# Spot


Team 12
Advisor: Professor Rakesh Vohra

Rohan Aggarwal (SSE), Scott Isett (SSE), Sam Leavitt (CIS), Kristen Tilley(SSE)

# Contents

# I. Executive Summary

For our senior design project, we chose to design Spot, an intelligent parking system. The motivation behind the project stems from the current state of parking in Philadelphia, where there are ample available street-side parking spaces but no system in place to help drivers find them. This is a significant inefficiency that causes not only frustration but also negative economic impact due to wasted time, fuel, and emissions caused by drivers having to circle around blocks looking for parking. The system we devised has three main components: a camera system, computer vision algorithm, and user interface. The cameras will be strategically placed on city blocks as to capture images of the on-street parking areas. The images taken by these cameras will then be analyzed using our computer vision algorithm which is able to identify cars parked in the designated on-street parking area, calculate the distances between them, and determine the availability of open parking spots. This information is then communicated to the user through our application, which allows them to view parking availability, select a parking area to be navigated to, and a host of other additional features.

Over the course of the semester, we were able to make significant progress on all three system components, along with successful final integration. Going into second semester, we had a semi-functional android application, computer vision algorithm that could only detect cars in parking lot-style spaces, and no functional camera. For our app, we decided to ditch the android interface and switched to developing a web app that could be optimized for mobile. We went beyond basic functionality and included a number of features such as color-coded parking availability mapping, Google Maps integration, user accounts, and a rating and feedback system. We also made significant progress on our computer vision algorithm; the algorithm is now robust enough to identify parked cars in on-street parking areas, calculate the distances between each, and generate the number of available spaces in that parking area. Finally, we decided to pivot away from developing our own camera and opted to use WYZE cameras for our system. This not only improved scalability, as these cameras are commercially available for a low price point, but it also provided additional functionality that helped us integrate the three components of our system. As demo day approached, we were able to integrate our code for both the application and algorithm so that images could be fed from the camera and successfully processed, allowing for real-time updates in our app, as demoed during our presentations.

# II. Project Overview

Spot is an intelligent parking system designed for implementation in densely populated urban areas with limited amounts of on-street parking in high-traffic areas. For the purposes of this senior design project, we chose to focus on one particular area, the city of Philadelphia, so that we could design our system with city-specific aspects in mind such as parking layout, laws and regulations, and transportation goals of local agencies.

The motivation for developing Spot was the issues that arise when commuters and other city drivers are unable to find available parking. According to a study conducted by the U.S. Department of Transportation, the average American driver spends about 17 hours per year looking for parking, and this number is higher in urban areas with more limited parking. This often-frustrating process of searching for available parking also has significant economic costs for both individuals and the economy as a whole. Searching for parking costs motorists $350 per year on average, and this adds to $73B nationally in wasted time, fuel, and emissions. Overpayment for parking is also a significant issue; when drivers are unable to find on-street parking within their desired location, they end up choosing much pricier options out of frustration. This costs the average driver about $97 per year, or $20B in the US annually. Finally, these estimates do not even include the negative externality of increases in congestion on the already busy roadways.

When we look more specifically at the city of Philadelphia, we find that parking issues are signficant despite ample parking options. In Philadelphia alone, there are about 2.2 million spaces, about 15,000 of which are on-street parking. However, since there is no system in place to help drivers locate these available spaces, drivers are unable to fully utilize available parking. This should not be an issue in a city that boasts almost 1.4 parking spaces per person, based on recent population data. For this reason, we felt that the pressing issue is not a lack of parking, considering the aforementioned fact that there are more parking spaces than people. Instead, we found it to be a problem of inefficient parking management, which is the problem our system is designed to solve.

Our solution to design Philadelphia's parking problem is a system is composed of three components: a camera system, computer vision algorithm, and user app. The first component is comprised of cameras placed on each block of the desired coverage area. The cameras will be posted on either street lights or traffic light to provide a bird's eye view of each block. These cameras will take images to be processed using the second component, our computer vision algorithm. The algorithm, after initial set-up of each camera during system implementation, will be able to be fed an image and identify cars parked in the on-street parking lanes and the space they take up, allowing us to calculate the availability of open spaces left on the street. Finally, this information is communicated to the end user through a simple interface. This application allows users to input their desired destination and view real-time parking availability. The user can then select their desired parking area and be directed to the available spots. Additional functionality of the application includes allowing users to create an account and save and rate their trips.

# III. Technical Description

## Specifications and Requirements

Two major considerations while developing our design were cost and keeping the system within the bounds of current city regulations. The project is designed for implementation in Philadelphia, making the city the main stakeholder. To make this project a viable and attractive offer for the city, we needed to keep implementation and upkeep costs down, which played a major role in the decision to implement a camera system. Also, during our idea development stage we considered different potential revenue models for the system, such as either charging membership fees to use the service or payment per spot, a portion of which would be going to the city. However, we found out that the city of Philadelphia has regulation in place that bans the parking association from making a profit off of street parking revenues. With this in mind, we decided to scrap the revenue models we originally considered to comply with Philadelphia standards.

## Solutions Considered

There were two major solutions we considered for parking availability data collection during the system design phase: a camera network or individual parking space sensors. We had to consider the cost of implementation, cost of maintenance, and general feasibility. Our analysis, as discussed further below, shows that a camera system has lower costs associated with both implementation and maintenance, along with being a more promising solution. This guided the rest of our design decisions.

In terms of cost, a camera system has significant advantages over a network of parking spot sensors. We evaluated overall cost by examining both implementation cost and maintenance cost. For implementation cost, we estimated that each sensor would cost about $24, and each block would require about 25 sensors, coming to about $600 per block in hardware alone. We were able to find a camera, the WYZE camera, that costs only $25 per camera, plus $8 for each mount. With about 6 cameras per block, this results in a per-block hardware cost of about $200, a third of the cost of sensor hardware per block. Besides the cost of hardware, the cost of implementation for sensors is significantly greater than that of cameras, since sensors would have to be placed in the ground instead of simply being attached to an existing pole as the cameras are. Finally, sensors are more susceptible to damage from hazards such as snow plows and individuals attempting to damage them. While cameras are also susceptible to damage and will need replacement on a rolling basis, they are out of reach of snow plows and most other hazards.

Looking at general feasibility of implementation, the camera system presents a number of advantages over a network of parking spot sensors. The biggest advantage in terms of feasibility is less disruption of traffic. Installing a camera on a traffic signal or lamp post has little to no effect on the flow of traffic, as streets do not need to be shut down. Sensors, on the other hand, will need to be installed on the street in each individual spot, meaning sections of the street or even the entire block may need to be shut down during installation. Since keeping roads open is in the best interest of the city, the camera system is once again the more favorable option.

## Societal and Economic Considerations

As our project will be public-facing, there are a few societal concerns that we have considered as we have developed our design: user-data privacy and cost of the design. We worked to creatively alleviate our concerns though by choosing design concepts that would both benefit our design and our stakeholders.

In terms of data privacy, as we are making a mobile application that users will use on their smartphones, we had to consider how to ensure that we were only going to take the data we specifically needed and that our application could not be for malicious purposes such as improper data collection. We thus made a conscious effort to not store user location data, instead giving the user an option to keep track of where they parked in their history if they choose.

As one of our key stakeholders is the City of Philadelphia, we wished to take into account the cost of our design as key consideration; if the city chooses to adopt our solution, the funding come from taxpayer dollars and we wanted to make our project low-cost to ensure that it is not a public burden. We did this through our design considerations. As we were initially deciding between using a network of sensors and a network of cameras to determine if a parking spot is open or not, we chose to implement a system of cameras as the city could use its existing security cameras as opposed to needing to install a new network of sensors at all parking spots in the city.

# Solution Overview

Our solution comprises of three components making up an integrated system: A camera network, a parking detection algorithm, and a user application.

## Camera Network

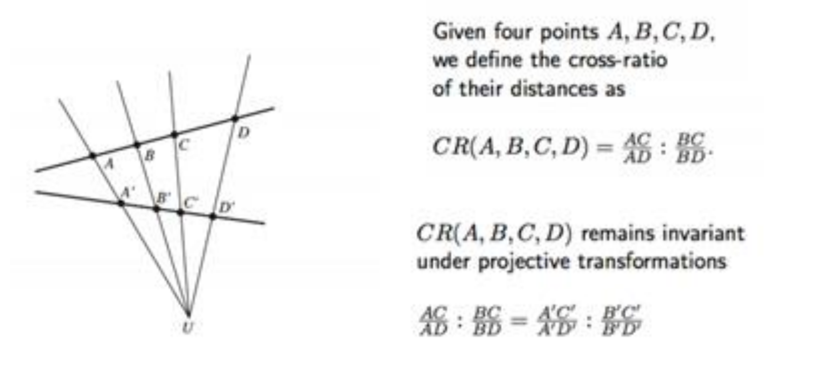
Our system currently uses a WYZE camera. The WYZE camera is designed for creating an in-home security system and allows a user to take photos remotely from a mobile app and store these photos on a phone or on the cloud. The camera is 1080p and connects directly to Wi-Fi with its own unique MAC address and IP address. Currently, the system is set to take pictures on user-input, stores the photo, and sends the photos via Dropbox to the device hosting our computer vision algorithm. We opted to go with Dropbox for integration between the camera and the computer vision algorithm due to its quick run time and lack of cost; images were transferred from the time they were taken to the computer in under ten seconds, and then loaded directly in our python-based Computer Vision algorithm.

Due to budget and practicality constraints, for our demonstration, we set up one camera over the stretch of Walnut Street in front of the Singh Nanotechnology Center. We did consider how the system would scale when designing the system, however. Our system is set up in a way such that information for each camera is stored in the database based on the IP address of the camera and we can easily add additional entries to the database. Each camera is low-cost and low-maintenance, and thus makes the costs of scaling predictable. Please see the budget and justification section for further discussion on scaling costs.

## Parking-Detection Algorithm:

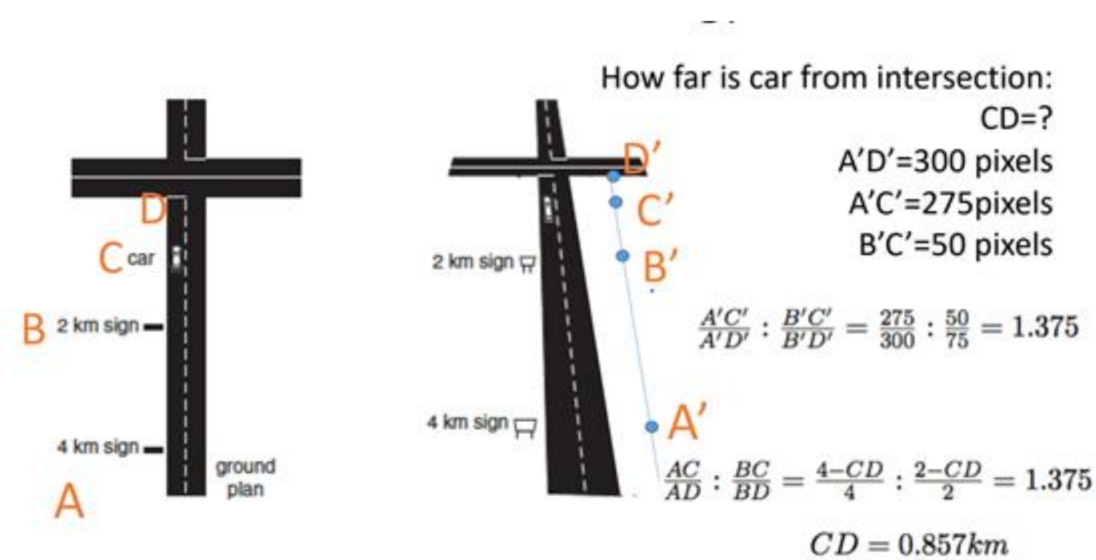
The next part of our solution involves detecting cars in the region we have already determined cars can legally park in. In order to accomplish this, we used an outside object detector called YOLO, that has already been trained on multiple types of objects. It is quite capable of detecting cars, trucks, bicycles, people and much more, as well as distinguish between them. However, there are a few steps to be taken before running the object detector on our image. First, using python, we pulled in the most recent photo in the Dropbox taken by the camera we are working with (which will be discussed later). Additionally, we have to obtain the coordinates of the boundaries of the region of interest from the database corresponding to the IP address of the camera. Now, we have all the information we need to run our object detector on the image. This process is done continually for each camera in our system.

Distance Calculation: Following the object detection, we needed to map the distance in pixels to a distance in feet between vehicles to determine if there was enough space for an additional vehicle. To do this, we reached out to a subject matter expert, Professor Kostas Daniilidis, who referred us to his teaching assistant, Kenneth (Ken) Chaney, for assistance. Ken recommend we use the Cross-Ratio Approach to map a distance in pixels to a distance in feet.




Brannan et al. Geometry

The Cross-Ratio Approach requires that we know the location of the two vehicles on the image and calculate the cross ratio as AC/AD: BC/BD (see image above) in pixels. Using a simplification that the vehicles of are the same size (12ft), we have values for A'B' and C'D' and can back out the value of B'C' using the fact that the cross ratio in feet equals the cross ratio in pixels, as the YOLO object detection algorithm provides locations the locations of vehicles in pixels. While this method does introduce error as not all vehicles are the same size, our discussion with Ken lead us to concluding this would be the best way to implement the calculation by balancing potential error with practical and time constraints.




After implementing the algorithm, we had to make an adjustment as we noticed that our errors were consistently in the same direction, that is they were consistently underestimating the distance between cars. To adjust for this, we set the distance threshold for an open spot to 10 ft. This helped reduce some of the error that our distance calculation was producing.

### User Application:

The final piece of our solution is a web app that allows users to quickly and seamlessly see the state of parking availability in real time. The process works as follows: Users first open up Spot. If it's their first-time using Spot, they have the option of creating an account; otherwise, cookies stored on their computer will keep them logged in and so they can proceed directly to the next step. Next, users enter their destination in a search box. A map then focuses in on their destination and shows the roads in the surrounding neighborhood as a series of colored lines, on a continuous spectrum of colors from red to green, with red indicating that most parking spots on a given are taken, green indicating that most spots on the street are available, and brown suggesting a moderate level of parking availability. Users can click on any line and see our precise calculation of the number of available parking spots as well as the theoretical maximum number of cars that can be parked on the corresponding stretch of street. Lastly, they click "Select and go!" and are redirected to Google Maps directions which direct them to the parking spot they selected. Afterwards, users can view their history and see each trip. They can see what Spot told them about the state of parking at the time of their trip (e.g. "there were 5 available spaces out of 12 possible"). They can then rate their trip, stating whether or not Spot gave an accurate description of parking availability on that stretch of street and alerting us to any potential problems, such as a fire hydrant, closed-down street, or temporary parking restriction.

Our app is built with the MEAN stack -- MongoDB, ExpressJS, AngularJS, and NodeJS. The application features responsive design -- that is, it works well on both mobile devices and on computers and laptops. It follows standard best practices in maintaining user cookies and history. On the back-end, it is linked with a MongoDB database with two relations -- "cameras" and "users". The former tracks each camera and stores data associated with it, such as the IP address (which serves as a unique ID for this relation), coordinates of the endpoints of the region of the image that captures a stretch of parking spaces, geographical coordinates of the road segment covered by the camera, the number of available spaces, and the number of total spaces. The latter -- "users" -- keeps track of users who have created accounts for the web app. It stores username, email address, password, and a list of trips -- each of which is a MongoDB subdocument with a timestamp, the number of open and total spaces Spot indicated were available to the user at the time, and optionally the user's rating and comments for the trip.

## Current Project Status and Testing Results

There are a few measurements relevant to our project to determine its current working status. First, in terms of the object detector that we are using, we want the probability that a car is detected given that we know there is a car in the image, and the probability of detecting a car when there is no car in the image. After looking at around 300 samples, the first ratio was determined to be around 94%. This is quite high, but missing a car be dangerous for our users since an open spot might be detected when there actually isn’t one. However, we anticipate that this number will improve as we increase the number of cameras in our system. Out of all the samples that we gathered, there were no instances of detecting a car when there isn’t a car, which is promising.

The next measurement, and arguably the most important, is the percentage of time we detect a spot when there is a spot. This ratio was not as accurate and came out to be around 60%. Although our distance algorithm still needs some adjustments, as we work with more images, we are confident that this number will improve as well. Additionally, we had no false positives in our measurements, so no spots were identified when there was no actual spot in the image. We do appreciate these results as we want to minimize type 2 errors as much as possible and be as conservative with our model as possible as to not fool drivers into thinking there are available spots.

# IV. Self-Learning

Over the course of the semester, our team has had to obtain new skills in order to develop our project. First, we needed to develop some knowledge in the realm of computer vision. Although we were using an outside source for object detection, we still had to acquire knowledge on how to correctly estimate distances of objects in images in order to detect open parking spots on street sides. Additionally, a couple of our team members was not particularly well versed in python and all of its libraries, so we had to spend time learning the language as we put together our project. In terms of the camera system we were using, we had to work out a way to capture and store images properly in order to pair the camera system with the computer vision portion of Spot. This led to application of basic systems practices and what works well in these integration scenarios.

| Department | Course | Skill Added |
|---|---|---|
| CIS | 110 | Basic programming knowledge |
| CIS | 120 | Basic programming knowledge |
| CIS | 450 | Databases and web development |
| CIS | 545 | Python |
| ESE | 111 | Systems integration |
| ESE | 204 | Testing, business case and systems scaling |
| ESE | 421 | Python, computer vision, and systems integration |
| EAS | 545 | Business practices in engineering projects |

# V. Ethical and Professional Responsibilities

Spot was developed to serve both a social and economic mission – to reduce congestion, wasted time, and fuel emissions by making cities smarter. It is a project that has meaning and significance in any country that has an abundance of cars and on-street parking. If implemented well, Spot can change the world for the better. However, there are several ethical pitfalls that could endanger Spot's social mission. We have carefully considered and addressed these pitfalls. We identified three main areas of concern with respect to our ethical and professional responsibilities: Data privacy, parking affordability, and driver safety.

To safeguard users' privacy, we only store the minimum amount of data needed for our app to function. We do not keep track of users' location data. While we do offer to keep track of users' parking history, this feature is entirely optional and users who opt out of this feature by declining to create an account are not locked out of any features as a side-effect.

Parking affordability is a crucial concern to millions. In Philadelphia, an annual parking permit currently costs $35 annually -- less than 10 cents per day. To ensure that our project doesn't contribute to a rise in parking costs, we decided not to implement any pricing features that would put us in the position of a market maker.

Safety is the single most important ethical responsibility of our project. Preserving human life takes precedence above all else, and to this end we designed Spot to minimize the likelihood of drivers using the app while on the road. When a user clicks on their destination, the app immediately closes and redirects to google maps directions to their spot, which is designed to be used by drivers through audio instructions. We recognize that users may nevertheless reopen Spot while driving, and we considered implementing safeguards which automatically disable the app when it detects that the user's device is in motion at a driving speed. However, we opted not to include this feature because it would be too restrictive, preventing passengers in the car from using the app and directing the driver to an open spot.

# VI. Advisor and Team Meetings

Throughout the semester, we made a conscious effort to meet regularly both with our advisor, Professor Rakesh Vohra, and as a team. As per last semester, our advisor opted for us to write weekly progress emails along with periodic meetings on a per-need basis. For the weekly progress emails, we included an update on what was accomplished in the previous week as well as what we intended to complete in the upcoming week. This helped keep us on track and created a sense of accountability between team members as we needed to be sure to have updates to share with our advisor. For in-person meetings, we attempted to schedule these roughly once a month, and this allowed us to schedule them to coincide with the three in-class demonstrations in the case that Professor Vohra could not attend. In addition to meetings with Professor Vohra, we had on-need meetings with subject matter experts. To assist with our computer-vision algorithm, we reached out to Professor Daniilidis, a Penn professor well known for his work in the subject. He redirected us to his teaching assistant, Kenneth (Ken) Chaney, who we met with in person and via email. In the previous semester, we also met with Professor Ryerson and with a contact of hers, Andrew Stober, an employee of the University City District.

As a team, we chose to plan weekly meetings early in the semester to discuss progress and upcoming deadlines. These meetings were not scheduled on a set date or time, but rather scheduled each week to work around everyone’s evolving schedules. This semester, we also made an effort to meet in smaller groups as it was a goal from the previous semester to split the workload up better, so we could work outside of meetings more productively. We chose to have Kristen and Sam work on the mobile app and database aspects of the project while Scott and Rohan worked on the computer vision algorithm and camera system. This was especially helpful towards the end of the project, as we were able to work on multiple aspects at the same time and make more progress than we would have all working on the same part of the project at the same time.

# VII. Proposed Schedule with Milestones

| Date | Milestone |
|---|---|
| Week of 2/4/19 | Finished initial prototype of web app, including google maps integration and user functionality. |
| Week of 2/11/19 | Investigated other camera options, ordered WYZE camera |
| Week of 2/18/19 | -- |
| Week of 2/25/19 | Took parallel parking test images |
| Week of 3/4/19 | ~Spring Break~ |
| Week of 3/11/19 | Devised approach for computer vision algorithm and began initial stages of development. Built a tool for outlining parking spaces and uploading the endpoints to a database. |
| Week of 3/18/19 | Added features to app: displaying parking availability with lines on the map |
| Week of 3/25/19 | Program camera so images can be processed by algorithm. Implemented YOLO object detection. |

| | |
|---|---|
| Week of 4/1/19 | Added features to app: User history and ability to rate past trips. Performed extensive testing and bug fixing for the app.<br>Implemented the distance calculation used in our CV algorithm. |
| Week of 4/8/19 | Added features to app: About page, redirection to google maps directions upon selecting a parking spot, and finished linking history/ratings with the database |
| Week of 4/15/19 | Set up image transmission and storage. Integrated app and computer vision code to fully integrate system. Prepared for final presentation and demo day. |

# VIII. Teamwork

Teamwork was a crucial element of our project. Our end product was the result of constructive synergies among the four of us, each of whom have slightly different areas of experience and expertise. Owing to these differences, we decided in the Spring semester to divide work into subgroups of two, with Scott and Rohan primarily working on the computer vision algorithm and camera , and Sam and Kristen working on the user app. This arrangement marked a shift from the fall semester, where each individual was in two subgroups -- Sam and Scott working on the computer vision algorithm, Sam and Rohan working on the app, Scott and Kristen working on the camera system, and Kristen and Rohan working on outreach to experts and city officials. We found that putting ourselves on a single, more encompassing subgroup enabled heightened accountability and responsibility. Moreover, this streamlined structure facilitated additional subgroup-specific meetings.

# IX. Budget & Scaling Considerations

Last semester, our budget consisted of parts primarily related to the camera system: a Raspberry Pi V3 with Starter Kit ($84.95) and a Raspberry Pi Camera Board ($29.85). This brought our total budget to $114.90.

This semester, we made a few adjustments. Instead of using the Raspberry Pi and Raspberry Pi Camera for our camera system, we shifted to using a WYZE Camera ($25.98) and a hosting the computer vision algorithm on our device. The WYZE camera is Wi-Fi enabled, making for an easier integration process, and eliminating the Raspberry Pi allowed for a more scalable system as it cut out much of the per unit cost. These changes brought our total budget for the semester to $140.88, although our final system only used $25.98 worth of the budget, reducing our per-unit cost by 77% from the previous semester.

In terms of a system cost for a fully-scaled product, we created a projected budget for what the system would cost if installed on an area the size of the University of Pennsylvania's campus. We considered the cost of parts (cameras, mounts, and power supplies), labor (installation), and maintenance (replacement parts and labor) and found the system would cost $9,072 to install and $3,640 annually to maintain. For a detailed breakdown of our assumptions, please see the appendix.

# X. Standards and Compliance

| Standard | Description | How project is impacted |
|---|---|---|
| E2259 from the ASTM | Standard Guide for Archiving and Retrieving Intelligent Transportation Systems-Generated Data | Must ensure that we document all data, that it is stored in electronically readable media, and data related to an archive must be unique to that archive (open space data for drivers, ticket data for parking authority, etc). |
| ISO / IEC 16085:2006 | Risk management of life-cycle process for systems / software engineering. Framework to ensure that risk is managed before incident occurs | In terms of our project, we need to make sure that all data is only being used for its designed purpose, ensure that no one outside of our organization has direct access to the data we collect, and the camera system is accurately planned out as to avoid tampering. |
| ISO 12207 | General framework for process of software lifecycle (organizational, development, maintenance processes) | For our project related to this standard, we want to make sure that software checks are being done on our CV algorithm in detecting spots. Our review section of our app will aid greatly in this process. |
| ISO/ TC 42 | Family of standards related to image taking (measurements, recommendations, specs, security) | The images our cameras take will have to comply with all of these standards in terms of their dimensions, clarity, color scheme, and storage process. |

# XI. Work Completed Since Last Semester

Since the last semester, we focused on breaking the workload into four distinct parts: Camera System, Computer Vision, Web App, and Integration. Each of the first three parts could be worked on individually, while the fourth part required that the first three be working. For each of the parts, our progress since last semester was as follows:

**Camera System:** After the fall semester, we chose to switch to using a WYZE camera, on the recommendation of Sid Deliwala, in place of Raspberry Pi Camera we used last semester. We chose to make this switch as the camera is Wi-Fi enabled and allowed for easier integration into our system, as the images taken could be transferred using Dropbox as opposed to having to connect the Raspberry Pi Camera to a Raspberry Pi device and then connect the device to Wi-Fi and then transfer the images. We were able to set up a working camera and for our final demonstration, we took live images from the street in front of the Singh Nanotechnology Center, which sent the images to our computer vision algorithm.

**Computer Vision Algorithm:** Last semester, our computer vision algorithm worked on in specific conditions on perpendicular parking spaces. This semester, we focused on making the algorithm more robust and for it to work on the parallel parking case. The first addition we made was adding a web application for drawing in the region of parking zones. When the camera is first set up, the user loads a sample image of the camera region and selects the parking region of interest within the image, whose coordinates are stored in database. Whenever a new image is taken, the algorithm pulls the coordinates from the database, uses the YOLO object detection algorithm to find vehicles in the region, and then uses our distance calculation algorithm to determine the number of open spots. The number of spots is then input into our database and displayed on our user application.

**User Application:** Last semester, we had planned to use Android SDK to write a mobile app for user interaction. At the end of the semester we had integrated the Google Maps API into our app but had not made much progress beyond that. This semester, we decided to shift to using a mobile-friendly web application, as it would work on both IOS and Android, and would allow us to use our existing capabilities in using the MEAN stack as opposed to having to continue to learn Android development from scratch. This semester, we rebuilt the app to include a login and logout system, an area for the user to enter the destination and link to Google Maps to route the user to the spot, a mechanism for displaying how many open spaces are available by each camera, a history and review page, and an about page.

# XII. Business Plan

## Problem/Need

Spot, our intelligent parking system, was developed to meet the needs of Philadelphia drivers while looking for available parking. According to a study conducted by the U.S Department of Transportation, the average American driver spends about 17 hours per year looking for parking. This issue may be even more prevalent in densely populated urban area, such as Philadelphia. Upon further research into Philadelphia’s unique parking situation, we found there are about 2.2 million spaces, about 15,000 of which are on-street parking spots. This led us to believe that the problem facing the city is not a lack of parking; there is an abundance of parking, but drivers are unable to locate the available spaces. Our parking spot identification system was developed to address this problem.

## Value Proposition

There are significant negative economic effects associated with drivers having to search for parking. As previously mentioned, the average American driver spends about 17 hours per year circling blocks in search of parking. This costs individual motorists upwards of $350 per year on average, which translates to $73B nationally in wasted time, fuel and emissions, making it a significant economic and environmental issue (Inrix). This inability to find parking spawns additional negative economic effects as motorists are discouraged from driving to retail areas when they think they may have trouble securing a spot, negatively impacting local businesses. Less quantifiable, but still worth noting, is the negative effect on drivers from the frustration caused by this process. As frustration mounts, drivers are more inclined to overpay for parking by opting for expensive garages or lots, which costs the average driver an additional $97 per year (Inrix).

By implementing a parking spot identification system, drivers will spend significantly less time searching for parking and feel more confident driving to retail areas with limited on-street parking. According to our analysis, our intelligent parking system has the potential to reduce the average number of hours drivers spend per year searching for parking by over 75%, down to 4 hours from 17, based on the time it takes to utilize Spot and identify potential parking areas.

Spot also presents a unique value proposition in that is it well-aligned with the mission of the Philadelphia Parking Association (PPA) and fulfills a number of the goals outlined in Philadelphia's Strategic Transportation Plan. The PPA's states mission is to "contribute to the economic vitality of Philadelphia and the surrounding region by effectively managing and providing convenient parking," which entails "implementing cutting-edge technology to improve the customer experience and enhance overall management and agency efficiency" (PPA). Spot, as an innovative parking spot identification system, fits these criteria well and can greatly improve the efficiency when it comes to drivers being able to find available parking. Our system also meets a number of goals laid out in the Strategic Transportation plan including supporting communities and commerce with efficient transportation systems and delivering transportation services efficiently and transparently (Office of Transportation).

## Stakeholders

Due to the nature of our system, there must be two main stakeholders: the entity responsible for implementing and maintaining the system, and the party that is utilizing the system in the form of a service. In this way, the city of Philadelphia is a major stakeholder since they must approve, implement, and maintain the system, and the second major stakeholder group is the citizens of Philadelphia, each of which we will describe in further detail.

The city of Philadelphia is, as stated, a major stakeholder in the project. The intelligent parking system we created cannot serve its intended purpose without a municipality agreeing to allow it to be installed. Therefore, the system had to be devised in such a way that we could convince the city of Philadelphia to approve and implement the system. Although we were unable to get through to the officials at the PPA that would be able to hear about and potentially approve such a thing, we designed our system in such a way that it is well aligned with the PPA's mission and strategic goals, as discussed above.

The second major stakeholder group are Philadelphia drivers. This is a group that has been loudly voicing their grievances about Philadelphia's parking situation, and we believe that backing from citizens would greatly help our chances of getting approval from the city. With that said, this is the group we considered while designing our app, working to add additional functionalities that would better serve them such as directing them to their spots, allowing them to review their experiences and provide feedback, and have their own account. We also worked to make the user interface simple and the experience as seamless as possible for the drivers who would be using it.

## Market Research, Size and Growth

To understand the size of the market we would be serving, we turned to census data. From our research, we found that Philadelphia County sits has among the highest number of commuters coming from another county in the nation, with about 253,000 workers commuting in daily on top of those who live and work in the county (US Census Bureau). Although the percent of commuters using public transportation in Philadelphia is significantly higher than the national average, about 25% compared to 5% nationally, the majority of commuters still either drive alone or carpool to work each day. This 59% majority that take a car to work represents about 343,000 workers. Now, of course not all of these 300,000+ workers are driving into the downtown area each day, as the census data covers all of Philadelphia county. However, we feel these numbers are indicative that there is a significant market for our parking system.

In terms of growth, we only expect our market to grow as Philadelphia continues its upwards trajectory in terms of population. The city of Philadelphia is now in its 11$^{th}$ straight year of growth after experiences previous decades of decline (Lubrano). Additionally, Philadelphia itself, our target market, has been outpacing the population growth of its suburban Philadelphia counties at a rate of 3.4% last year compared with only 2.1% growth in the suburbs. Based on these statistic, we expect our target market to continue to increase in the near future.

## Customer Segmentation

In terms of customer segmentation, we see three main segments: commuters, consumers, and locals. As previously discussed, Philadelphia has some of the highest commuting rates in the nation, so commuters would likely make up the majority of our target market. In terms of consumers, we are including any individuals who are seeking parking in order to browse local retail, dine at one of Philly's many restaurants, or simply explore what the city has to offer. We believe this will be our second-largest segment, as Philadelphia boasts plenty of retail shops, restaurants, and other attractions. Our third and smallest segment is locals seeking out parking when returning to their residence. The reason we believe this to be a small subset of our target market is due to residential parking permits and regulations which allow these drivers to have much more predictable and reliable parking availability compared to our two other segments.

## Competition

Spot's major competitor, and the only other app that provides info on available public street parking, is SpotAngels. SpotAngels is an app that relies on crowd sourced data from its users, meaning its only source of availability knowledge comes from users reporting to the app while using it. Its interface is very similar to that of Spot in that the basic procedure for a user is to open the app, search for parking near their desired location, select a desired parking area, and be directed to it. In this way, Spot and SpotAngels share all the same basic functionalities. Where they differ is how each sources their information. Spot, as described in our report, utilizes a camera system along with a computer vision algorithm to determine the parking availability on street blocks. SpotAngels does not have any infrastructure in place and relied on its users reporting about availability on street blocks, which is limited since users are not incentivized to provide this information and typically only provide info when they arrive at their parking location. In order to be competitive, Spot needs to provide more accurate parking information, which we are confident that it can.

There are a number of other apps that provide parking information in Philadelphia such as the more well-known and widely-used SpotHero and Parkwhiz. However, these apps only provide data on parking lots and garages, not on-street parking. Since providing on-street parking information is our only service, we do not view these other parking apps as direct competitors.

# XIII. Discussion and Conclusion

Today, we have a fully-functional smart parking system with three interconnected components -- a camera system, computer vision algorithm, and web application. We've built an algorithm that can detect the availability of street-side parking in a given image, with accuracy, robustness, and speed. We developed a feature-rich web app which includes responsive design, an intuitive user interface, and a seamless overall experience. Lastly, we've linked up the various systems we used via a speed- and memory-efficient system of databases and through the camera's integration with Dropbox.

Should we continue to work on the project past our graduation date, we would first and foremost work on further improving the accuracy and robustness of our distance calculations. We would expand the app to offer users more options, such as displaying the price of parking. We would work on improving the design and aesthetics of the app. And lastly, we would actively seek municipal partners to help us bring Spot from vision to reality.

Spot has offered us a wealth of learning experiences. In addition to valuable technical knowledge in computer vision, web development, systems integration, and more, Spot taught us what it takes to manage a project, coordinate as a team, and interact with important municipal and university officials.

# XIV. Citations

Inrix. “Searching for Parking Costs Americans $73 Billion a Year.” *PR Newswire: Press Release Distribution, Targeting, Monitoring and Marketing*, 26 June 2018, www.prnewswire.com/news-releases/searching-for-parking-costs-americans-73-billion-a-year-300486543.html.

Lubrano, Alfred. “Philadelphia's Population Increased in 2017, but It's Not All Good, According to Census Data.” *Https://Www.philly.com*, The Philadelphia Inquirer, Daily News and Philly.com, 18 Apr. 2019, www.philly.com/philly/news/philadelphia-population-census-data-20180322.html-2.

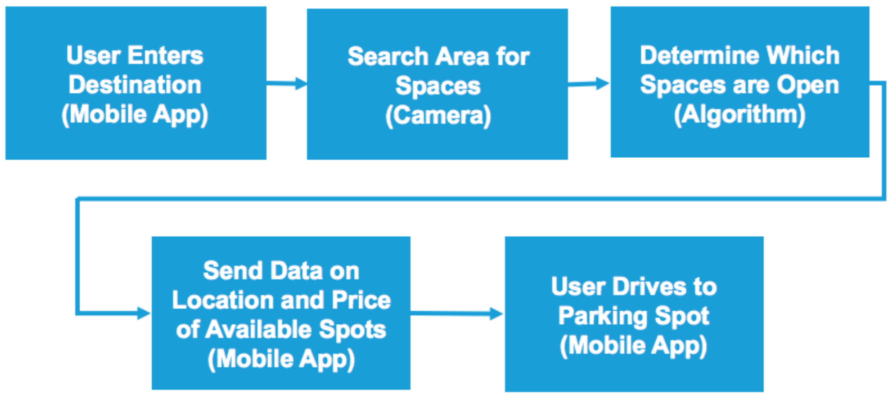
Office of Transportation. “City Releases CONNECT, Philadelphia's Strategic Transportation Plan | Office of Sustainability.” *City of Philadelphia*, Office of Transportation, 10 Oct. 2018, www.phila.gov/2018-10-10-city-releases-connect-philadelphias-strategic-transportation-plan/.

PPA. “About PPA.” *The Philadelphia Parking Authority*, www.philapark.org/about-ppa/.

US Census Bureau. “253,000 Workers Commute into Philadelphia County, Pa., Each Day.” *The United States Census Bureau*, 6 Sept. 2014, www.census.gov/newsroom/press-releases/2013/cb13-r20.html.

# Appendices

## Solution Overview




## Scaling Consideration

| Total Initial Investment | |
|---|---|
| Cameras | 4,200 |
| Mounts | 672 |
| Power Sources | 1,680 |
| Labor | 2,520 |
| **Total Cost** | **$ 9,072** |

| Total Annual Investment | |
|---|---|
| Replacement Parts | 1,120 |
| Maintenance | 2,520 |
| **Total Cost** | **$ 3,640** |

| Assumptions | Value | |
|---|---|---|
| Camera Cost | $ | 25 |
| Mount Cost | $ | 8 |
| Power Source Cost | $ | 30 |
| Hourly Labor Cost | $ | 15 |
| Blocks | | 14 |
| Street Sides Per Block | | 2.00 |
| Cameras Per Street Side | | 6.00 |
| Mounts Per Camera | | 0.50 |
| Power Supplies Per Camera | | 0.33 |
| Installation Hrs per Camera | | 1.00 |
| Annual Maintenance Hrs Per Camera | | 1.00 |
| Annual Camera Replacement Probability | | 0.17 |