GAIA Lab: Parking Spot Finder

02.24.2020

Travis Bonneau

Allen Wang

Walter Han

James Landry

Md Siamul Islam

Abstract

A mobile app combined with a geospatial map that will use cameras in the UTD parking lot to determine if a parking spot is occupied or not. We will use the security cameras already in place to look at the mapped parking spots and then use a model to determine if each spot is occupied, and will tell the user in the mobile app or the website. This will solve a big problem with parking where students waste a lot of time looking for free spots.

Table of Contents

Executive Summary

All drivers have had a time where they had to drive around looking for an open parking spot, wasting gas and lots of time that may be crucial to getting to an event on time. What if there was an easy way to find an open parking spot? That is the problem that our project is meant to solve.

Our solution is a parking spot finder app combined with parking status recognition technologies to be able to locate specific free parking spots and help guide users to those spots. The recognition solutions aims to tap into security camera feeds of parking lots in order to recognize individual free parking spots.

Existing solutions to parking spot finders generally focus on locating a free parking lot rather than a free parking spot. They are capable of maintaining count of the cars in a parking lot and can direct users to the lot, but they are otherwise incapable of locating and directing users to specific spots.

This is the case seen with mature preexisting solutions, such as the East Tennessee State University Parking App and Parkme.com. Additionally, existing solutions do not account for non-temporary parking permits that users may have, such as the colored passes at UTD, and are geared for undifferentiated or public spaces. For example, Parkme.com is geared towards finding public parking spaces while East Tennessee State University does not have differentiated parking spots, asides from handicap, staff, and visitor parking, and the parking app makes no distinctions between such spots.

Where our solution differs is that it must identify not only parking lots that are free, but specific parking spaces that are free and possibly help route the user to the spot. Moreover, it must do so without the aid of specialized sensors for detecting cars, such as the ones seen in the parking structures at UTD or near certain traffic lights; instead, it must extract parking statuses from the feed of general-purpose security cameras located in parking lots. Additionally, our solution must also find spots that are compatible with the parking passes that users have unlike other solutions that are geared towards generic or public parking spaces, and must also be able to filter and prioritize spots according to user preferences, such as people preferring to park near their classrooms.

Introduction

Our parking spot finder project is one that encompasses both the front end and back end of a parking spot detector. It will be a tool accessible through the web, whether it be on desktop or mobile, and in the future through a downloadable application as well.

The goal of our project is to develop a tool that can be used to help drivers find open parking spots through the use of geospatial imagery analytics. This will be done by processing video footage from the available parking lot cameras and using algorithms that we will develop to be able to detect whether or not a parking spot is occupied or not. We hope that with a tool like this that it will make the lives of students a bit easier and that it will be able to save them time that they would instead be wasting on driving around looking for parking spots if they did not have the application.

Currently, there already exists real-time tracking of parking spot availability in most parking garages. However, as for open parking spaces, there is a clear lack in this kind of technology. That is where our project comes into play as its focus is on providing the kind of parking assistance available in parking garages to open parking spaces. The difference being that for open parking spaces there are usually no dedicated cameras for detecting parking spaces. This means that security cameras, which most open parking lots will have, will have to be our source of information so that we can process that video and determine what spots are open.

Discussion

Frontend Solutions

Frontend solutions are required to present parking status and allow the user to specify their preferences.

Four major candidate platforms have been identified as potential frontend targets - desktop web, mobile web, iOS, and Android. For the purposes of the project, mobile web has been identified as the most efficient to target because doing so would make the app available to all mobile devices, the most likely device that will be used by people searching for parking spots while driving.

iOS/Android platforms offer superior integration and performance, especially with regards to bandwidth consumption as the app would only be downloaded once on installation instead of each time the user uses the app. This would be beneficial to mobile users who may have metered cellular data plans. However, targeting only one of these platforms would exclude large segments of mobile users, and targeting both could potentially render the project schedule infeasible due to two sets of solutions needed. As the project is intended as a proof of concept, the benefits of superior performance do not justify the increased development time. However, should time permit after completion of all other project milestones, the targeting of iOS/Android may be considered.

Required technologies to target the mobile web platform are HTML, CSS, Javascript - technologies standard to nearly all forms of web development. As the state of parking spots is often in constant and rapid flux, the React framework, which automates the process of updating content in dynamic web pages, will greatly simplify the frequent updates that must occur to the view of parking states.

The Bootstrap framework will be used as it provides premade UI component and layout solutions which will help expedite the process of creating the mobile web frontend.

The ArcGIS Javascript API provided by Esri will be used to render the map of the campus along with parking statuses overlaid as the library provides excellent integration with the existing extensive ArcGIS databases that are presently used to store auxiliary information about parking spots (parking pass type, location, etc.). Furthermore, the ArcGIS API provides integrations for efficiently performing queries on many parking spots, such as driving directions, nearest spots, and spots that are compatible with a pass type.

These solutions are regarded as mature technologies; however, unlike other solutions, these are combined and applied to create a unified system that will be integrated into existing campus systems, namely the existing campus map, ArcGIS databases, and security camera feeds.

Parking Spot Recognition Solution(s)

The parking spot recognition solution(s) are required to identify vacancies or occupancy in the parking spaces using Geospatial Imagery Analytics methods. It utilizes Computer Vision, and training Machine Learning models, to extract open parking spots in a parking space based on data extraction from video feeds from the security cameras.

The recognition solution(s) will be used for object detection to identify all of the parked vehicles and to verify overlapping of the vehicles with the respective parking spot. Convolutional Neural Networks (CNNs), that fall under the domain of Deep Learning Scheme for Computer Vision, were used to analyze each parking spot and to predict occupancy or vacancy of those spots. The vehicles and objects of interest are located and classified within certain boundaries. Deep neural networks differentiate the vehicles from objects that are not vehicles by analyzing a multitude of features. Appropriate artificial intelligence systems or machine learning models were developed as best fit for the solution, and were trained using the video feed data. Image segmentation and other techniques and tasks were undertaken for refinement (as required), to achieve a satisfactory accuracy of spot vacancy/occupancy detection with the model. The phases include detection and IoU(Intersection Over Union). Detection decisions are returned as Boolean. Research and some experimentation was required with the existing popular AI models to compare results and to identify the best-fitting model for the solution; Fine tuning and modifications were undertaken.

For the Parking Spot Recognition solution(s), deep learning architecture in CV: VGG-16 was used for one of two solutions, while the second solution was based on a custom 3X64 Node Convolutional Neural Network model. The VGG16 model trains on RGB color image dataset and the 3X64CNN model on grayscale data upon preprocessing; One model can yield better accuracy/loss than the other, depending on the type/quality/amount of data we are able to train with (considering the limited resources during the crisis timeframe). Both models implements the following:

Once a dataset is populated for training the model, it first undergoes Pre-processing where the images are normalized, converted, etc. to prepare it to be fed into training the model. Data augmentation could be done to improve the datasets if there is a lack of data in this timeframe and deal with overfitting to some extent. The validation sets and test sets were prepared as well and directories set-up accordingly.

The models were then built. The VGG-16 model was fine-tuned using eg. Keras (a NN-library and API for building and training deep learning models) and the CNN model was tweaked to fit the project solution. Next, the models were trained using the pre-processed datasets. The epoch value and other parameters can be tweaked for optimizations.

Predictions were run and statistical tools such as confusion matrix were plotted to assess the outcome of factors such accuracy and losses. The models were tweaked and fine-tuned as necessary, and were re-trained. The 3X64-CNN model, during a session, yielded >97% accuracy on the test set which could be. The prediction outputs along with the spot locators or indexed locations were returned as JSON and saved. Endpoints were established so that these output objects can be transmitted and utilized by the front-end.

The model was saved as a .h5 file and was deployed on GeoEvents-ArcGIS Server so that calls can be received by the front-end of web-application at the prediction endpoint.

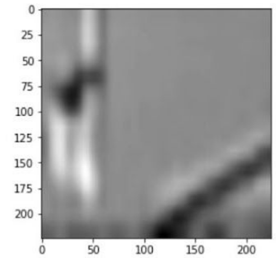
The SpotNN_vgg16 model code includes detailed comments, including some documentations and some important scripts/commands that might be very useful for future developers to work on or improve on this proof-of-concept solution, and makes the code very easy to follow. In addition, a demo front-end and Flask web services were used to pseudo-deploy and test the CV solution components prior to pushing it to github/project.

Some of the technologies, libraries, and/or platforms include OpenCV, Tensorflow, Matplotlib, numPy, keras, skLearn, Python, and Project Jupyter. Google CoLab was primarily used to avoid dependency and many other issues that arose initially. Tools and technologies found via research were used for the project as needed. Polygon selection tools, spot extractors, and other tools developed by the team supplements the technological needs for this project.

The scope and features of the recognition solution was dependent on the time constraint, access to, and availability of resources during the crisis. Parking lots are often empty and have a limited variety of parked vehicles and hence the dataset to train the model was limited. The model could benefit from image data extracted with different weather patterns, shadows, time of days, variety and size of vehicles, and so on, however, the model can simply be re-trained with a better dataset in the future in order to get better predictions on a larger variety of instances. The solution(s) are aimed to demonstrate a proof-of-concept as a deliverable; Additional features or capabilities can be added later or improved, with sponsors and mentors approval and given additional time and resources.

A screenshot of the run incl. accuracy/losses stats is provided below:

```
(30, 56)
```




```
381
0
1
0
1
1
1
1
1
1
1
Epoch 1/10
WARNING:tensorflow:Layer conv2d_39 is casting an input tensor from dtype float64 to the layer's dtype of f
loat32, which is new behavior in TensorFlow 2.  The layer has dtype float32 because it's dtype defaults to
floatx.

If you intended to run this layer in float32, you can safely ignore this warning. If in doubt, this warnin
g is likely only an issue if you are porting a TensorFlow 1.X model to TensorFlow 2.

To change all layers to have dtype float64 by default, call `tf.keras.backend.set_floatx('float64')`. To c
hange just this layer, pass dtype='float64' to the layer constructor. If you are the author of this layer,
you can disable autocasting by passing autocast=False to the base Layer constructor.

11/11 [==============================] - 13s 1s/step - loss: 4.9211 - accuracy: 0.6901 - val_loss: 0.5901
- val_accuracy: 0.8205
Epoch 2/10
11/11 [==============================] - 13s 1s/step - loss: 0.5146 - accuracy: 0.7456 - val_loss: 0.3189
- val_accuracy: 0.7692
Epoch 3/10
11/11 [==============================] - 13s 1s/step - loss: 0.3393 - accuracy: 0.8304 - val_loss: 0.2174
- val_accuracy: 0.8718
Epoch 4/10
11/11 [==============================] - 13s 1s/step - loss: 0.3014 - accuracy: 0.8450 - val_loss: 0.2054
- val_accuracy: 0.8974
Epoch 5/10
11/11 [==============================] - 13s 1s/step - loss: 0.2454 - accuracy: 0.9123 - val_loss: 0.1825
- val_accuracy: 0.9231
Epoch 6/10
11/11 [==============================] - 13s 1s/step - loss: 0.2247 - accuracy: 0.9591 - val_loss: 0.1587
- val_accuracy: 1.0000
Epoch 7/10
11/11 [==============================] - 13s 1s/step - loss: 0.2216 - accuracy: 0.9503 - val_loss: 0.1398
- val_accuracy: 0.9744
Epoch 8/10
11/11 [==============================] - 13s 1s/step - loss: 0.2031 - accuracy: 0.9678 - val_loss: 0.1476
- val_accuracy: 0.9744
Epoch 9/10
11/11 [==============================] - 13s 1s/step - loss: 0.2081 - accuracy: 0.9708 - val_loss: 0.1489
- val_accuracy: 1.0000
Epoch 10/10
11/11 [==============================] - 13s 1s/step - loss: 0.1995 - accuracy: 0.9678 - val_loss: 0.1561
- val_accuracy: 0.9744
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/resource_variable_op
s.py:1817: calling BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops) with c
onstraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
INFO:tensorflow:Assets written to: 64X3-CNN.model/assets
```

Resources

The two main resources used are the UTD security cameras in the parking lot, and the computers in the GAIA lab. Each person can also work on their own personal laptops if they want, but the GAIA lab is always open to work there and the project will eventually be hosted on their server at the end. Should the optional goal of developing and deploying native iOS or Android apps, licenses will be procured to enable publishing on each platform's respective app store. The mentor we will be in contact with is Brent Dell and the advisor that is leading the project is May Yuan. We will be using the CompleteView software to look at the parking lot footage, and will be using ArcGIS for the geomap of the parking lots.

Key Roles

Travis Bonneau

Working on the computer vision side of the project which will happen in the backend and will provide the front end with the information needed to show the consumer the open parking spots. Note taker and point-of-contact for all the meetings with our mentor, Brent Dell.

Allen Wang

Working on the computer vision side of the project which will happen in the backend and will provide the front end with the information needed to show the consumer the open parking spots.

Walter Han

Developing the mobile app (web and optionally iOS/Android) for presenting parking spot statuses and, optionally, routing and assigning people to parking spots. Will work with backend to understand and possibly aid in designing means of representing, storing, and disseminating parking information.

James Landry

Team Leader (subject to change). Researching and developing an Artificial Intelligence system to train a model to determine whether a parking space is open. This will be accomplished by using various tools that separates each spot for training and use of the model.

Md Siamul Islam

Researching and developing an Artificial Intelligence system to train a model to determine whether a parking space is open. This will be accomplished by using various tools that separates each spot for training and use of the model.

Communication Plan

The main forms of communication among teammates that will be utilized are GroupMe, Discord, and weekly in person meetings along with Microsoft Teams.

Communication through GroupMe is generally for quick interactions to answer any simple questions anyone has, inform the group quickly over something, or solidify any meeting times or places.

On Discord, we utilize the voice chat functionality for anything that would be hard to communicate through text and for planned group meetings. Group voice chats are held on an as-needed basis. We also utilize the ability to create different chat categories to organize our communication.

Our weekly in person meetings are held every Friday afternoon and include our entire team and sponsor/supervisor. During these meetings we discuss the progress we have made, any problems we have run into, and the next steps in the project. Along with some written supplements and demonstrations, this constitutes our weekly report. Additionally, Microsoft Teams is used to contact our mentor and sponsor outside of meetings.

Risk Analysis/Contingency Plan

There is a risk of losing team members to sickness or other unforeseen accidents which would reduce available manpower. For that reason, all work is to be accessible via GitHub to all other team members so that work already done is not lost with losing a team member. By the same token, team members are expected to submit changes frequently. Team members are expected to brief each other of their work, generally weekly, so that it is possible for a person's to be continued by the remaining members.

Additionally, there is also risk of being unable to procure the needed infrastructure and licensing to deploy the mobile app and web services for the parking spot finder. This would delay the completion of Phase III: Deployment. However, delay is largely considered acceptable as the project acts as a proof of concept, meaning that there is more value placed on the deliverables of Phase II: Prototyping than any other phase.

Difficulty in securing live feeds from camera needed to perform real time status updates is a major blocker of complete deployment and general availability. Procuring access to such feeds is likely to take longer than the time permitted by the semester, given the negotiations required with Salient Systems, the operator of the cameras. However, granted that the project is intended as a proof of concept, an inability to achieve general availability is considered acceptable. Therefore, achieving real time updates and general availability have been excluded from the scope of the project.

Costs

All of the cameras are already installed so no real costs are required at this time. However, this project is really a proof of concept so if successful more cameras in the parking lots may be required, but that would occur after this project is due, so we are not considering that as a potential cost. Costs may be incurred should the optional goal of iOS/Android apps be pursued as licenses will be needed to publish on their respective app stores. The estimated cost of pursuing such licenses is $99 a year and $25 one-time fee, respectively.

Timetable

Phase I

Research

Due 02/14/2020

Research possible technologies that can be used to address means of determining the status of parking spots, efficiently representing, storing, and disseminating parking statuses, and means of presenting such information efficiently and ergonomically. An optional issue to address is efficiently assigning and routing people to parking spots.

Our deliverables for this stage will be three lists of possible technologies and design solutions to address each of the three aforementioned major issues.

Phase II

Prototyping

Due 04/03/2020

This phase will consist of preparing usable prototypes of solutions for the three major issues identified. Integration between the solutions is also to be part of the deliverables. Acceptance testing will be performed at regular intervals during the phase to ensure completed portions meet customer specifications.

Deliverables will consist of a mobile app (web with the optional goal of native iOS/Android), web server with database for storing parking statuses, and working service for recognizing parking spot states, all integrated with each other.

Phase III

Deployment

Due 04/30/2020

This phase will comprise of deployment of mobile apps and web servers needed to make the system ready for general use in production. Infrastructure needed will be acquired and configured to prepare for production use. Additionally, final acceptance testing will happen during this phase to ensure all systems meet customer specifications.

However, due to anticipated difficulty in securing live feeds of data needed to determine parking statuses, generally availability will not be part of the deliverables.

Evaluation

The success of the mobile app will be gauged on the acceptance by mentor, sponsor, and (optionally) random selected individuals, especially with regards to user experience, ergonomics, and expediency of finding a parking spot.

Accuracy of model in determining parking spot status will be used as the measure of success with regards to the recognition solution. A mark of success is the accuracy reaching levels that are perceived as acceptable based on a subjective review by mentor and sponsor.

Accuracy, expediency, and quality of user experience will be considered gauges of the overall success of the entire project. These will be measured by mentor, sponsor, and (optionally) randomly selected individuals.

Contact Information

Travis Bonneau

tmb170230@utdallas.edu

832-260-6567

I am a Computer Science student at the University of Texas at Dallas, and my main interest is in computer vision and image processing. I am about to finish my Bachelors degree and am currently doing fast track for my Masters degree. I have had an internship in the past at State Farm as a software developer intern, and I have an up-coming internship at Collins Aerospace.

Allen Wang

axw161630@utdallas.edu

512-363-8083

I am a Computer Science student at the University of Texas at Dallas who has interests in Artificial Intelligence. Currently finishing my Bachelors and pursuing my Masters in Intelligent Systems.

Walter Han

wch170030@utdallas.edu

832-302-4301

I am a Computer Science senior, enrolled in Fast Track for Data Sciences, with a strong focus on web development, computer graphics, and artificial intelligence. I have had some industry experience with web development and partake in game production labs in ATEC.

James Landry

jtl160130@utdallas.edu

832-715-2440

I am a senior undergraduate Fast Track student for Information Assurance at The University of Texas at Dallas. I graduate this Spring (2020). I have taken a graduate level Artificial Intelligence course that contained a semester project. My teammate and I used a model to run Image Classification using Glimpse Actions, specifically on the FashionMNIST dataset. My pursuit of Information Assurance comes from my interest in cyber security. I started my love of computer science in high school when I was introduced to basic web development languages such as HTML, CSS, and PHP using Codecademy. I am a Freshman Mentor on campus, as well as President of the Pokemon League and Fundraising Coordinator of Anim3, the anime club.

Md Siamul Islam

mxi170830@utdallas.edu

469-226-5502

I am a senior-year, undergraduate-level student majoring in Computer Science at The University of Texas at Dallas; I will be graduating this Spring (2020). I have pursued UG research on Machine Learning, and Data Science. I have worked on related project(s) that identifies mineral/salt deposits using seismic imagery utilizing ML, alongside software development project at Tyler Tech, and have been interested in ML/DS/AI/CV field and related domains ever since, trying to get exposure to prepare myself for a career. Also a car enthusiast, and enjoy travelling and playing guitar.

Sources

Apple Inc. “Choosing a Membership.” *Choosing a Membership - Support - Apple Developer*, 2020, developer.apple.com/support/compare-memberships/.

Google. “How to Use the Play Console - Play Console Help.” *Google*, 2020, support.google.com/googleplay/android-developer/answer/6112435?hl=en.

Owens, Lee. “Easier Parking at ETSU? There's an App for That.” *WCYB*, WCYB, 24 May 2016, wcyb.com/news/tennessee-news/easier-parking-at-etsu-theres-an-app-for-that.

Parkme Inc. “We Make Parking Easy.” *ParkMe*, 2020, www.parkme.com/dallas-parking.

Appendix

Original Project Outline provided by Sponsor

1.1. Project Title	Where are the open spots? Develop a location-based service mobile app, SpotFinder, to guide parking in a large outdoor parking lot
1.2. Discipline / Body of Knowledge addressed	Geospatial imagery analytics emerges as a key player in smart city technologies. From security surveillance to driverless cars, geospatial imagery analytics incorporates machine learning algorithms to recognize people, landmarks, and other geographic features for information about human activities and environmental conditions. Location-based services (LBS) integrate mobile, web services, and cloud technologies in geospatial information systems (GIS) to provide useful information dynamically based on one's location. The combination of geospatial imagery analytics and LBS allows individual access to relevant information about what and where extracted from image feeds. The proposed project contributes to Geospatial Information Science (GIScience), a multidisciplinary field cross-cutting geography, data science, and computer science.
1.3. Project Abstract	The proposed project aims to develop LBS mobile app to guide drivers to open spots in an open parking lot based on geospatial imagery analytics. Most parking garages now provide real-time information on open parking spaces on different levels. However, parking assistance on open parking lots is mostly lacking. Existing smart parking solutions generally fall in to four categories: sensor-based indicators, reservation-based systems, dynamic pricing management, and participatory crowdsourcing. These solutions have been deployed in parking garages, metered parking, and roadside or apartment parking. Large parking lots in shopping districts, college campuses, airport or sports facilities usually lack means for smart parking. Nevertheless, most parking lots are equipped with security video cameras that constantly monitor the environment. The proposed project will develop geospatial imagery analytics methods to extract open spots from parking lot video feeds and develop a mobile GIS app to map the open spots and guide drivers to the open spots. The UTD police has granted us access to real-time video feeds of parking lots on campus to develop geospatial imagery analytics methods and the LBS SpotFinder app. The methods and app will be transferable to other parking lots with video feeds.
1.4. Projected Project outcomes and deliverables	• Geospatial imagery analytics methods to extract open parking spots in a parking lot based on video feeds from security cameras • SpotFinder, an LBS mobile GIS app to map and guide drivers to open parking spots in a parking lot • Analyze occupancies of parking lots and parking needs in space and time

Signatures

Allen Wang 	Travis Bonneau 
Walter Han 	James Landry 
Md Siamul Islam 	
Company Mentor: Brent Dell	Faculty Advisor: May Yuan