PROJECT DESIGN PHASE REPORT

ON

PARKING MANAGEMENT USING COMPUTER VISION

A project report submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted by

SUJITH S (14402113)

DEPT. OF ELECTRONICS&COMMUNICATION ENGINEERING
SREE CHITRA THIRUNAL COLLEGE OF ENGINEERING,
THIRUVANANTHAPURAM-18

SEPTEMBER 2017

SREE CHITRA THIRUNAL COLLEGE OF ENGINEERING

THIRUVANANTHAPURAM-18

CERTIFICATE

his is to certify that this project entitled "**Parking Management using Computer Vision**" is a bonafide record of the seminar done by SUJITH S in partial fulfillment of the requirements for the award of B.Tech degree in Electronics and Communication Engineering of the University of Kerala, during the academic year 2017-18

PROJECT COORDINATORS:

Sri. KUMAR G.S.
Assistant Professor
Department of ECE
SCTCE

Smt. Preetha V. H.
Assistant Professor
Department of ECE
SCTCE

PROJECT GUIDE

Prof.(Dr.) Sheeja M.K.
Professor and HOD
Department of ECE
SCTCE

HEAD OF DEPARTMENT

Prof. Dr. Sheeja M K
Professor
Department of ECE
SCT College of Engineering

.

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to our project guide Prof.(Dr.)Sheeja M.K. (Head of Department of Electronics and communication engineering, SCTCE) for her exemplary guidance, mentoring and constant encouragement throughout the course of this Project. The blessing, help and guidance given by her from time to time shall carry me in a long way in the journey of life in which i am about to embark.

I also take this opportunity to express a deep sense of gratitude to Prof. Dr. S H Anilkumar (Principal, SCTCE), for his cordial support, valuable information and guidance, which helped us in completing this project through various stages.

I am really thankful to Sri. Kumar G S (Assistant Professor, ECE, SCTCE), our project coordinator for his invaluable guidance and assistance, without which the accomplishment of the task would have been impossible.

I would not forget to remember Mrs. Asha S (Assistant professor, ECE Department, SCTCE), Mrs. Preetha V H (Assistant professor, ECE Department, SCTCE) and Mr. George M Joseph (Associate professor, Department of ECE, SCTCE) for their encouragement and more over for their timely support and guidance till completion of this project.

I am also obliged to staff members of SCTCE for the valuable information provided by them in their respective fields. We are grateful for their co-operation during the period of our assignment.

Lastly, I thank my parents, brothers, sisters and friends for their constant encouragement without which this assignment would have been impossible.

SUJITH S

.

Abstract

In recent years, with the growing fleet of vehicles in cities, the task of finding parking has become increasingly difficult. An output found by drivers looking for a place to leave their cars are the public car parks and private, which are very common mainly in large centers. These parking lots hundreds of vacancies and the task of administering them can be difficult, if the management of vacancies is carried out manually. One of the ways to manage a parking lot is to collect information on the status of each of their vacancies. This solution can provide information on the number of parked vehicles, but also their The more widely used the extraction, the horizons in which detergents are more occupied, etc. Individual management of vacant parking spaces can go beyond the parking information, given that the information on the occupancy rate of public places present on the sides of the tracks can also be of great importance for the engineering of traffic. However, in these cases it is not possible to collect data using the perception because of the large number of people needed to fulfill this task. In addition to parking management, a methodology for detecting vehicles can serve as a basis for other systems such as Parking, detection of vehicles parked in forbidden systems that guide the driver directly to the next vacant parking space, sparing so time of the drivers and helping in the traffic de congestion.

Contents

List of Figures

List of Tables

1 Introduction

Information on the allocation of a particular parking space, as well as the location of occupied and empty spaces, can save a lot of time of the drivers, in addition to enabling better management of these establishments. Although simple systems, such as those based on gates, exhibit good results maintenance of the number of vehicles in a parking lot input and output control, the status of each of the individual defined in this way. In addition, this type of system is limited to enclosed parking lots, making it impossible to use them in public places and free access. An alternative to provide a global view of the current state of a parking lot with details on each of their vacancies' and the installation of sensors in their individual vacancies , because this solution presents a high cost, be difficult to maintain, especially in large. A simpler approach than the installation of individual sensors can be the multiple vacancies simultaneously by means of images, which can indicate if each of the places present in the image being processed is free or not. These systems, which can use as source images surveillance cameras installed in the parking lot, are potentially cheaper and compared to systems based on individual sensors, because they present some complexities inherent in their constructions. Among these complexities are the facts that images collected over a in open environments may undergo large variations in brightness, contain shadows that projected at different points depending on the period of the day, and contain noises introduced by changes in weather conditions such as rain or fog, for example. These complexities can be accentuated when images captured in different parking lots are compared. Examples are some of these problems, showing the variations that can the same location under different climatic conditions and different times of the day. The images also show the difficulty of obtaining a representation (Model o) of occupied space, since it is possible to extract the number of color combinations and models of vehicles parked in a space, other than that these vehicles may be many ways. An image-based job classification system that nullifies or effects of such complexities could make it cheaper, and therefore more feasible, to implementation of management systems

2 Objectives of this project

The main objective of this work is to propose a method of classifying vacancies of open parking spaces from fixed-camera images, considering the combination of trained classifiers with textural features. It is understood by classifying a vague, the task of defining its condition as free or occupied. The textural features were chosen for this work because it is believed that this type of characteristic can absorb the effect of the variations of luminosity, which occur due to climatic or day-to-day variations, without major impacts on the rates of occurrence of the classification system. In addition to the main objective, this work aims to fulfill the following secondary objectives Are:

- Create a dataset and a solution for parking troubles using the method of computer vision.
- Evaluate methods of combining classifiers with the aim of maximizing the system. For this, classifiers with character sets will be trained and the final response of the state

of a position will be given by means of a matching scheme of the responses of these classifiers.

- Check how generalized the models of the parking lots will be, placing them by using angled or standing images that are not part of their training set. This way, it will be possible to evaluate if a classifier created from the parking samples can be used to classify a parking lots, which did not contribute any training samples.
- Create a database of parking lots that enables the execution of the tests necessary in this work.
- Compare the proposed method with trained classifiers with color characteristics, which are used in several other works

3 Contributions of this project

The contributions of this work can be separated basically from those of a scientific and social nature.

3.1 Scientific contribution

The main scientific contributions expected for this are:

- The evaluation of the performance of ConvNet for the distinction between vehicles and terrain.
- The evaluation of the generalization capability of ConvNets through tests involving images collected from parking lots or angles that did not participate in their training.
- The creation of a parking image base and an image capture protocol that guarantees the extensibility and usability of the base for several tasks related to Intelligent Transport Systems

3.2 Contributions to society

- The creation of a method for classifying parking spaces that can serve as a basis for other systems, such as systems capable of indicating the most immediate place to the driver, thus saving travel time and reducing congestion, as well as helping to reduce the pollution caused by vehicles, which will need to circulate for a shorter time until they find a place.
- Enable the management of parking facilities in a more efficient way, providing information on each of their vacancies
- Provide a method of classifying vacancies that can be used by traffic engineering to detect locations with overcrowded parked vehicles.

4 Background and basic knowledge

Requirements:

1. Knowledge in Convolutional neural networks
2. Good knowledge in Linear algebra and some other mathematical concepts
3. Good knowledge in Python
4. Good knowledge in Tensorflow
5. Images to be trained
6. A system to train the model
7. Basic knowledge in web development

4.1 Convolutional neural networks

Convolutional Neural Networks (CNN[1]) can be recognized as an extension to regular Artificial Neural Networks (ANN). The main difference between these two methods is the usage of convolutional layers and pooling layers in CNN. In the convolutional layers, the value of each hidden unit is not just a linear transformation of all the hidden units of previous ayer, which is the case for fully connected layers in ANN. Instead, the value is a result of convolving a three dimensional filter with values of previous layer. The Pooling layer is a maximum spatial response filter that passes the maximum values of region in the input layer to the output.

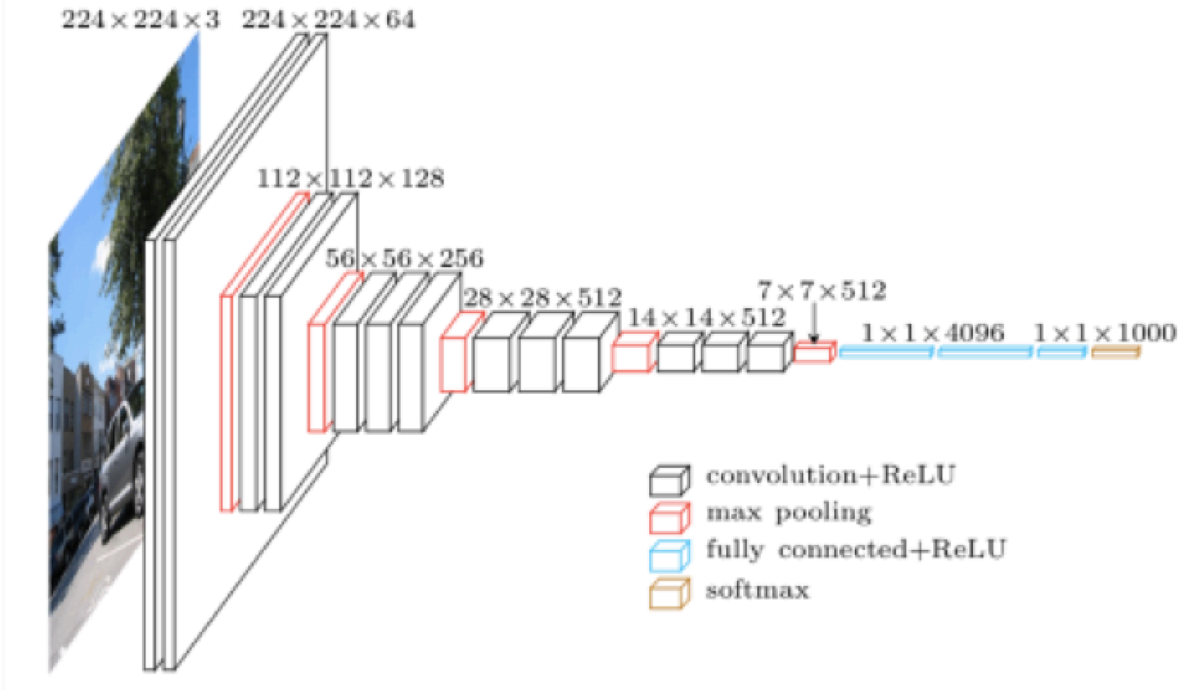


Figure 1: CNN

These two innovations let CNNs have more trainable layers compare to ANN, hence the name deep. Learned filters in convolutional layers are convolved with the entire feature map. Therefore, size of these filters are not commensurate to the spatial size of feature maps, as

[1] Complete guide to CNN is available at: `https://github.com/SUJITHhubpost/Convolutional-neural-networks`

it was the case for ANN. Accordingly, it dramatically shrinks search space for each layer. Pooling layers effectively reduce the spatial size of its input by the assumption that spatially close features are co related, which is mostly true for images, and therefore one of them can represent them all. It also makes the network less sensitive to translation of the input image. Based on these fundamental elements, many different networks have been designed and trained on large datasets for image recognition tasks. A few of the most successful ones are GoogleNet, VGGNet and AlexNet

4.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

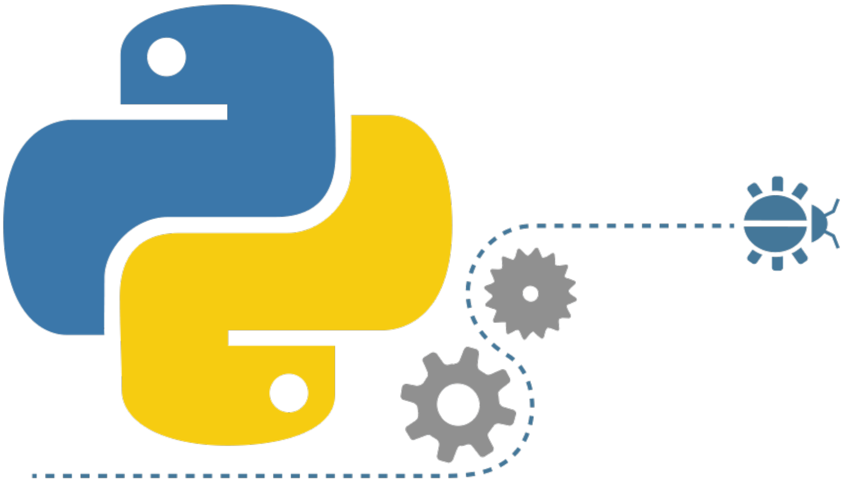


Figure 2: Python

4.3 TensorFlow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research

Figure 3: TensorFlow

organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

4.4 System Requirements

For a system, the considerations are:

1. Intel - Core i7-5820K 3.3GHz 6-Core Processor
2. Cooler Master - Hyper 212 EVO 82.9 CFM Sleeve Bearing CPU Cooler
3. Asus - X99-A ATX LGA2011-3 Motherboard
4. G.Skill - Ripjaws Series 32GB (4 x 8GB) DDR4-2400 Memory
5. Samsung - 850 Pro Series 512GB 2.5" Solid State Drive

6 eagate - Barracuda 3TB 3.5" 7200RPM Internal Hard Drive
7 EVGA - GeForce GTX Titan X 12GB Superclocked Video Card
8 EVGA - SuperNOVA G2 1300W 80+ Gold Certified Fully-Modular ATX Power Supply
9 Asus - DRW-24B1ST/BLK/B/AS DVD/CD Writer
10 Acer - XB270HU bprz 27.0" 2560x1440 144Hz Monitor
or Use cloud services

4.5 Web Development

Following are the basic needs in doing web development

4.5.1 Hyper Text Markup Language 5

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as ”heading”, ”paragraph”, ”table”, and so on
- Browsers do not display the HTML tags, but use them to render the content of the page
  An example code is given below

```
<!DOCTYPE html>
<html>
  <body>
    <!-- Paragraphs are fine -->
    <div id="box">
      <p>Hello World</p>
      <p id="test">Hello World</p>
            <pig weight=120kg></pig>
    </div>
    <div>Test</div>
  </body>
</html>
```

4.5.2 Cascaded Style Sheets

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. it saves a lot of work. It can control the layout of multiple web pages all at once External stylesheets are stored in CSS files
An example code is given below

```
<!DOCTYPE html>
<html>
<head>
        <style>
                h1 {
                background-color: green;
                        }

        div {
        background-color: lightblue;
                }

        p {
        background-color: yellow;
                }
        </style>
```

```
</head>
<body>

        <h1>CSS background-color example!</h1>
        <div>
        This is a text inside a div element.
        <p>This paragraph has its own background color.</p>
        We are still in the div element.
        </div>

</body>
</html>
```

4.5.3 Javascript

JavaScript is a programming language that allows you to implement complex things on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, or interactive maps, or animated 2D/3D graphics, or scrolling video jukeboxes, etc. —JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS).
Example code is given below

```
<html>
<body>
        <script type="text/javascript">
        document.write("JavaScript");
        </script>
</body>
</html>
```

4.6 Raspberry Pi

he Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

I am using raspberry pi 3 model B

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.

4.6.1 Web Hosting

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. A web host, or web hosting service provider, is a business that provides

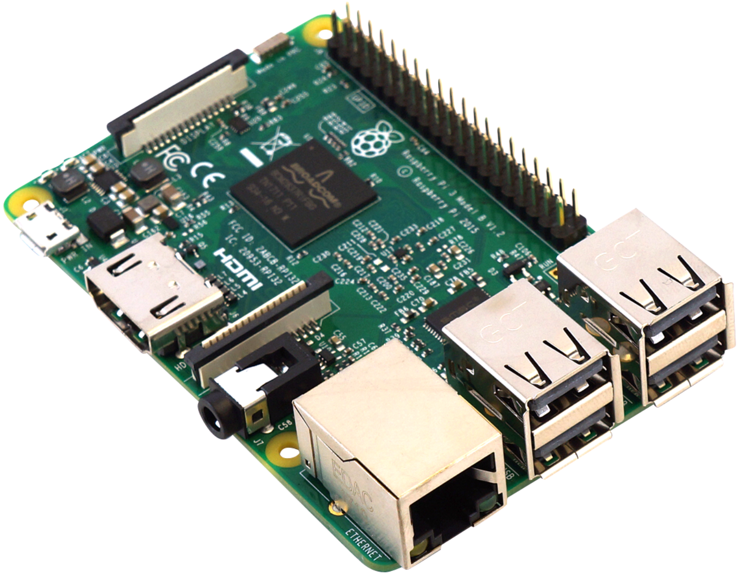


Figure 4: Raspberry pi

the technologies and services needed for the website or webpage to be viewed in the Internet. Websites are hosted, or stored, on special computers called servers.

4.6.1.1 000Webhost

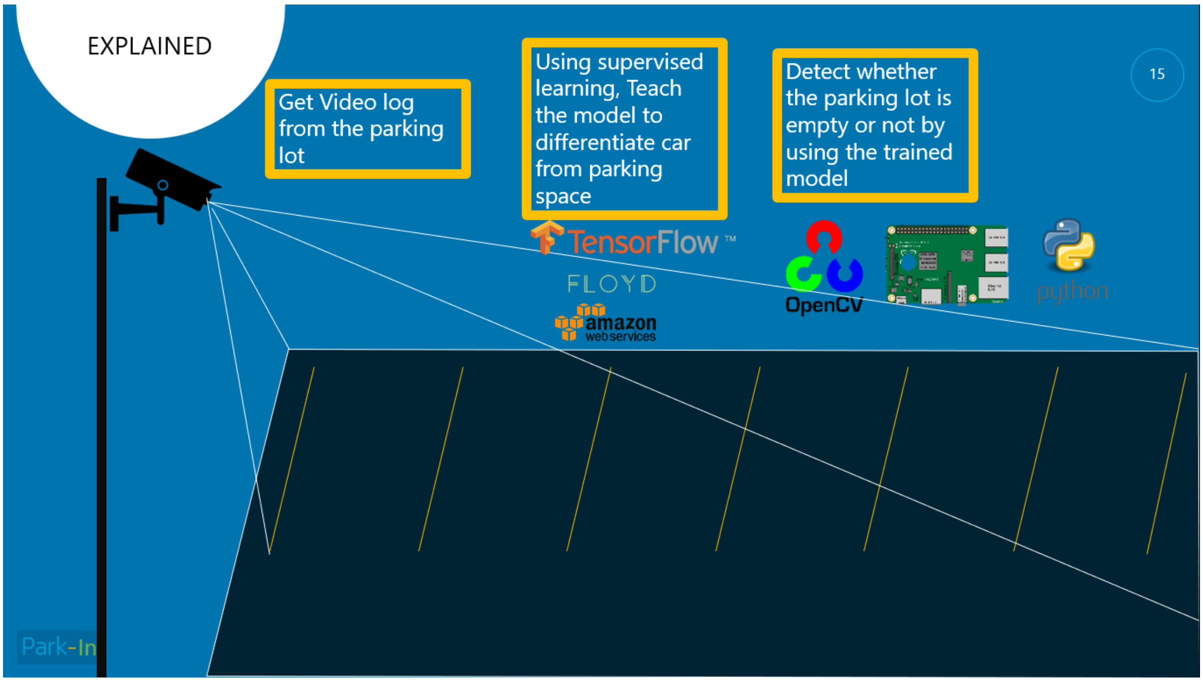
Located in the US, 000Webhost has been providing free hosting services since 2007. Its free hosting service includes 1.5GB of space, 100GB of bandwidth per month and five accompanying email accounts. PHP and MySQL are supported with no restrictions and users are guaranteed 99

The service is supported by donations from its community of users (nearly 14 million at the time of writing) and revenue generated from 000Webhost's premium hosting packages (24hosting). Starting at $2 per month, they aren't going to break the bank and give customers the option of moving to a more comprehensive plan should the need arise.

5 Implementation

5.1 Initial design

The proposed system is composed of three parts. First is the visual nodes, namely cameras, that are connected to the server through either local wireless network or through the Internet. The server is the second component. It includes a database, detection module, web service and event handler. The server collects images from visual nodes, feeds them to the detection module and collects and stores its output in the database and provides web service for front-end applications to get information from the database. The third element is the front-end that presents parking lots vacancies to users. In the following each element is discussed in detail.



EXPLAINED

Get Video log from the parking lot

Using supervised learning, Teach the model to differentiate car from parking space

Detect whether the parking lot is empty or not by using the trained model

Figure 5: A screenshot from the presentation
The figure shows the implementation stages.

- c *Visual Nodes*

  Regular color camera images are used in this project while there exist other choices such as microwave radar. As discussed, camera is a suitable choice for parking management system, due to its low maintenance, low cost per stall and ease of scalability. However, a drawback is that camera raw output images are highly sensitive to environmental parameters. Ranging from different light (time of the day) to different weather. Adding to these, we should mention camera parameters and point of view. Camera parameters include, camera intrinsic parameters, image size, imaging frequency and low level filtering such as noise cancellation. These parameters vary drastically between each camera model and manufacturer. Camera point of view is also effective in terms of what view of the stall is being observed. A logitech 720p camera is used in this project.

- b *Server*

  The server in our system has four responsibilities: First, it is to host the database. A relational database is used in this system. It stores separate tables for different parking lot where each element in the table corresponds to a stall. Each stall has four mandatory fields. 1) Stall ID. A unique number in the parking lot. 2) Stalls bounding box coordinates (in image space). Coordinates are entered through our GUI by the administrator. 3) Image blob. Cropped image of the stall from current visual feed. This will be updated by the server on fixed intervals 4) Status. A binary value indicating vacant or occupied. This is also being updated by the server from the detection method's response.

  Second, it is collecting data from cameras. Cameras can be connected to the server in a local network or can be connected through Internet. If cameras do not use HTTP protocol, a local communication protocol will be used on the server. If they do, requests and responses can be done with HTTP. If the cameras are connected to Internet, there is no need for the server to be located close to visual node and a server on the cloud is a valid option

Third, it is serving a web-service. It bridges the database to our system's front-end. The main functionality supported is retrieving status of all stalls in each parking lot. Finally, it feeds the images from visual nodes to the detection module along with bounding boxes of stalls and receives the detection module predictions.

c *Detection Module*

The detection module is responsible for reporting the occupancy status of a parking stall given the image of the stall. We use a Convolutional Neural Network for this task. Having few constraints on the input data puts a heavy burden on the detection system. Therefore robustness and generality of the detection algorithm have the highest priority. To achieve this, choosing the right network and training procedure is crucial. The design of our network is based on VGGNet-F. It has five convolutional layers where each is followed by a pooling layer and Rectified Linear activation function. It has three fully connected layers at the end that use the features from the convolutional layers for classification. The VGGNet architecture features a simple and uniform design throughout the network.

5.2 First phase Implementation

Successfully designed and implemented a car detection algorithm using MobileNet SSD, Caffe and ImageNet dataset Using the programming language python and the free and open source computer vision C++ library OpenCV. The model worked perfectly during the test stage. The implemented model contains

5.2.1 Camera

The camera I used here is Logitech C270 HD Webcam. The Logitech C270 HD Webcam is a high utility device that helps you to enjoy seamless video calling. This device comes with easy installation process that offers a hassle-free set up. The ergonomic design and sleek body helps in saving space and makes it easy to install the webcam on your PC or laptop. The adjustable design makes it easy to tilt and use it according to your needs. It features 'Logitech Fluid Crystal Technology' and has a 3 MP camera which enhances picture quality while the integrated microphone delivers perfect sound quality. It supports video calling and enables you to record videos in HD quality.

Figure 6: Camera

This webcam comes with Logitech Fluid Crystal Technology that offers high quality video calling. This feature records visuals with complete detailing and allows you to enjoy vivid colours and more depth. It also records videos with sharpness and captures every movement without any distortion. This technology also provides superior audio quality.

This camera was perfect for the first phase implementation of this project.

5.2.2 Dataset : COCO

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

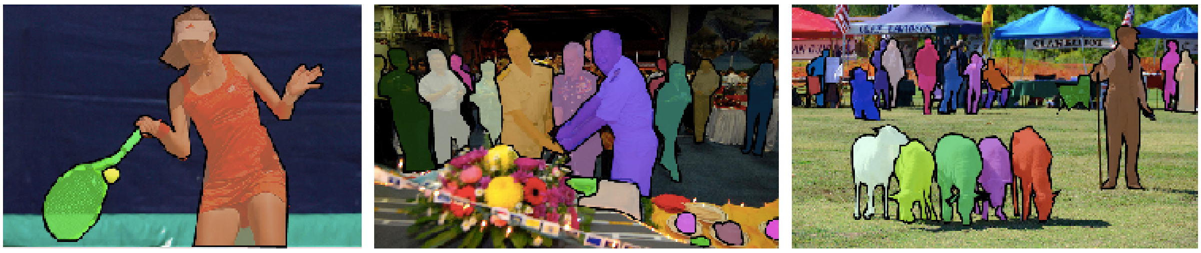
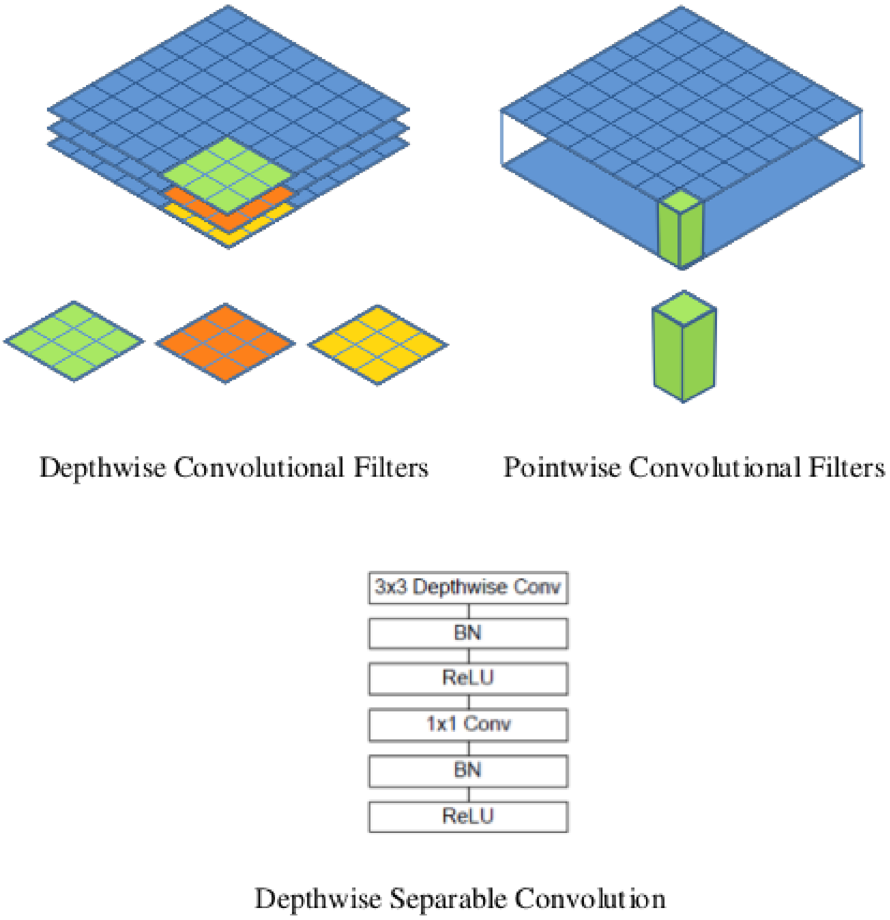


Figure 7: COCO

Dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

5.2.3 MobileNet SSD as Meta Architecture

Deep learning has fueled tremendous progress in the field of computer vision in recent years, with neural networks repeatedly pushing the frontier of visual recognition technology. While many of those technologies such as object, landmark, logo and text recognition are provided for internet-connected devices through the Cloud Vision API, we believe that the ever-increasing computational power of mobile devices can enable the delivery of these technologies into the hands of our users, anytime, anywhere, regardless of internet connection. However, visual recognition for on device and embedded applications poses many challenges — models must run quickly with high accuracy in a resource-constrained environment making use of limited computation, power and space.



Depthwise Convolutional Filters

Pointwise Convolutional Filters

Depthwise Separable Convolution

Figure 8: Mobilenet

MobileNets, a family of mobile-first computer vision models for TensorFlow, designed to effectively maximize accuracy while being mindful of the restricted resources for an on-device or embedded application. MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. They can be built upon for classification, detection, embeddings and segmentation similar to how other popular large scale models, such as Inception, are used.

SSD is an unified framework for object detection with a single network. You can use the code to train/evaluate a network for object detection task.

I trained this model from a MobileNet classifier(caffemodel and prototxt) converted from tensorflow

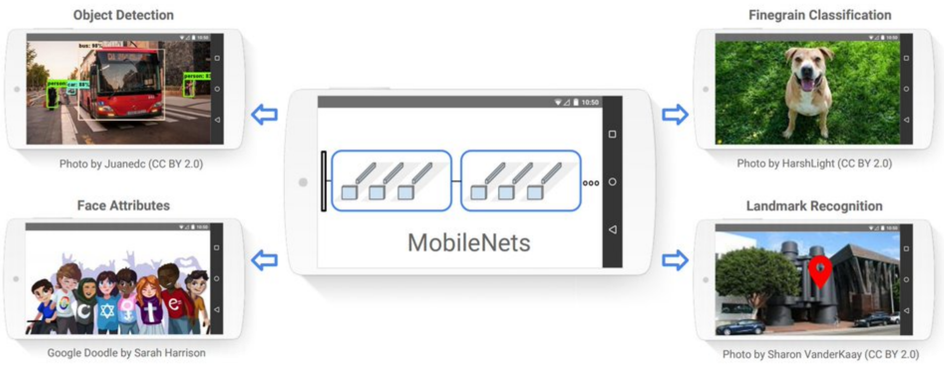

Object Detection
Photo by Juanedc (CC BY 2.0)
Face Attributes
Google Doodle by Sarah Harrison
MobileNets
Finegrain Classification
Photo by HarshLight (CC BY 2.0)
Landmark Recognition
Photo by Sharon VanderKaay (CC BY 2.0)

Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

Figure 9: Architecture

5.2.4 OpenCv

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

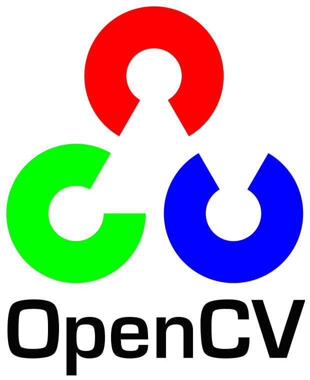


Figure 10: OpenCv

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

5.2.5 Caffe

Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by Berkeley AI Research (BAIR) and by community contributors. Yangqing Jia created the project during his PhD at UC Berkeley. Caffe is released under the BSD 2-Clause license.

5.3 Results

The model successfully detected cars and other vehicles when tested. Further, I would like to add some more data.

Figure 11: The model detecting Car

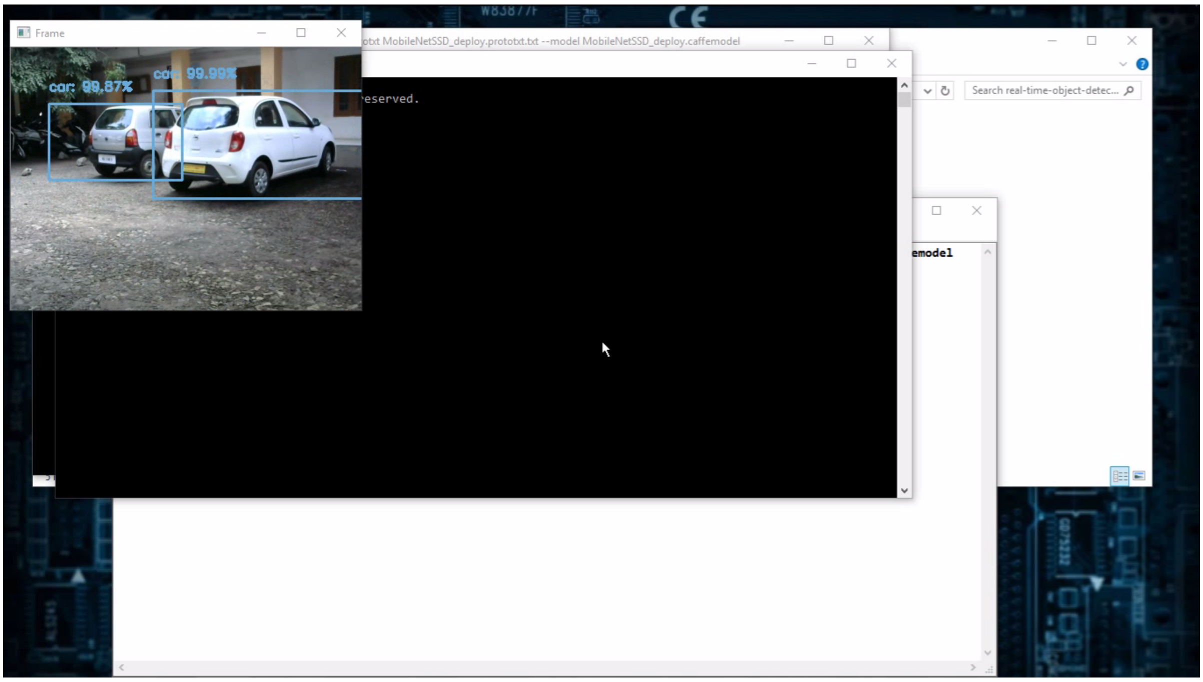
Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding. Switch between CPU and GPU by setting a single flag to train on a GPU machine then deploy to commodity clusters or mobile devices.

Extensible code fosters active development. In Caffe's first year, it has been forked by over 1,000 developers and had many significant changes contributed back. Thanks to these contributors the framework tracks the state-of-the-art in both code and models.

Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process over 60M images per day with a single NVIDIA K40 GPU*. That's 1 ms/image for inference and 4 ms/image for learning and more recent library versions and hardware are faster still. We believe that Caffe is among the fastest convnet implementations available.

5.4 Results

The model successfully detected cars and other vehicles when tested. Further, I would like to add some more data. With further research and adding a website to this project, The project is expected to be completed at the middle of 8th semester. The total cost of this project is 2 Lakh Rs.


Frame
car: 99.87%
car: 99.99%
otxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.caffemodel
Search real-time-object-detec...

```
eserved.
```

```
emodel
```

Figure 12: The model detecting Car

5.5 Next step

Next step is to create a dataset that can clearly distinguish between free space and occupied space in a parking lot using more training and spending some more resources.

.

Conclusion

this project[2] is to design and implemented a novel parking management system that uses deep convolutional neural networks for stalls' status detection. A complete system for visual Parking Guidance and Information system including, detection method, server and front-end website is to be implemented.

In the future work, I want to expand the real world experiments, so more conclusive performance results can be obtained. Another objective is to improve the dissemination of the information such that instead of broadcasting raw parking status, the system optimizes the data that each user receives based on their location and parking lot vacancies.

[2] presentation to the project is available at `https://github.com/SUJITHhubpost/Main-project`

Reference

[1] Speed/accuracy trade-offs for modern convolutional object detectors, Huang et al., CVPR 2017 (paper describing this framework)

[2] Towards Accurate Multi-person Pose Estimation in the Wild, Papandreou et al., CVPR 2017

[3] YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video, Real et al., CVPR 2017 (see also our blog post)

[4] Beyond Skip Connections: Top-Down Modulation for Object Detection, Shrivastava et al., arXiv preprint arXiv:1612.06851, 2016

[5] Spatially Adaptive Computation Time for Residual Networks, Figurnov et al., CVPR 2017

[6] AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions, Gu et al., arXiv preprint arXiv:1705.08421, 2017

[7] MobileNets: Efficient convolutional neural networks for mobile vision applications, Howard et al., arXiv preprint arXiv:1704.04861, 2017

[8] nception-v4, Inception-ResNet and the Impact of Residual Connections on Learning (Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke):arXiv:1602.07261v2 [cs.CV] 23 Aug 2016