



Vacant Parking Lot Information System Using Transfer Learning and IoT

Edwin K. Jose* & Shanmugham Veni

Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

*E-mail: edjoz1010@gmail.com

Abstract. Parking information systems have become very important, especially in metropolitan areas as they help to save time, effort and fuel when searching for parking. This paper offers a novel low-cost deep learning approach to easily implement vacancy detection at outdoor parking spaces with CCTV surveillance. The proposed method also addresses issues due to perspective distortion in CCTV images. The architecture consists of three classifiers for checking the availability of parking spaces. They were developed on the TensorFlow platform by re-training MobileNet (a pre-trained Convolutional Neural Network (CNN)) model using the transfer learning technique. A performance analysis showed 88% accuracy for vacancy detection. An end-to-end application model with Internet of Things (IoT) and an Android application is also presented. Users can interact with the cloud using their Android application to get real-time updates on parking space availability and the parking location. In the future, an autonomous car could use this system as a V2I (Vehicle to Infrastructure) application in deciding the nearest parking space.

Keywords: *Android application; CCTV surveillance; CNN; IoT; MobileNet; outdoor parking; TensorFlow; transfer learning; V2I.*

1 Introduction

Vehicle parking systems have become an important part in modern urban planning. Land earmarked for parking itself is economically unproductive [1]. Besides, it is estimated that drivers spend a considerable amount of time [2] in search of parking (blind search), leading to traffic congestion and fuel wastage. Hence, the benefits of smart parking solutions [3] are twofold: offering better parking space utilization and cutting down blind search.

Vision based systems with CCTV surveillance are scalable and cost-effective for large outdoor parking spaces. They have been used for multiple applications, such as occupancy detection, vehicle type detection [4,5] and theft prevention [6]. The existing methods for occupancy detection (finding the presence or absence of cars at parking lots) employ a two-part strategy: extracting relevant features from camera images (obtained from the parking field) and training a

classifier. Features such as color, shape and texture of cars or the ground surface are commonly used. The latest approach is to apply deep learning techniques. While these methods provide satisfying results, practical implementation is not very easy. For example, we have to create a 3D model of the whole parking space to use the solution proposed by Huang, *et al.* [7]. Building a deep learning classifier [8] requires extreme training and tuning. Yet, a model developed for one parking field may not be portable to another with the same level of accuracy.

This work proposes a low-cost, easy-to-implement, vision-based detection-cum-guidance system for outdoor parking spaces. The low cost is achieved by using existing surveillance camera feed instead of deploying extra sensors. Transfer learning is proposed for easy implementation. This is a branch of deep learning wherein a new model is developed from a pre-trained model. An analogy with real life is that if you have already mastered riding a bicycle, it will be easy to learn to drive a motorbike [9]. This can be thought of as transferring the existing knowledge to solve a new but related problem. From a vision-based-solution point of view, major issues (noise) with outdoor parking spaces are [7]: (i) shadow, (ii) perspective distortion of the ground scene, (iii) partial occlusion of parked cars by other cars or walking humans, (iv) varying illumination and climatic conditions from day to night. It can also be observed that we lose a clear view of individual parking spaces as we move away from a direct camera view (from ROI-5 to ROI-1 in Figure 2(b)). Hence, instead of processing the entire parking field information at once a more resilient method is to divide the whole CCTV image frame into multiple regions of interest (ROI). This can be a single parking space or a group of parking spaces. ROI selection in this work is based on the amount of subjective perspective distortion and partial occlusion (Figure 2(b)). Individual classifiers were developed to take care of different occupancy conditions and noises for each ROI. Occupancy detection is a binary classification problem. All we want to detect is the presence or absence of cars in each of the ROI. Overall vacant space count in the current frame is found by adding the results from each of the ROI. The count from multiple frames is taken and the most repeated count is posted to the cloud. Additionally, an Android application was developed for real time guidance. The application helps users to get the count of vacant parking spaces and the location for parking.

The rest of the paper is organized as follows: Section 2 provides a detailed review of related works, Section 3 describes the proposed system's architecture and methodology. Section 4 presents a performance analysis to estimate the accuracy of the proposed method. Finally, the conclusions are discussed in Section 5.

2 Related Works

Almeida, *et al.* [10] created a large and robust dataset for parking lots classification called PKlot. It has images taken from multiple parking fields with different weather conditions (sunny, overcast and rainy). They used textural-based descriptors for the classification of parking spaces. Huang, *et al.* [7] provided a plane-based method that adopts a structural 3D parking lot model consisting planar surfaces. The method divides the 3D space around a parking space into the faces of a cuboid and conducts classification on each those face patches. For example, if the ground patch is taken, an occupied parking space would feature car tires but a vacant parking space would give the texture of the ground itself. The 3D modeling helps to avoid inter-object occlusion and perspective distortion. Al-Kharusi, *et al.* [11] proposed a direct image processing approach for occupancy detection. The parking spaces are marked with green dots. Occupancy is detected when the green dots become invisible due to a parked car. Postigo, *et al.* [12] suggested a computationally efficient method for parking area estimation. Their method is to perform background subtraction and create a transience map, which in turn will be used for occupancy detection. Ling, *et al.* [13] created an overall system that can learn the topology of parking lots as well as detect the occupancy of identified parking spaces. The algorithm learns the parking lot from the location where parking events occur over time. Fraifer, *et al.* [14] proposed a design approach for utilizing CCTV images. Using a simulated parking environment, they developed an end-to-end IoT model including an Android application. The authors suggested real-time implementation as a future work. Amato, *et al.* [8] developed their own dataset (CNRPark) and a CNN based deep learning solution. They manually created a mask to segment out each parking space. The segmented image is given to a CNN classifier to find out occupancy status.

Szegedy, *et al.* [15] introduced a deep learning model called Inception. It was trained with ImageNet dataset [16] and capable of classifying 1000 classes. Xia, *et al.* [9] applied transfer learning on Inception-v3 [17] for flower classification. Gogul *et al.* [18] combined CNN with transfer learning to obtain better accuracy for flower classification. Wang, *et al.* [5] used deep transfer learning for vehicle type recognition from surveillance images. Training data were generated from labeled web images of different vehicle types. Their classifier architecture is similar to AlexNet [19] with fine-tuning applied to the last two fully connected (FC) layers. Transfer learning has been used in other image classification problems, for accelerated Magnetic Resonance Imaging (MRI) [20], lung pattern analysis [21] and Synthetic Aperture Radar (SAR) target recognition in remote sensing [22]. Howard, *et al.* [23] introduced the MobileNet CNN model. This is an efficient and lightweight pre-trained deep neural net that uses depth-wise separable convolution to reduce computation at its initial layers. Unlike

traditional deep nets, MobileNets are capable of conducting recognition tasks in a timely fashion that is suitable for mobile and embedded vision applications.

3 Methodology

3.1 Proposed System Architecture

Figure 1 depicts the system architecture of the proposed method. It consists of two sub-modules: one for vacant parking space detection and another for giving parking guidance to end users. The faculty parking lot (Figure 2(a)) at our institution was chosen for the proposed work. A 700 TVL CP-plus analog camera was mounted at the parking field. The DVR (digital video recorder) provides 960×576 RGB images at 25 fps. There are 8 parking spaces that are in clear view of the camera.

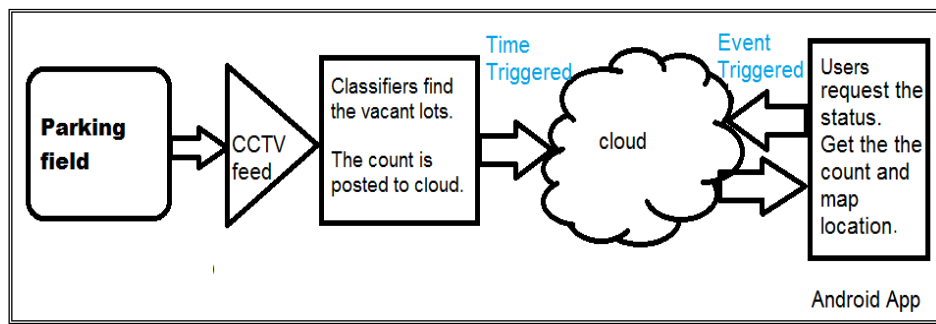


Figure 1 System architecture.

The workflow starts with cropping the captured CCTV frame into five sets of ROI (Figure 2(b)). Then the ROI patches are given to three different classifiers: (i) for ROI-1, (ii) for ROI-2 and (iii) for ROI- 3, 4, 5. This ‘divide-and-conquer’ approach gives better resilience against issues such as partial occlusion and perspective distortion. The respective classifiers evaluate the number of vacant parking spaces in the corresponding ROI patch.

The results from the three classifiers are added and stored. This process is repeated for eleven upcoming frames. Defining such an observation window helps to avoid occlusion of a parking space by moving cars or walking humans. Among the eleven stored values, the most repeated count is posted to the cloud (parking field location is updated once during initial set-up). The next cycle of the observation window begins thereafter. Using an Android application, users get both a vacant parking space count and the parking location from the cloud.

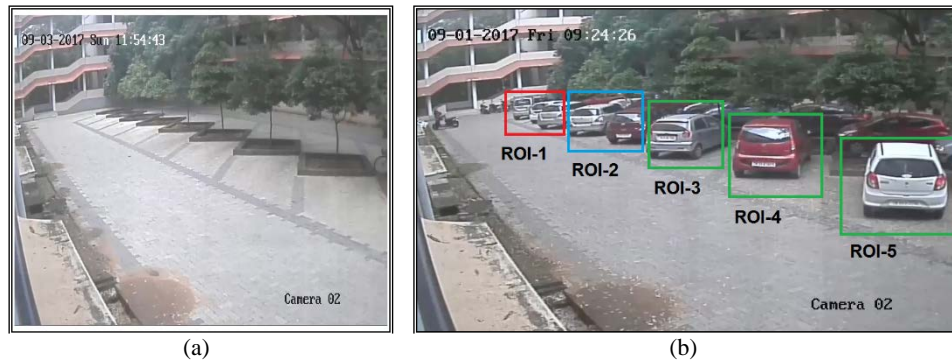


Figure 2 (a) Faculty parking field at the institution. Eight parking spaces are visible. (b) Different levels of perspective distortion in the CCTV image. ROI-1 (marked in red) denotes the highest distortion. ROI-5 (marked in green) has the least.

The following sections elaborate the design details of the classifiers, the cloud platform and the Android application.

3.2 Transfer Learning: An Overview

Pang & Yang [24] define transfer learning as follows: “Given a source domain D_S and learning task T_S , a target domain D_T and learning task T_T , transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S , where $D_S \neq D_T$, or $T_S \neq T_T$.”

In machine learning tasks this can be seen as learning from a pre-trained base model to solve a related problem. For natural image classification, a deep CNN model (AlexNet, Inception, etc.) trained with a large dataset (such as ImageNet) can be taken as the base model. The initial layers of the CNN are constituted by stacking up convolutional ReLU (Rectified Linear Unit) and pooling structures. These layers are responsible for feature extraction. For classification, fully connected (FC) layers followed by a Softmax layer are used.

The features extracted by the initial layers are ‘general’ for any image dataset, whereas the final layer’s extract features are ‘specific’ to the chosen dataset and task [25]. Hence, we can use CNN as an ‘off-the-shelf’ feature extractor for a new dataset to train the target classifier [26]. To avoid overfitting on small datasets, the initial layers of the base model are frozen [25]. Using extracted features from the frozen structure, we train only the last few FC layers (of the base model) with our own class labels. This can be thought of as transferring knowledge of the pre-trained model for the new classification task. The crucial part in transfer learning is collecting sufficient data for the new classification task.

3.3 Dataset Collection

The recorded video footage from the DVR across multiple days was collected to create the dataset. The video footage was split into frames and stored. The image dataset was created considering the following scenarios at outdoor parking spaces:

1. Images for all climatic conditions – sunny, overcast, rainy, shadowy (from trees, building), different illumination cases from morning to evening.
2. Images where one parking space is partially occluded by another car.

The next task is defining the ROI for 3 classifiers as described in Section 3.4. Three datasets were created by cropping ROI patches from the frames. The coordinates for cropping were found out manually using Matlab.

3.4 ROI and Classifier Labels

There were 8 parking spaces in the parking field (Figure 2(a)). Since the CCTV camera was fixed, the coordinates of the parking spaces in every frame are the same. Masks were created to segment out the 8 parking spaces into 5 patches (Figure 2(b)). The occupancy status of a parking space is either ‘free’ or ‘busy’ (class label 0 or 1).

ROI-1: contains last three parking spaces, which are the farthest from the CCTV camera. For these 3 parking spaces there are 8 occupancy classes (000 to 111).

ROI-2: contains the middle two parking spaces, hence, there are 4 class labels (00 to 11).

ROI-3 to ROI-5: respectively contain the 3 parking spaces that are closest to the CCTV camera. Each has 2 output classes (0 or 1).

The advantage of this kind of labeling is that we can specifically understand the occupancy status (free or busy) of each of the parking spaces.

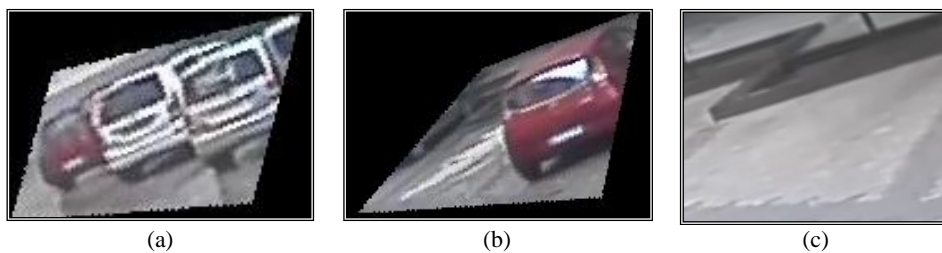


Figure 3 (a) ROI-1 with class label ‘111’, (b) ROI-2 with class label ‘01’, (c) empty parking space with class label ‘0’ in ROI-3/4/5.

Table 1 Sample set of class labels and their meaning for ROI-1 (Figure 3(a)).

Class Label	Occupancy Status		
	Lot1 (left)	Lot2 (middle)	Lot3 (right)
000	free	free	free
101	busy	free	busy
110	busy	busy	free

3.5 Training the Classifiers Using MobileNet

TensorFlow [27] was used for training the classifiers. Figure 4 depicts the training procedure for the ROI-1 dataset using MobileNet. This is a light-weight CNN model from Google, trained with the ImageNet dataset, capable of classifying 1000 classes. It uses depth-wise separable convolution at the initial layers to reduce the number of computations and training parameters to produce a small-sized model (around 5MB) at the cost of reduced accuracy. This makes MobileNets suitable for embedded platforms like Raspberry Pi. For the proposed method, MobileNet_v1_1.0_224 was used. The input for this model are 224×224 RGB images and it achieved a maximum accuracy of 70.7% on the ImageNet dataset. As the ImageNet contains many car images and MobileNet has learned to understand the features of cars, it is a suitable choice for parking space classification tasks.

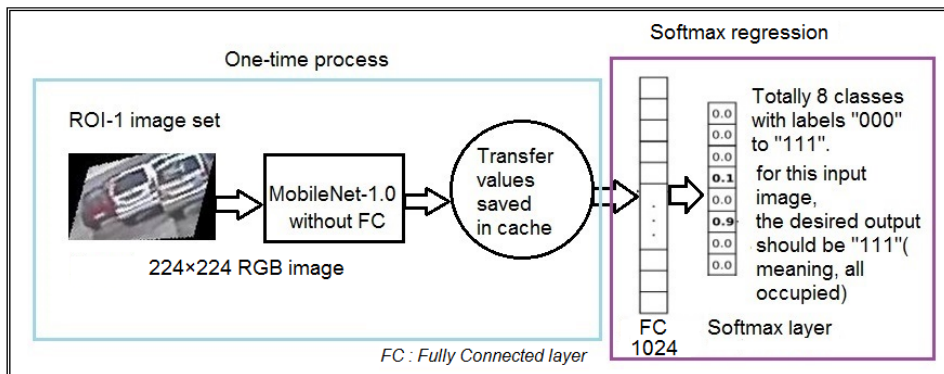


Figure 4 Schematic of transfer learning with MobileNet (using TensorFlow).

The transfer learning process using TensorFlow starts with freezing the initial layers of the base model (MobileNet). When the new image dataset is given to this frozen structure, it outputs ‘transfer values’ (called ‘bottlenecks’ in TensorFlow terminology) for each image. These values will be cached in the system memory. This is a one-time process. Transfer values represent the features extracted from new images. The fully connected layer (of the MobileNet) undergoes supervised learning using these transfer values as input and the ground truth label of the new dataset as the desired output. Ten-fold

cross validation [28] was used to construct the training dataset and the testing dataset. Softmax regression with stochastic gradient descent (to minimize cross entropy) was used for training [28,29]. Training was conducted on a 64-bit Windows 7 machine with Intel i5 CPU with 4 GB of RAM. Table 2 summarizes the details of the training.

Table 2 Details of training.

Classifier	Labels	Avg. Number of Images per Class	Number of Training Steps	Training Accuracy
Classifier 1 (for ROI-1)	8 labels (000 to 111)	300	500	99%
Classifier 2 (for ROI-2)	4 labels (00 to 11)	280	500	99%
Classifier 3 (for ROI-3,4,5)	2 labels (0 or 1)	250	500	99%

3.6 Cloud Platform

ThingSpeak (<https://thingspeak.com>) was used for developing the IoT part of the proposed work. This is a MATLAB-owned IoT platform that provides a free evaluation service for small non-commercial projects. The platform gives RESTful (Representational State Transfer) web service with the HTTP (Hypertext Transfer Protocol) POST and GET methods. These methods can be directly implemented using MATLAB commands. The ‘posted’ data (vacant parking space count) are saved to the owner’s channel (Figure 5(a)) in the ThingSpeak cloud.

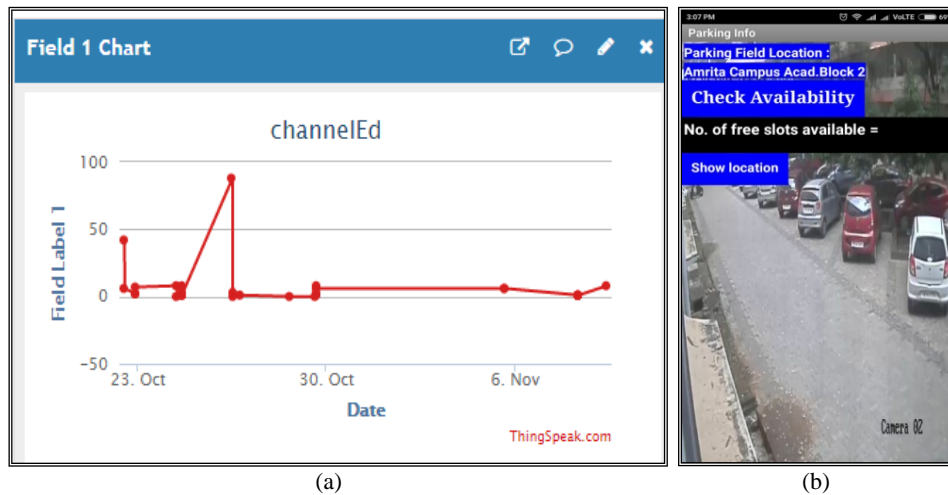


Figure 5 (a) Line graph showing update history to the channel. (b) Android application user interface.

3.7 Android Application

The Android application helps users to enquire the current vacant parking space count and parking field location. MIT's App Inventor was used to develop the application (Figure 5(b)). The below functionalities were implemented:

1. Enquiry of vacant parking spaces: an HTTP GET request will be initiated. The response value (the count) from the cloud will be displayed in the application.
2. Enquiry of location: an activity will be started for invoking the Google Maps application on the phone. When Google Maps launches, it displays the location of the parking field.

The application was implemented and tested successfully on a Xiaomi Redmi Note 4 smartphone running Android 7.0.

3.8 Sample Workflow

When a new test frame comes in, it will be given to ROI extraction. Then, the classifiers for each of the ROI will be called. It outputs the classes with different Softmax scores. The class with the highest score is taken as the desired output. The output label of each classifier is a string containing 1s and 0s. Outputs from 3 classifiers will be assimilated into an array of size 8 (corresponding to 8 parking spaces). For example, if the input frame is Figure 2(a), the combined output from three classifiers should (ideally) be the string **00000000** and for figure 2(b) it should be **11111111**. The number of vacant parking spaces is calculated by counting the number of zeros in this array. This count will be stored in another array. The process will be repeated for 11 upcoming frames and most the repeated number among the eleven counts will be uploaded to the cloud.

4 Performance Evaluation

To test the accuracy of the proposed method, an image repository was created. It contained 100 images corresponding to 9 states of occupancy (0 = all occupied to 8 = all vacant). The images were CCTV frames from multiple days with different levels of illumination. Table 3 summarizes the confusion matrix. Accuracy was estimated at 88%. As transfer learning is essentially a data driven approach, the accuracy can be improved by providing the classifiers with more image samples.

$$\text{Accuracy} = (\text{true positives} + \text{true negatives}) / \text{Total} = 795 / 900 = 88.34\%.$$

Table 3 Confusion matrix.

Actual number of vacant parkings spaces	Predicted number of vacant parking spaces								
	0	1	2	3	4	5	6	7	8
0	92	6	2	0	0	0	0	0	0
1	5	87	7	1	0	0	0	0	0
2	0	13	80	7	0	0	0	0	0
3	0	0	8	83	6	3	0	0	0
4	0	0	2	10	82	6	0	0	0
5	0	0	0	3	6	87	4	0	0
6	0	0	0	0	2	5	93	0	0
7	0	0	0	0	0	0	9	91	0
8	0	0	0	0	0	0	0	0	100

5 Conclusions

Although existing methods in the literature have provided satisfactory solutions to vacant parking space detection from surveillance images, its practical implementation is laborious and time-consuming in terms of extracting the right features and training the classifier. This work addressed these issues by using the transfer learning technique. The method automatically extracts suitable features from a new dataset using a pre-trained CNN model in order to train a simple classifier. In this paper, transfer learning was conducted using MobileNet followed by Softmax classification. Problems due to varying illumination conditions, perspective distortion and partial occlusion were addressed by dividing the CCTV frame into different regions. Datasets for each of the regions were created from the recorded CCTV video available in the DVR. Then multiple classifiers were trained for each of the regions. An end-to-end application model for parking guidance system with IoT and an Android application was also presented. The accuracy of the proposed method was estimated to be 88%. While the proposed method is suitable for small parking spaces, its generalization for large parking fields needs to be tested.

References

- [1] Roychowdhury, A. & Nasim, U., *Parking Policy for Clean Air & Liveable Cities: A Guidance Framework*, Centre for Science and Environment, New Delhi, India, 2016.
- [2] Dowling, C., Fiez, T., Ratliff, L. & Zhang, B., *How Much Urban Traffic is Searching for Parking*, arXiv:1702.06156 [cs.CY], Submitted to applications track to KDD 2017.
- [3] Faheem S.A., Mahmud G.M., Khan M. & Zafar, R.H., *A Survey of Intelligent Car Parking System*, Journal of Applied Research and Technology, **11**(5), pp. 714-726, 2013.

- [4] Murugesan, G., Vasudevan, S.K., Ramachandran, S., Vasudevan, S. & Arumugam, B., *Vehicle Identification using Fuzzy Adaline Neural Network*, Journal of Computer Science, **9**(6), pp. 757-762, 2013.
- [5] Wang, J., Zheng, H., Huang, Y. & Ding, X., *Vehicle Type Recognition in Surveillance Images from Labeled Web-Nature Data Using Deep Transfer Learning*, IEEE Transactions on Intelligent Transportation Systems, **PP** (99) (IEEE Early Access), pp. 1-10, 2017.
- [6] Sreedevi, A.P. & Nair, B.S.S., *Image Processing Based Real Time Vehicle Theft Detection and Prevention System*, in Proceedings of International Conference on Process Automation, Control and Computing (PACC), 2011.
- [7] Huang, C.C. & Vu, H.T., *Vacant Parking Space Detection Based on a Multilayer Inference Framework*, IEEE Trans. CSVT, **27**(9), pp. 2041-2054, 2017.
- [8] Amato, G., Carrara, F., Falchi, F., Gennaro, C. & Vairo, C., *Car Parking Occupancy Detection Using Smart Camera Networks and Deep Learning*, in IEEE Symposium on Computers and Communication (ISCC), pp. 1212-1217, 2016.
- [9] Xia, X., Xu, C. & Nan, B., *Inception-v3 for Flower Classification Image*, in 2nd International Conference on Vision and Computing (ICIVC), pp. 783-787, 2017.
- [10] Almeida, P.R. de, Oliveira, L.S., Britto, A.S., Silva, E. J. & Koerich, A.L., *Pklot – A Robust Dataset for Parking Lot Classification*, Expert Systems with Applications, **42**(11), pp. 4937-4949, 2015.
- [11] Al-Kharusi, H. & Al-Bahadly, I., *Intelligent Parking Management System Based on Image Processing*, World Journal of Engineering and Technology, **2**(2), pp. 55-67, 2014.
- [12] Postigo, C.G. del, Torres, J. & Menéndez, J.M., *Vacant Parking Area Estimation Through Background Subtraction and Transience Map Analysis*, IET Intelligent Transport Systems, **9**(9), pp. 835 -841, 2015.
- [13] Ling, X., Sheng, J., Baiocchi, O., Liu, X. & Tolentino, M.E., *Identifying Parking Spaces & Detecting Occupancy Using Vision-Based IoT Devices*, in Global Internet of Things Summit (GIoTS), pp. 1-6, 2017.
- [14] Fraifer, M. & Fernström, M., *Smart Car Parking System Prototype Utilizing CCTV Nodes: A Proof of Concept Prototype of a Novel Approach Towards IoT-Concept Based Smart Parking*, in IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 649-654, 2016.
- [15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A., *Going Deeper with Convolutions*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015.
- [16] Jia, J., Dong, D.W., Socher, R., Li, Kai Li, L.J. & Fei-Fei, L., *ImageNet: A Large-Scale Hierarchical Image Database*, in IEEE Conference on

- Computer Vision and Pattern Recognition, pp. 248-255, 2009.
<http://image-net.org/>
- [17] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z., *Rethinking the Inception Architecture for Computer Vision*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818-2826, 2016.
- [18] Gogul, I. & Sathiesh Kumar, V., *Flower Species Recognition System Using Convolution Neural Networks and Transfer Learning*, in Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1-6, 2017.
- [19] Krizhevsky, A., Sutskever, I. & Hinton, G.E., *ImageNet Classification with Deep Convolutional Neural Networks*, in Proc. Neural Inf. Process. Syst. (NIPS), pp. 1097-1105, 2012.
- [20] Dar, S.U.H. & Çukur, T., *A Transfer-Learning Approach for Accelerated MRI using Deep Neural Networks*, arXiv:1710.02615v2 [cs.CV], 2017.
- [21] Christodoulidis, S., Anthimopoulos, M., Ebner, L., Christe, A. & Mouggiakakou, S., *Multisource Transfer Learning with Convolutional Neural Networks for Lung Pattern Analysis*, IEEE Journal of Biomedical and Health Informatics, **21**(1), pp. 76-84, 2017.
- [22] Huang, Z., Pan, Z. & Lei, B., *Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data*, Remote Sensing MDPI, **9**(9), pp. 1-21, 2017.
- [23] Howard Andre, G., Menglong, X., Bo, C., Dmitry, K., Zeijun, W., Tobias, W., Marco, A. & Hartwig, A., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, arXiv:1704.04861v1[cs.CV], 2017. (The model can be downloaded from <https://research.googleblog.com/2017/06/mobilenets-open-source-models-for.html>. (Last accessed:01-Dec-2017)
- [24] Pan, S.J. & Yang, Q., *A Survey on Transfer Learning*, **22**(10), pp. 1345-1359, 2010.
- [25] Yosinski, J., Clune, J., Bengio, Y. & Lipson, H., *How Transferable are Features in Deep Neural Networks*, in Proc. Neural Inf. Process. Syst. (NIPS), pp. 3320-3328, 2014.
- [26] Razavian, A.S., Azizpour, H., Sullivan, J. & Carlsson, S., *CNN Features Off-The-Shelf: An Astounding Baseline for Recognition*, Proc. 2014 IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pp. 512-519, 2014.
- [27] Abadi, M. & Agarwal, A., *TensorFlow: Large-scale Machine Learning on Heterogeneous Distributed Systems*. arXiv:1603.04467v2[cs.DC], 2016.
- [28] Alpaydin, E., *Introduction to Machine Learning*, 3rd ed., The MIT Press, Massachusetts, 2014.
- [29] Goodfellow, I., Bengio, Y. & Courville, A., *Deep Learning*, The MIT Press, Massachusetts, 2016.