autonomous landing on a moving vehicle with an unmanned aerial vehicle", Journal of Field Robotics, 36(5), pp. 874–891, 2019. <https://doi.org/10.1002/rob.21858>
18. Yaqoob, I., Khan, L.U., Kazmi, S.M.A., Imran, M., Guizani, N., Hong, C.S. "Autonomous driving cars in smart cities: Recent advances, requirements, and challenges", IEEE Network, 34(1), pp. 174–181, 2020. <https://doi.org/10.1109/MNET.2019.1900120>
19. Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., Pollefeys, M. "Leveraging 3D city models for rotation invariant place-of-interest recognition", International Journal of Computer Vision, 96(3), pp. 315–334, 2012. <https://doi.org/10.1007/s11263-011-0458-7>
20. Zhang, M., Li, W., Du, Q. "Diverse region-based CNN for hyperspectral image classification", IEEE Transactions on Image Processing, 27(6), pp. 2623–2634, 2018. <https://doi.org/10.1109/TIP.2018.2809606>
21. Ren, S., He, K., Girshick, R., Sun, J. "Faster R-CNN: Towards real-time object detection with region proposal networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), pp. 1137–1149, 2017. <https://doi.org/10.1109/TPAMI.2016.2577031>
22. Girshick, R., Donahue, J., Darrell, T., Malik, J. "Rich feature hierarchies for accurate object detection and semantic segmentation", In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014.
23. Zhao, X., Jia, H., Ni, Y. "A novel three-dimensional object detection with the modified You Only Look Once method", International Journal of Advanced Robotic Systems, 15(2), 2018. <https://doi.org/10.1177/1729881418765507>
24. Ćorović, A., Ilić, V., Ðurić, S., Marijan, M., Pavković, B. "The real-time detection of traffic participants using YOLO algorithm", In: 2018 26th Telecommunications Forum (TELFOR), 2018. <https://doi.org/10.1109/TELFOR.2018.8611986>
25. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. "You Only Look Once: Unified, real-time object detection", In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.
26. Li, H., Feng, M., Wang, X. "Inverse perspective mapping based urban road markings detection", In: 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, 2012. <https://doi.org/10.1109/CCIS.2012.6664569>
27. Greenhalgh, J., Mirmehdi, M. "Automatic detection and recognition of symbols and text on the road surface", in Pattern Recognition: Applications and Methods, 2015. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-27677-9_8>
28. Kheyrollahi, A., Breckon, T.P. "Automatic real-time road marking recognition using a feature driven approach", Machine Vision and Applications, 23(1), pp. 123–133, 2012. <https://doi.org/10.1007/s00138-010-0289-5>
29. Lee, Y., Lee, J., Hong, Y., Ko, Y., Jeon, M. "Unconstrained road marking recognition with generative adversarial networks", In: 2019 IEEE Intelligent Vehicles Symposium (IV), 2019. <https://doi.org/10.1109/IVS.2019.8814057>
30. Lowe, D.G. "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60(2), pp. 91–110, 2004. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
31. Wolpert, D.H., Macready, W.G. "Coevolutionary free lunches", IEEE Transactions on Evolutionary Computation, 9(6), pp. 721–735, 2005. <https://doi.org/10.1109/TEVC.2005.856205>
32. Pratt, L.Y. "Discriminability-based transfer between neural networks", in Advances in Neural Information Processing Systems 5, 1992. Morgan Kaufmann Publishers Inc.
33. ImageNet Challenge, <http://www.image-net.org/challenges/LSVRC/> (cited 2021 November 14).
34. ResNet50, <https://keras.io/api/applications/resnet/#resnet50-function> (cited 2021 December 18).
35. Pascanu, R., Mikolov, T., Bengio, Y. "On the difficulty of training recurrent neural networks", In: International conference on machine learning (pp. 1310-1318). PMLR.
36. Deep Residual Network, <http://primo.ai/index.php?title=(Deep)_Residual_Network_(DRN)_-_ResNet> (cited 2021 December 22).
37. Venkatesan, R., Li, B., "Convolutional neural networks in visual computing: a concise guide", 2017, CRC Press.
38. Girshick, R. "*Fast R-CNN"*, In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), December 2015, pp. 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
39. Zhou, S., Bi, Y., Wei, X., Liu, J., Ye, Z., Li, F., Du, Y. "Automated detection and classification of spilled loads on freeways based on improved YOLO network", Machine Vision and Applications, 32(2), 44, 2021. <https://doi.org/10.1007/s00138-021-01171-z>
40. Kwan, C., Chou, B., Yang, J., Rangamani, A., Tran, T., Zhang, J., Etienne-Cummings, R. "Target tracking and classification using compressive sensing camera for SWIR videos". Signal, Image and Video Processing, 13(8), pp. 1629–1637, 2019. <https://doi.org/10.1007/s11760-019-01506-4>

[41] YOLOv5 <https://github.com/ultralytics/yolov5> (cited 2022 May)

[42] Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[43] Zhang, Haoyang, et al. "Varifocalnet: An iou-aware dense object detector." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[44] Zheng, Zhaohui, et al. "Localization Distillation for Dense Object Detection." CVPR, 2022.

[45] Feng, Chengjian, et al. "Tood: Task-aligned one-stage object detection." 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, 2021.

[46] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.