An evaluation of a lightweight camera-based smart parking system made with a simple convolutional neural network

Maxim Kostin
Innopolis University
m.kostin@innopolis.ru

Alena Yuryeva
Innopolis University
a.yuryeva@innopolis.ru

April 10, 2018

Abstract

In this paper, we compare a novel simplistic low-cost smart parking system to state-of-the-art solutions involving SVM and HOG. The system is evaluated by the accuracy of predictions, as well as by speed, memory consumption, and usability. We propose use cases and ways of future development for the system.

An evaluation of a lightweight camera-based smart parking system made with a simple convolutional neural network

I. Introduction

Nowadays, the concept of the smart city has been in trend for several years, with large cities like Las Vegas trying out to implement the concept in real life. An ambitious task, this involves many adjustments as compared to the traditional way of life. Thoughtful implementation of smart city systems would include extensive secure information exchange between all subsystems, new standards, massive introduction of smart houses into the city infrastructure, in-depth analytics and flexible centralized logistics management, along with educating citizens about the new ways of life.

As a means of material communications within the city, transportation plays a crucial role in such a plan, requiring a full-blown infrastructure of intelligent transport systems, aimed to solve major problems in the industry. One of such problems is an issue of parking lots. In crowded cities, it is often difficult to find an empty parking slot, which makes some people avoid crowdy places. Merchants lose in revenue, and so does the government.

There has been a substantial amount of research performed to be able to determine parking lot occupancy. All solutions presented in academia can be thus divided into two broad categories: sensor-based systems and camera-based systems.

Thinking on global issues, we opted to concentrate on camera-based solutions which are more affordable and thus can be used more widely in developing countries. These solutions only require a camera to grab an image and a computer to run the system on, while on the other hand requiring more extensive data processing with use of computer vision technologies.

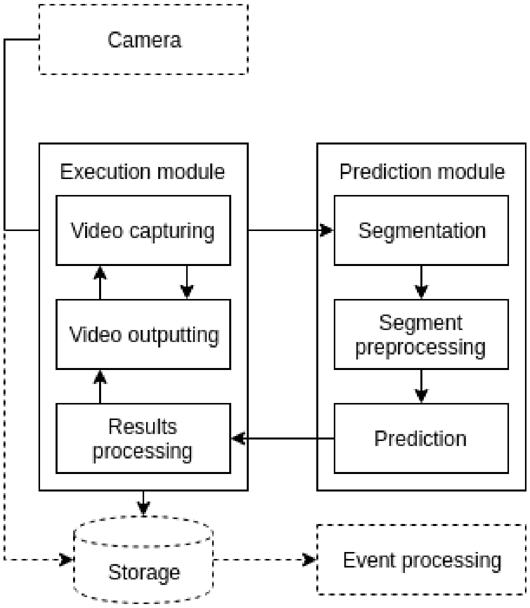
Aside from complex neural network architectures which require good GPUs to be used, the most popular predictive model at the core of a basic parking management system turns out to be SVM working on feature descriptors to determine labels. Among the feature descriptors, however, HOG is one of the most widely used, with another popular method being background subtraction.

In this work, we present a new architecture of a lightweight parking management system with the use of a primitive neural network to process raw pictures. We compare the quality of prediction with the most popular method combination of SVM and hog to evaluate the system, as well as present other statistical data.

II. Methods

The architecture of our system is inherently modular, consisting of an execution module and a prediction module, which should communicate with each other (Figure 1).

Figure 1: *High-level system architecture*



The execution module would capture a video stream from a camera and send it to a prediction module, while also outputting it on the screen with current labels visualized and continuing video capturing. After the prediction is made, it would update the labels and save them into a log, and then the cycle would start again.

The prediction module cycle has, on the other hand, much simpler work cycle. It accepts a frame from the execution module, retrieves parking slots from this frame, preprocesses the slots, predicts their status, and outputs prediction back to the execution module, waiting for the next frame to predict on.

The whole system is two separate processes running in parallel and exchanging information via queues.

In the core of the system, a simple Keras neural network is living. Its setup is quite simple, consisting of 3 convolutional layers with relu activation, each followed by a max pooling layer. On top of that, two more fully connected layers are put. The network was trained on 20 thousand pictures of empty and occupied parking spaces, for five epochs, and gave accuracy score 86% on validation.

Now let us talk about algorithms and concepts related to our ground models to compare with. HOG is a standard feature descriptor, used mainly for object detection. The idea behind HOG is that one can use edge directions of gradient distributions to describe the local shape of an object efficiently. The feature vector is robust to photometric and geometric transformation, while sensitive to rotation, and is suitable for human detection, especially face detection, mainly through binary classification.

Next, in the line, let us consider SVM. It is notable because it tries to provide the most optimal class boundaries, such that the minimal margin for an element is the highest possible. SVM is used virtually everywhere, in all kinds of tasks requiring classification, including such areas as text detection and handwriting recognition.

We have used SVM from the Scikit-learn framework, along with the Scikit-image implementation of HOG. The model was trained on 70000 images of empty and occupied slots, resized to 40x60, using three-fold cross-validation through GridSearchCV, with a linear kernel.

III. Results

We have tested the quality of SVM automatically through Scikit-learn functionality, and CNN from our system manually. The actual label distribution in test sets is shown in table 1. Here we can see that we can suppose the numbers of empty and occupied spaces to be approximately equal.

Table 1: *Testing sets label distribution*

Test set version	Occupied	Empty	Total
CNN	201	303	504
SVM	338498	302173	640671

The prediction quality metrics are shown in the table 2. Here we can see how SVM has excellent parameters, all over 90% which shows that state-of-the-art methods are indeed already good enough to use in production. With our CNN model, however, results are not so brilliant, with recall being a little below 0.7 meaning that one-third of occupied spaces are predicted to be empty. However, precision is quite decent, despite it being still a bit less than SVM has. The overall accuracy is about 0.85 which is still a decent result, even though it fades compared to SVM accuracy. VACC - accuracy on validation, TACC - accuracy on testing.

Table 2: *Prediction quality assessment*

Model	TPR	PPV	VACC	TACC
SVM	0.933	0.983	0.98	0.956
CNN	0.697	0.909	0.86	0.851

Now let us talk about time and memory consumption. For our work, we had two computers, one was used to train and evaluate SVM, and the other was used for training CNNs and testing the whole proposed system. The first one has AMD Ryzen 7 1700, 24GB RAM DDR4 2400 kt/s, 1 TB cached HDD + SSD storage and 10GB SSD 450Mb/s r/w + 60GB HDD 7200 rpm swap. The second one is Intel Core i7, four cores, 2400MGz, 8GB RAM, NVIDIA GeForce GT 635M 2048MB, and HDD 7200 rpm.

The SVM took about an hour to train, taking up about 16GB RAM in the process, including datasets. Training the CNN, however only took approximately 5 minutes, with negligent memory expenses.

We benchmarked the proposed system using the POSIX time tool, on the video 2m1s long, where it took 1m40s to process, which proves that our system can be used in real time. The speed of prediction was about 12 frames per seconds, while all the extra frames were skipped. As for the memory expenses, memory profiling has shown that the memory expenses do not go anywhere above 500MiB.

IV. Discussion

As can be seen from results, the state-of-the-art SVM version indeed has a superior quality which we are yet to beat. There are significant concerns both regarding its speed and memory consumption, mainly because our system is likely to be used on cheap computers with small speeds and below-average hardware, and thus should support such a possibility.

CNN quality, while is lower than SVM, is also well-acceptable, mainly because other parts of infrastructure could be possibly able to neutralize prediction errors. Our system with CNN at the core is fast enough to produce just-in-time predictions, light-weight, and does not consume much memory.

The usability of the system is another issue, as it lacks comprehensive GUI and is overall console-dependent. Furthermore, it is not so useful without any additional modules feeding on the prediction outputs.

Plenty optimizations could also have been applied, and we indeed tried some of them. There, for example, was a trick which forced the model only to predict those frames that have changed since the last predicted frame. It slightly increased the accuracy, yet the speed drastically degraded, so that real-time operation no more was possible.

There was also an issue with logging. In previous versions of software, we used to output all the predictions into files for future usage, but this slowed the system down significantly enough to disrupt proper visualization. Appropriate logging with existing logging tools ought to become present in one of the next versions.

V. Future work

We have successfully shown that the proposed system, despite being inferior in terms of accuracy, is viable and can work in a real-time environment. The future work in the field may include creating a GUI for the system, possibly with user experience investigation, adaptation to multiple listeners waiting for the system output, adding proper logging and additional functionality.

Another direction of work could go into optimization, which might require rewriting the system into C or similar low-level language. This change can increase the speed of the application, as well as reduce RAM consumption due to the lack of runtime to run.

References

[1] Zhao, Xiaotong and Li, Wei and Zhang, Yifan and Gulliver, T. Aaron and Chang, Shuo and Feng, Zhiyong (2017). A faster RCNN-based pedestrian detection system *IEEE Vehicular Technology Conference*

[2] Ardianto, Sandy and Chen, Chih-Jung and Hang, Hsueh-Ming (2017). Real-time traffic sign recognition using color segmentation and SVM *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*

[3] Neumann, Daniel and Langner, Tobias and Ulbrich, Fritz and Spitta, Dorothee and Goehring, Daniel (2017). Online Vehicle Detection using Haar-like , LBP and HOG Feature based Image Classifiers with Stereo Vision Preselection *IEEE Intelligent Vehicles Symposium, Proceedings*