

Ticket Torpedo

HCI 575 PROJECT

Team 02 Members:

Justin Wheeler

Jakub Hladik

Tim Lindquist

Revised: Date 4/25/18, Version 3

Table of Contents

1 Introductory Material	6
1.1 Acknowledgement	6
1.2 Problem Statement	6
1.3 Operating Environment	6
1.4 Intended Users and Intended Uses	6
1.5 Assumptions and Limitations	7
1.6 End Product and Other Deliverables	7
2 Approach and Statement of Work	9
2.1 Objective of the Task	9
2.2 Functional Requirements	9
2.3 Constraints Considerations	9
2.4 Previous Work And Literature	10
2.5 Previous Experiences	10
2.6 Proposed Design	10
2.6.1 Hardware	11
2.6.2 Software	11
2.6.2 Computer Vision Algorithms Used	12
2.7 License Plate Confidence	12
2.7 GPS Position Acquisition	13
2.9 Estimating Target Vehicle Location	14
2.10 Matching Parking Spots to the Estimated Target Vehicle Location	15
2.10.1 Plate Perpendicular Distance	16
2.11 Technology Considerations	19
2.12 Safety Considerations	20
2.13 Task Approach	20
2.14 Possible Risks And Risk Management	21

2.15 Project Proposed Milestones and Evaluation Criteria	21
2.16 Project Tracking Procedures	21
2.17 Expected Results and Validation	21
2.18 Test Plan	21
3 Project Timeline, Estimated Resources, and Challenges	23
3.1 Project Timeline	23
3.2 Feasibility Assessment	23
3.4 Other Resource Requirements	24
3.5 Financial Requirements	24
3.6 Results	24
3.8 Lessons Learned and Improvements	26
3.9 Web camera and Obtaining Images	26
3.10 GPS Position Estimation	26
3.11 License Plate Recognition	26
3.12 Environmental Conditions	26
3.13 GUI	26
4 Closure Materials	28
4.1 Conclusion	28
4.2 References	28
4.3 Appendices	29

List of Figures

Figure 1: Concept Operation Vehicle

Figure 2: Concept In-Progress Operation

Figure 3: Concept Operation Report

Figure 4: Project Pipeline

Figure 5: Project Timeline

Figure 6: Map of Howe Parking Lot

Figure 7: Plate to Plate Confidence Computation

Figure 8: Recorded GPS path through Howe Parking Lot

Figure 9: Car of Interest Position Measurement

Figure 10: Raw Distance Data Collection and Interpolation Fit

Figure 11: Distance Error Comparison

Figure 12: Distance Geometric Relation

Figure 13: Distance Calculations

Figure 14: Spot matching

Figure 15: Height and Width Calculation

Figure 16: System Output

Figure 17: System Information

Figure 18: Calculated Information

Figure 19: Generated Map

List of Tables

Table 1: Sample Parking Lot Scenario

Table 2: Hardware costs

List of Definitions

ROI: Region of Interest

GPS: Global Positioning System

WAAS: Wide Area Augmentation System

ISU: Iowa State University

AMS: AutoVu Management System

OCR: Optical Character Recognition

LPR: Licence Plate Recognition

1 Introductory Material

1.1 ACKNOWLEDGEMENT

This project was inspired by the technical information on computer vision provided by Dr. Alexander Stoytchev. Assistance and additional motivation from ISU parking was also received from a meeting with Mr. Miller, Director of ISU Parking Division.

1.2 PROBLEM STATEMENT

Iowa State University's current parking violation system requires a worker to manually check each car for a parking permit. If a violation is identified, a physical ticket must be generated along with a photo for proof of the incident. This process bottlenecks the amount of cars seen by the worker and leaves a large amount of cars undetected. Because of this tedious approach, many students do not feel threatened to park in illegal spots, resulting in overcrowding of the lots. Our goal is to automate and streamline the process of detecting violations and delivering tickets to increase throughput. By doing so, we hope to make it easier to find parking on campus. We accomplished this using computer vision to detect licence plate numbers that are then compared to a list of allowed cars for the specific parking lot. If a violation is detected, a flag is set on the map in the output video. This system will be mounted to the workers vehicle and require only a little user setup to operate.

1.3 OPERATING ENVIRONMENT

The end product is a camera and GPS mounting system attached to the roof and window of a vehicle. It is able to withstand outdoor elements and adapt to different lighting conditions.

Before starting, a concern to our group was the speed at which the car could travel to adequately take a picture every foot. It was found that if the car traveled 15 mph (parking lot speed limit) it would need a camera with a frame rate of at least 22 fps and a computer which can process those frames at 22 fps as well. However, after some testing, we found that 15 mph is too fast not only because of the fixed camera frame rate, but it is also a safety hazard. Speed of 5 mph seemed a reasonable speed for checking for license plate violations.

All of the computations were done on a laptop inside the car.

1.4 INTENDED USERS AND INTENDED USES

The end user of Ticket Torpedo is intended to be ISU parking division or any other parking regulatory agency. With this in mind we tailored our front end user interface to be as friendly to use as possible. The user will drive through a parking lot while using our

system to generate a output video that reports all cars that don't have permits for the current lot.

1.5 ASSUMPTIONS AND LIMITATIONS

The end product is capable of reading only clean/visible license plates. Any license plate which is covered by mud, snow, or any foreign object obstructing the view will be ignored. Some special purpose parking spots are able to be detected, but currently treated as a normal parking spot. This includes the following parking spots: parking meters, time limited, emergency services, and others which don't require a permit for the entire parking lot. The speed of the enforcement vehicle is limited to five miles per hour to achieve the most accurate results of multiple frames on each plate and ensure overall safety during the driving and scanning process.

The data being used to compare each license plate is stored in a local database that is manually populated. This is due to the lacking privileges to access the ISU Parking Division database. The ISU legal team deemed the release of the license plate to parking spot information too risky and denied our request. The local database contains real data which was obtained by walking around the parking lot and retrieving licence plate numbers.

1.6 END PRODUCT AND OTHER DELIVERABLES

The end product is a hardware and software system placed on an enforcement vehicle that detects licence plates while cross checking the results with a database to detect violations. The report consists of a video file which contains information on each frame. The frame number, timestamp, latitude, longitude, and vehicle speed for the current frame are always shown. Information regarding the vehicle detected has two states, instantaneous and final. The instantaneous reading shows the current reading from OpenALPR and its confidence level. After scanning the entire car, all the readings are merged together to create histograms for characters frequencies to acquire the best result.

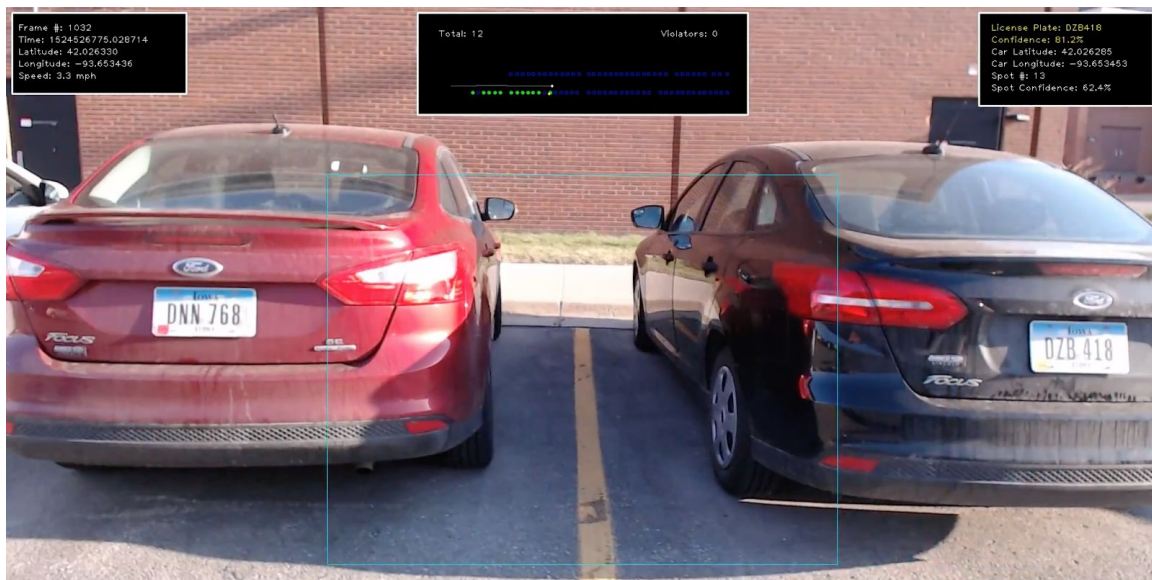


Figure 16: System Output

```

Frame #: 387
Time: 1524526752.398971
Latitude: 42.026334
Longitude: -93.653856
Speed: 3.5 mph

```

Figure 17: System Information

After scanning the whole plate, the final results are shown, including the plate number, confidence level, car latitude, car longitude, parking spot, and parking spot confidence. The shown counter for total cars scanned in the middle of the video will also increment. If the vehicle's license plate is not in the local database, the total violations counter will get incremented.

```

License Plate: GCG114
Confidence: 82.4%
Car Latitude: 42.026289
Car Longitude: -93.653844
Spot #: 2
Spot Confidence: 78.8%

```

Figure 18: Calculated Information

In the middle, there is a simple map which shows the trajectory of the scanning vehicle and the parking positions. Each parking position has three states: red for violation detected, green for valid vehicle, and blue for unknown or unscanned (or empty, at the end of the scan).

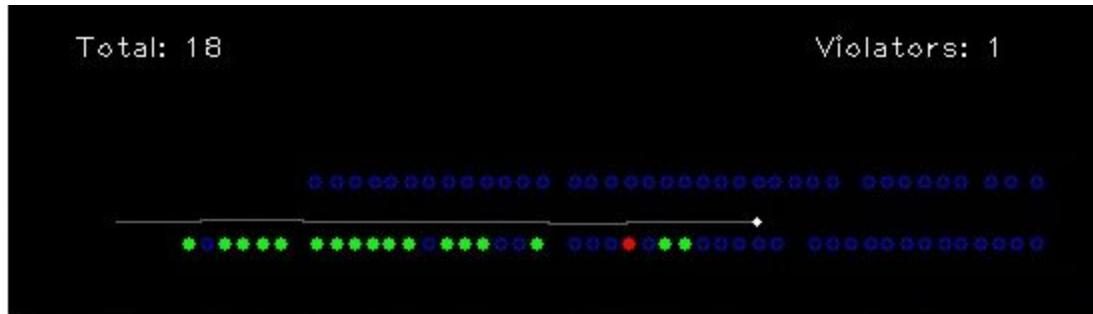


Figure 19: Generated Map

2 Approach and Statement of Work

2.1 OBJECTIVE OF THE TASK

The goal of this project is to decrease the amount of time it takes to check an entire parking lot for cars that shouldn't be parked there. It would also be less prone to human error by increasing the detection rate and increasing the amount of revenue the ISU Parking Division would receive. Due to the higher detection rates, students might be more willing to buy parking passes increasing the legal effective parking lot utilization. It might also turn away some parking violators because the detection rate would increase.

2.2 FUNCTIONAL REQUIREMENTS

The functional requirements of our project includes the following:

- Accurately interpret licence plate numbers
- Match target car location with GPS coordinates
- Ability to acquire usable data at parking lots driving speeds (5 mph)

2.3 CONSTRAINTS CONSIDERATIONS

Constraints of our project includes the following:

- 5 mph is the max speed in a parking lot
- GPS accuracy of 0.9 meters to work with
- Camera resolution
- Parking lot irregularities
- Various lighting conditions
- Frame rate of the camera

- Computational power

2.4 PREVIOUS WORK AND LITERATURE

To gain the needed background information and do our market research, the team set up a meeting with Mr. Miller, the Director of ISU Parking Division. He informed us that over \$300,000 in revenue is generated each year from issued parking tickets and meters on Iowa State University's Campus. Mr. Miller also gave us information of a system, similar to our proposed description, that was implemented at the University of Wisconsin-La Crosse. The system is called AMS and costs over \$45,000 to implement. According to a case study, UW-La Crosse saw a increase in revenue by 10% each year with 150 fewer citations and the ability to issue 200 more permits [1]. Some features this system has is that it is capable of scanning cars at 40 mph, and able to identify licence plates from parallel parked cars.

Our group also explored some research articles on the subject of licence plate recognition. One in particular was from Aboura and Al-hmouz that recommend a strategy for going about LPR. Their approach recommended first localizing where the plate is on a vehicle, then segmenting the plate to extract characters and finally used processing to recognize the characters [4]. We discovered that a similar flow was used in OpenALPR [6] implementation and it became the best candidate for us to use.

2.5 PREVIOUS EXPERIENCES

Our group had a wide variety of skills that was used to accomplish this task:

Tim Lindquist is a graduate student in Electrical Engineering. He has a background in hardware and software integration from his internships and research experience.

Justin Wheeler is a senior studying Computer Engineering with a focus on software. He has gained more software experience from his intern and co-op experiences as a software engineer.

Jakub Hladik is an undergraduate student in Computer Engineering with a focus in FPGA and ASIC development. He has past experience in GPS data acquisition and processing, and OpenCV.

2.6 PROPOSED DESIGN

In this section we will discuss possible solutions and design alternatives to this project. The first solution the team came up with was to eliminate physical permits and have the licence plate serve the purpose. This is beneficial as it cuts down on labor and material costs. It is also more fraud-resistant and easier to manage since each is unique to the car.

2.6.1 HARDWARE



Figure 1: Operational Vehicle

For handling the computer vision system the team proposed mounting two cameras on the exterior of the enforcement vehicle. As shown in Figure 1 above, the camera will be mounted in horizontal orientation on the enforcement vehicle. The GPS unit will be attached to the top of the vehicle. The laptop will be inside the vehicle for the driver to easily see while stopped. See Figure 1 for a mock setup of the operational vehicle.

2.6.2 SOFTWARE

Data is extracted from each vehicle using OpenALPR. The extracted data is then cross compared with a local database of valid cars in the Howe parking lot. If the license plate read is not in the local database of valid license plates, the number of violations will increase and the mini-map on the report will show the parking spot filled with a red circle. All the necessary data will also be on the frame in the report such as the latitude and longitude of the capture vehicle, speed of the capture vehicle, unix time for current frame, and the confidence for each reading. These confidences include the plate detection confidence and parked vehicle parking spot confidence. Our system is capable of finding which parking spot the parked vehicle is within. The ability to estimate the parking spot

the parked car is giving our system the ability to account for irregular spots such as handicap or emergency vehicle by having the locations mapped out.

2.6.2 COMPUTER VISION ALGORITHMS USED

The license plate localization and optical character recognition will be done by OpenALPR library [6]. The position estimation will be done using OpenCV [7] and projection estimation from the geometry of the license plate which never changes. The approach to estimate distance is derived from a model created using raw distance capture data. We are accurately interpolating a fit line to predict measurements based off plate dimensions in the frame. The angle to the license plate from the camera lense will be extracted from simply the position of the camera lense, the center of the image, relative to the position of the license plate in the image.

The camera will capture frames and run them through OpenALPR library algorithm to determine whether there is a license plate in the picture. If there is, it will continue and detect the bounding box for the license plate and then read it. With the capturing vehicle traveling around 5 mph, the license plate should get read about 15 times. These additional reads will help further boost the confidence measure of the neural network which performs the license plate search and read in OpenALPR.

Once the license plate is extracted, the position estimation takes place. The GPS position data is matched with the frames and the new geographical coordinate of the target license plate is calculated.

By obtaining this data, the final product will check against a database of parking spots and license plates which are allowed in the particular spot. If there is no matching entry, the particular vehicle will be matched as a violator.

2.7 LICENSE PLATE CONFIDENCE

A simple measure of how well our system is working is to display a confidence reading along side the plate results. Confidence is a measure of how much discrepancy is seen while calculating a final plate number. Discrepancy comes in two forms. 1.: On a plate level it is the measure of how close of a match each letter is to a predefined training set. 2.: On a plate to plate level it is a measure of how multiple readings match one another. Measure 1. is easily implemented by accessing result data from the plate interpretation functions. Measure 2. is made by looking at the total measures of a plate while it is in the region of interest.

$$\frac{\sum_{col=1}^{\max(hist(length))} \frac{\maxIndex(hist(platesChar[col]))}{numplates}}{\maxIndex(hist(length))}$$

Plate=ABC123

numPlates=4

platesChar=

Col	1	2	3	4	5	6
Plate1	A	B	C	1	2	3
Plate2	A	B	B	1	2	-
Plate3	A	B	C	1	2	3
Plate4	B	B	C	L	3	3

hist(length)=

Length	1	2	3	4	5	6
Amount	0	0	0	0	1	3

maxIndex(hist(length))=6

hist(platesChar[col])=

Ascii	0	1	255
Amount	X	X	X

Figure 7: Plate to Plate Confidence Computation

The figure 7 above illustrates how the plate to plate confidence level is reached. For a batch of plates the program will analyze character “n” in all the plates and get a percentage of the dominate occurrence relative to all the plates. The percentages are summed up and divided by the number of plates and then divided by the total length. This confidence level shows how many of the plates displayed the same data.

Our overall confidence reading is reached by a summation of “plate level” confidences divided by the total plates in the batch multiplied by the “plate to plate” confidence level.

2.7 GPS POSITION ACQUISITION

We used a cheap USB GPS receiver with U-Blox 7 chipset which supports WAAS (Wide Area Augmentation System). WAAS is needed in order to achieve the 0.9 meter accuracy. We logged the GPS position output into a file with timestamps for later processing. Data can be interpreted in Google Earth mapping software as shown in the figure below.



Figure 8: Recorded GPS path through Howe Parking Lot

2.9 ESTIMATING TARGET VEHICLE LOCATION

The GPS receiver provides us with a coordinate accurate to 0.9 meters every second. We log these coordinates as we capture the video. Since the processing is done after the recording and the scanning vehicle speed is fairly slow and constant, the coordinates can be simply interpolated. We obtain the interpolated instantaneous position of the scanning vehicle and use it to calculate the location of the target vehicle.

From the camera feed, we calculate the perpendicular distance to the plate using the known license plate geometry and the size of the license plate in pixels. We calculate angle θ , the angle between the camera line and the line from camera to the plate (marked as “Distance to the plate”), by measuring the license plate offset from the center of the picture. The distance to the plate gets calculated similarly, using the perpendicular distance and the offset. By knowing these parameters, we estimate the approximate coordinate of the license plate. Later, we apply length of 2.5 meters in similar fashion to find the approximate coordinate of the center of the vehicle. Formulas for estimating new coordinates based on the given information were obtained from Edward William’s website “Aviation Formulary V1.46” [2].

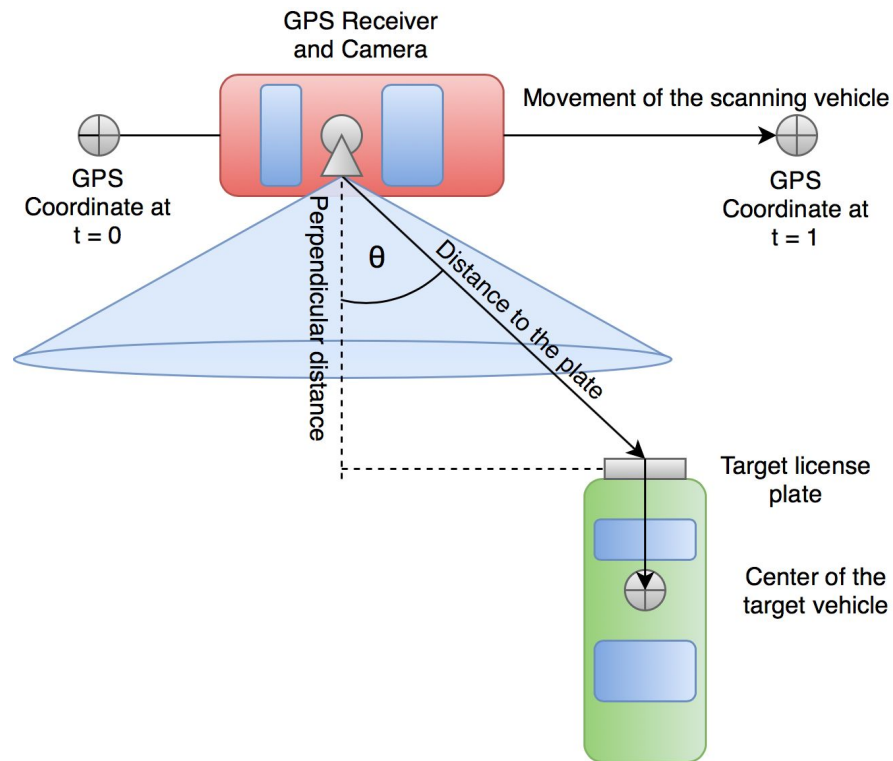


Figure 9: Car of Interest Position Measurement

2.10 MATCHING PARKING SPOTS TO THE ESTIMATED TARGET VEHICLE LOCATION

The parking lot is represented as an array of coordinates where each individual coordinate represents the center of a parking spot. The matching of a parking spot to the target vehicle is done by snapping to the closest parking spot center point. Distances to the individual parking spots are calculated and the minimum distance parking spot is chosen. Figure below depicts a situation where four parking spots are being processed to find the best candidate for the calculated target vehicle location.

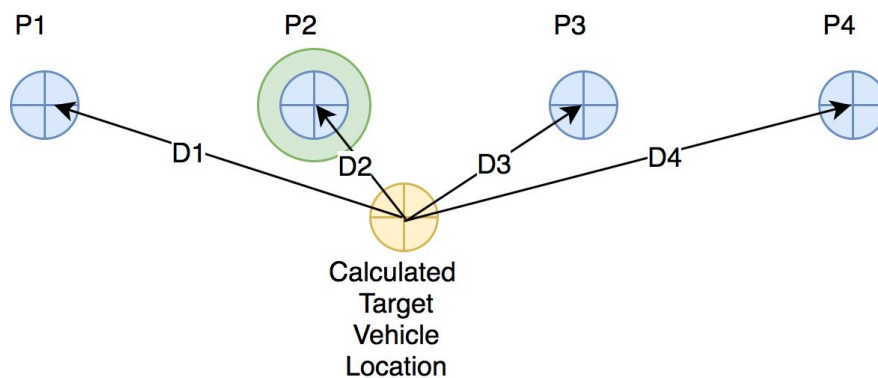


Figure 14: Spot matching

The confidence of the matched parking spot is calculated using the distance of the target vehicle to the parking spot. If the distance is 0.0, the confidence is the highest. If the vehicle deviates from the parking spot coordinate, the confidence lowers. When confidence crosses below 50%, the vehicle could be in the adjacent parking spot and we can no longer guarantee the exact parking spot.

2.10.1 PLATE PERPENDICULAR DISTANCE

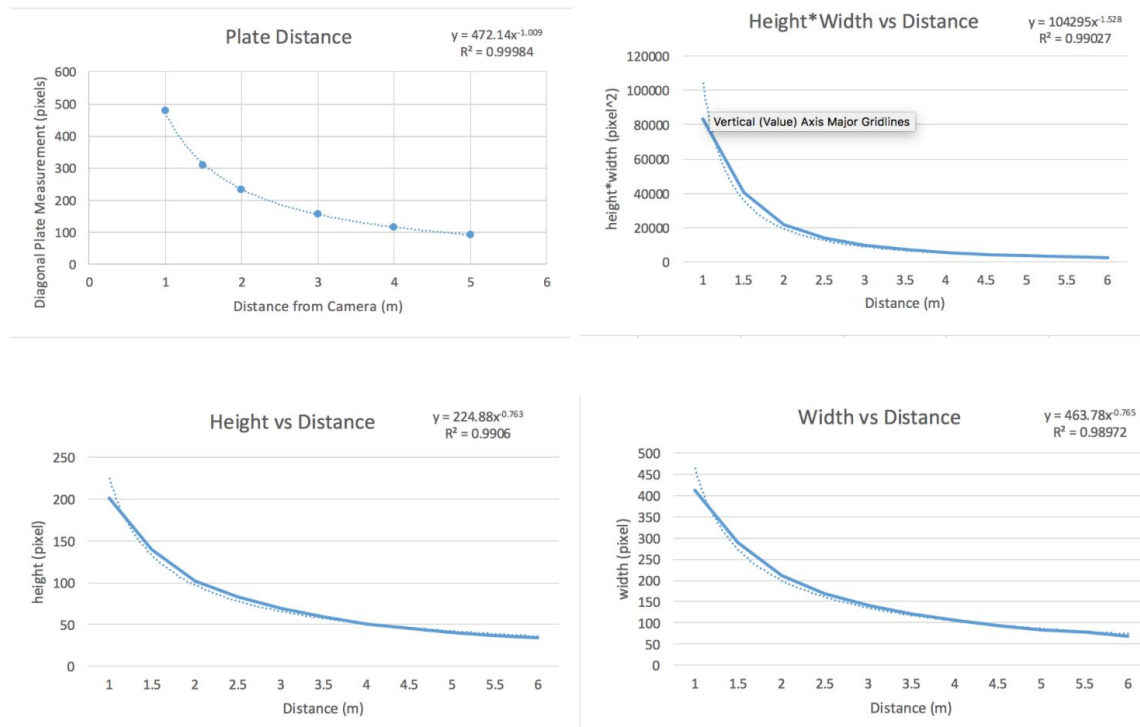


Figure 10: Raw Distance Data Collection and Interpolation Fit

A requirement for our project was to accurately find the position of each vehicle. To do this we used a object of reference to gain information on the scale of the intended vehicle. Our object of reference was licence plates because it has uniform size on every car. The standard United States plate is 12" wide by 6" height. Using the OpenALPR library we are able to extract information about the corner coordinates of each plate. From the pixel coordinates a simple calculation rendered the width and height in pixels of the plate.

$$height = ((bottomleft.y - topleft.y) + (bottomright.y - topright.y))/2$$

$$width = ((topright.x - topleft.x) + (bottomright.x - bottomleft.x))/2$$

Figure 15: Height and Width Calculation

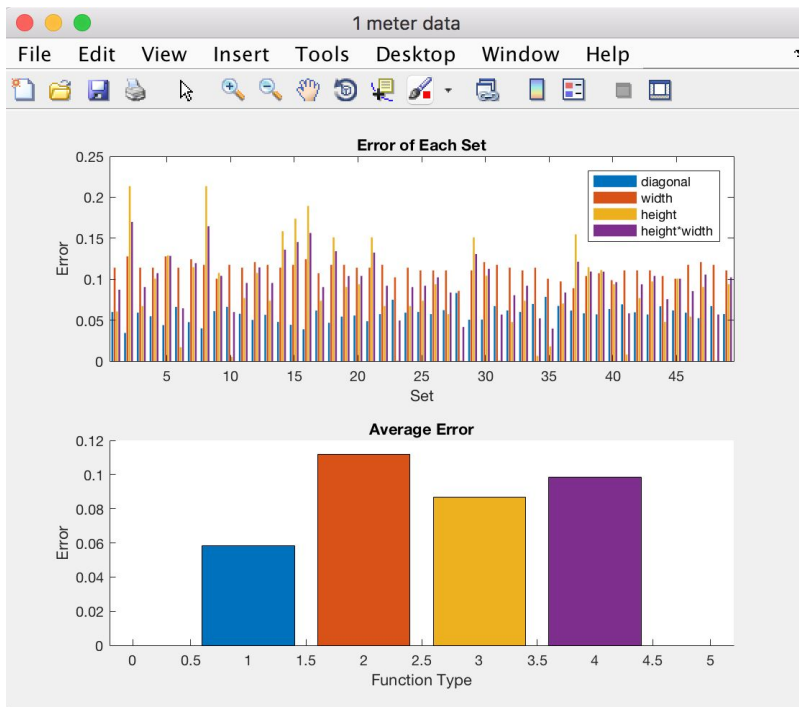


Figure 11: Distance Error Comparison: 1 Meter

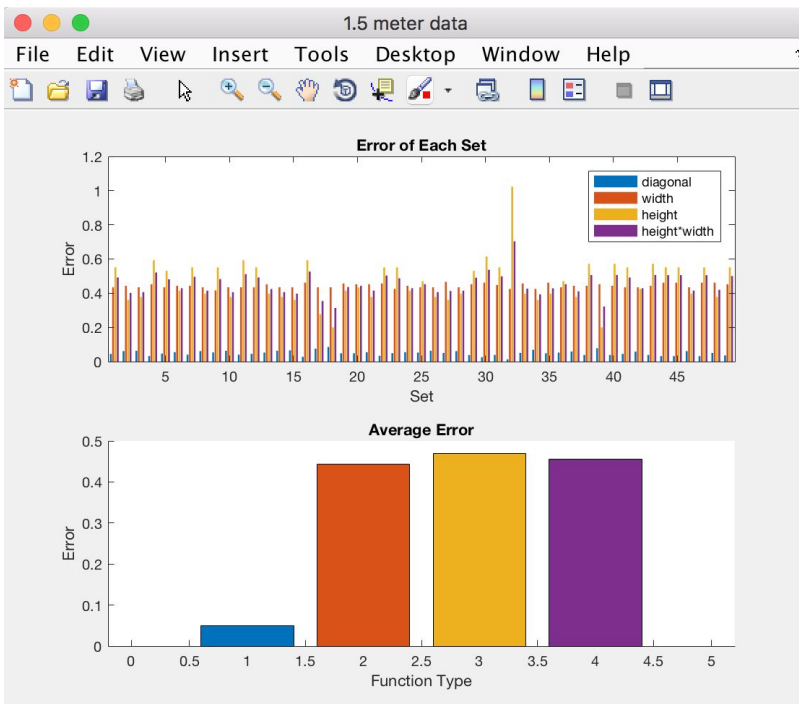


Figure 11: Distance Error Comparison: 1.5 Meter

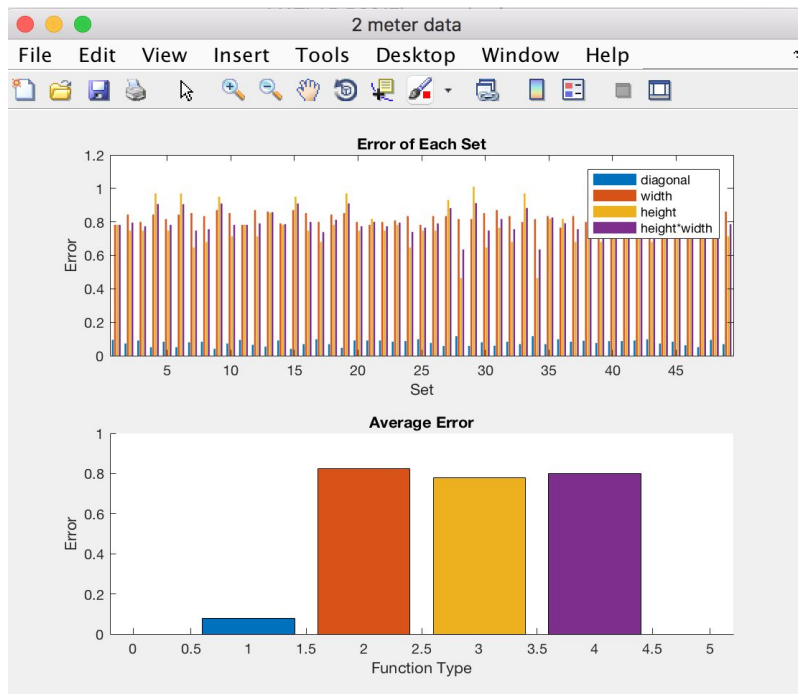


Figure 11: Distance Error Comparison: 2 Meter

We next focused on using the width and height of plate pixels to determine the distance the plate is to the camera. Our approach was to collect raw data of the licence plate at fixed distances from the camera. In figure 10 you can see Jake positioning the plate a distance of 1 meter from the camera. The plate was positioned in direct line of sight to create ideal conditions. This was repeated for distances of 1.5, 2, 3, 4, and 5 meters. After the data was collected it was processed by averaging the readings at each distance to produce an average width and height. Four plots were generated as follows and a best fit line was added figure 10 diagonal vs distance, height vs distance, width vs distance, height * width vs distance. This equation interpolates points in between our raw data points. To test how well the four functions worked, a MATLAB script was written to calculate error in figure 11 with the error being meters deviated. It was found that the diagonal method had the least error at multiple distances figure 11 so it became the function of our choosing. We were able to reach 10 cm accuracy for plates < 3 meters away and 7 cm accuracy for plates < 1 meter away.

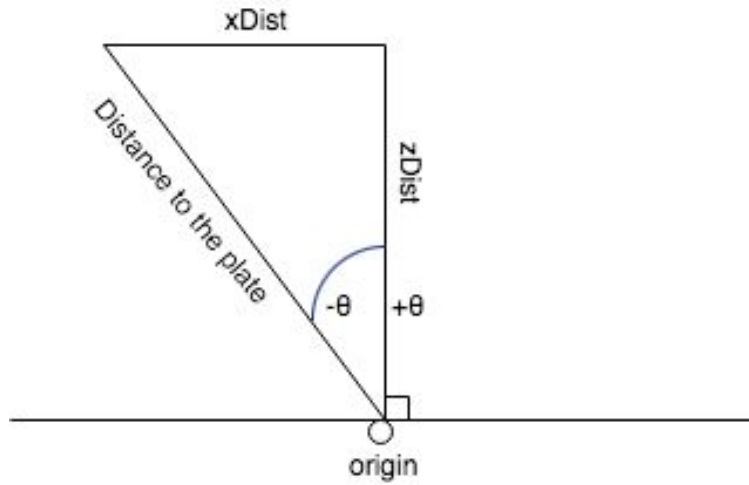


Figure 12 Distance Geometric Relation

Calculations to get to the final distance of plate are as follows:

$$\begin{aligned}
 plateWidth &= 12in = 0.3048m \\
 zDist &= perpendicular\ Distance \\
 xDist &= (plateCenter.x - frameWidth/2) * (0.3048/plateWidth) \\
 \Theta &= atan(xDist/zDist) \\
 distanceToPlate &= CarCourse + 90 + \Theta * (\pi/180)
 \end{aligned}$$

Figure 13: Distance Calculations

Some issues we had while implementing the distance functions was when the licence plate detected extra material around the perimeter of the plate. This was present on cars with licence plate frames or prominent lines surrounding the region. Because of this distortion was introduced in the width and height parameters resulting in error from the true distance. Another Issue the team ran into was if the plate is not normal to the camera you get unparallel plate widths in the measurements. This was resolved naturally as the car is always driving perpendicular to the parked cars in the lot.

2.11 TECHNOLOGY CONSIDERATIONS

The team has looked into is the use of a high resolution camera vs a low resolution camera. The high resolution camera provides a clearer picture which makes the licence plates easier to read. However, the high resolution picture will also take longer to process as it encompasses more data. The low resolution camera is capable of finding the licence

plates and cars in most cases. A compromise we looked into making was to continually process on low resolution until we find a car then process the licence plate ROI in high resolution. We found this to not work well with our code as resolution differences were not easily transferred between functions. Instead, we settled on a tight ROI using the high resolution camera. It proved to be effective in producing a clear image and reasonable computation time.

2.12 SAFETY CONSIDERATIONS

Operating any device while driving is a concern. For this reason, we decided to split the capturing and processing up into two parts. This will ensure the driver is focused only on the road. This will ensure they're not distracted by any software and will help prevent accidents. After driving through the entire parking lot, the driver will stop the recording and the processing of each frame will take place. This will ensure the driver only accesses the data while parked which will help prevent hazardous situations.

2.13 TASK APPROACH

The method for approaching this project is seen in the block diagram below:

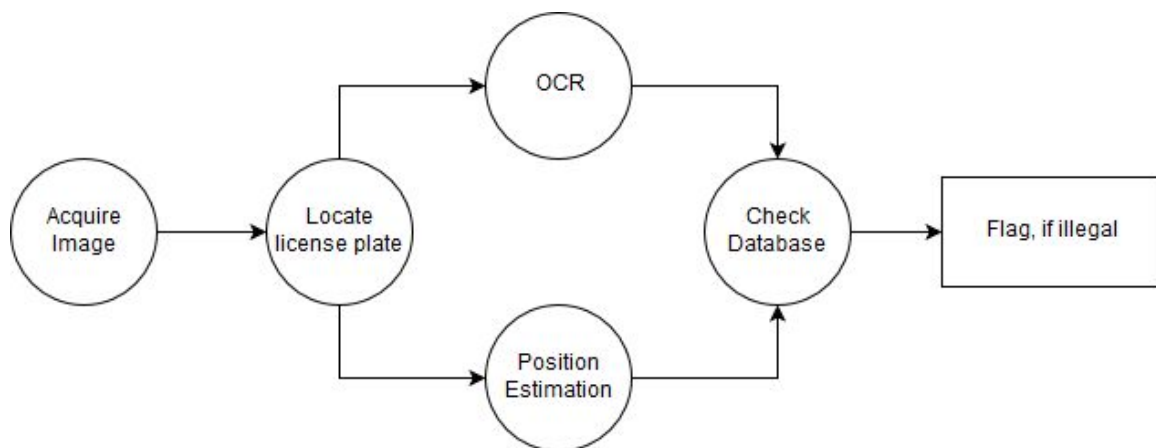


Figure 4: Project Pipeline

As shown in Figure 4, the image will be acquired by the a side mounted camera on the enforcement vehicle. The image will be scanned to find whether there is a license plate in the picture or not. Then the image gets processed in two ways. The first will be to get the position estimation based on the angle and size of the license plate. The second will be to get the textual representation of the license plate by using OCR. Once the position of the target vehicle has been determined and the license plate has been read, it can be compared to the database for the current parking lot. If the target vehicle doesn't have a permit linked to their license plate, a flag will be raised.

2.14 POSSIBLE RISKS AND RISK MANAGEMENT

Some things that may slow the progress of this project are getting the video feed settings correct so that the licence ROI is in focus. Integrating each part of the project together may have some unforeseen difficulties. Other challenges include the camera sensitivity, GPS device accuracy, projection estimation, and processing frame rate.

After hearing back from Mr. Miller and being informed that ISU cannot provide a list of licences registered to each lot because of legal risks, we reverted on our backup plan to manually create a fake dataset ourselves.

2.15 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

Our project can easily be split down into 3 distinct objectives.

1. Identifying if a car is in the frame and mask the ROI of the licence plate
2. Character recognition on the licence plate and check with comparison table
3. GPS location estimation of the vehicle of interest.

As a group of three we have decided to split this work amongst each individual.

Justin Wheeler will be handling the first objective. His evaluation criteria will be correctly identifying if a car is in the frame, and licence plate region up to 90% accuracy.

Tim Lindquist will be working on the second objective of character recognition evaluation on the licence plate. He will evaluate his criteria by seeing if the detected letters match up to the table scenario.

Jakub Hladik will be working on the third objective. He will test the position estimation with a known GPS location to an object and see if he can calculate that from a image to the nearest 1 meter.

2.16 PROJECT TRACKING PROCEDURES

To track the progress of this project we will follow the timeline and make sure each key milestone is met by the projected date. By doing so we will be able to complete the project before the due date and have time for problems that may arise.

2.17 EXPECTED RESULTS AND VALIDATION

The expected parking violation detection accuracy is at least 85%. The detection accuracy will be validated by a real-life test scenario in one of the Iowa State University's parking lots. Due to legal issues, license plate parking permissions data will be randomly generated by hand after analyzing a video footage of the parking lot drive through.

2.18 TEST PLAN

Reading license plates will be the most crucial part for this project to succeed. Because of this we will be adding a confidence measure as an indicator of how likely the computation

is to being correct. Getting accurate location results is also very important which can be tested by creating a pinpoint of the received latitude and longitude as a reference to compare with.

Our tests consisted of taking multiple videos while traveling through the Howe parking lot around 5 mph. We weren't able to get access to the ISU parking division database so we had to manually generate a list of the license plates in the parking lot. While processing the captured video, we were able to read the license plate with OpenALPR and cross-compare with our manually generated list. If the license plate didn't exist, we would mark it as a violation. Throughout three test runs, we were able to get a detection rate accuracy average of 67.29% without character substitution and 77.58% with character substitution.

Testing the parked car parking spot was done using mini-map which is stored in the center of the report video. When a car was scanned, we would use distance calculations to determine the angle at which the plate was being read to determine the exact location. Based on the exact location, we would cross compare it with the coordinates of all the parking spots that we manually mapped out in the parking lot. It would then snap to the closest parking spot and color it either red or green. Our final report showed a position accuracy of 98.26% while testing a portion of the Howe parking lot.

The final system test involves running a footage of a car driving through a parking lot through our system and comparing results with analysis by hand. This will determine our violation detection accuracy.

The ability to run our system in real-time was something we originally planned to do. We found that we had to drive incredibly slow, < 1 mph, for the decreased frame rate to capture reasonable data. Because of this we decided to split the system into a capture phase and processing phase. The capturing phase allows the driver to focus on driving instead of looking at the results. Once done cycling the lot, the driver can hit process the video and return to violated positions. This creates a much safer operating environment by eliminating a distraction and resolves our real time computational issue.

3 Project Timeline, Estimated Resources, and Challenges

3.1 PROJECT TIMELINE

		February			March			April		
Discription	Date									
Initial Proposal	2/1									
Capture Test Footage	2/10									
Oral Proposal	2/20									
Written Proposal	3/9									
Car Detection	3/15-4/1									
Licence Plate Char Recognition	3/20-4/3									
Projection GPS Location	3/20-4/3									
Incorporate Parts	4/1-4/10									
Debug/ Final touches	4/1-4/15									
Project Writeup	4/19									

Figure 5: Project Timeline

3.2 FEASIBILITY ASSESSMENT

The realistic outcome for this project will be a camera system which will capture a constant video stream on the right side of the car. The stream will then be sent to a laptop using OpenCV which will process those frames to extract license plates of cars. The location of the target vehicle will be estimated. The results will then be sent to our database to check if there is a permit associated with the plate number in the specific parking lot.

3.4 OTHER RESOURCE REQUIREMENTS

The following are also required to complete the project.

- Comparison table of permissible licence plates.
- Stock enforcement vehicle POV footage for preliminary testing.
- Portable Computer

3.5 FINANCIAL REQUIREMENTS

Hardware	Cost
GPS Unit	\$30
Camera	\$60

Table 2: Hardware costs

3.6 RESULTS

Throughout the plates read, we noticed a strong pattern where certain characters were very hard to distinguish from each other. For example, the number zero and the letter ‘o’ look very similar. This is the same for other characters such as the number zero and the number 8. With character substitution we were able to check the database for license plates similar to the plate read. An example of this would be a license plate number being “CXB8o9” but the OpenALPR would detect the plate number as “CX88o9”. The number “8” and “B” look similar to the human eye and for the visual detection system. We believed it this was an acceptable solution to check each case as there is a slim probability for a similar licence in the parking lot. If there was a mistake and it matched a car in the lot that didn't have a permit a false negative would occur with a consequence of the car avoiding a ticket. The resulting improvement of detection accuracy outweighed this risk by ensuring the car isn't in-properly marked as a violator.

With our demonstration footage, we got 3 incorrect readings with character substitution and 6 incorrect readings without character substitution. This was done on a total of 29 cars read. This small test showed a 10% increase in license plate detection rates. Overall, on the demonstration footage the license plate detection rate successfully detected 89.6% of the license plates read.

Trial Data:

Without Character Substitution

Trial	Total # of cars	Correct Detections	Incorrect Detections	Missed Detections	Licence Detection Rate	Average Plate Confidence
1	30	23	6	1	76.67%	80.90%
2	41	24	14	3	58.54%	77.60%
3	36	24	11	1	66.67%	80.99%
Avg					67.29%	79.83%

With Character Substitution

Trial	Total # of cars	Correct Detections	Incorrect Detections	Missed Detections	Licence Detection Rate	Average Plate Confidence
1	30	26	3	1	86.67%	80.90%
2	41	28	10	3	68.29%	77.60%
3	36	28	7	1	77.78%	80.99%
Avg					77.58%	79.83%

Parking Approximation Data

Trial	Total # of cars	Incorrect Spot Predict	Spot Detection Rate	Average Spot Confidence
1	30	0	100.00%	63.04%
2	41	1	97.56%	78.60%
3	36	1	97.22%	65.35%
Avg			98.26%	69.00%

In the tables above Average Plate Confidence is the average of the final confidence for each plate. Average Spot Confidence is the average of the confidence for each spot where a vehicle was detected.

3.8 LESSONS LEARNED AND IMPROVEMENTS

In this project, we learned about license plate recognition and processing inputs from devices such as GPS and a web camera. We also learned lessons from the algorithms we decided to use and their weaknesses.

3.9 WEB CAMERA AND OBTAINING IMAGES

The web camera caused us some difficulties. Autofocus was constantly changing as the different cars have different lengths and are parked at different distances. Also, automatic exposure settings takes some time to optimally adjust, causing some frames to be misread. The poor quality of the lens optics is noticeable when compared to the iPhone camera of the same resolution. An improvement would be to use a more expensive and more capable camera.

3.10 GPS POSITION ESTIMATION

It would be better to use a Kalman filter instead of simply interpolating between two points. It also might be beneficial to use additional position and orientation sensors to yield better results from the Kalman filter. Another approach would be to use differential GPS.

3.11 LICENSE PLATE RECOGNITION

OpenALPR factory neural networks are not trained enough for all characters. Some license plates have frames which cripple the recognition. For example, some registration stickers slightly cover up characters causing the neural network to misinterpret the character or the issue of is misinterpreting a zero with a slash in the middle for a number 8.

3.12 ENVIRONMENTAL CONDITIONS

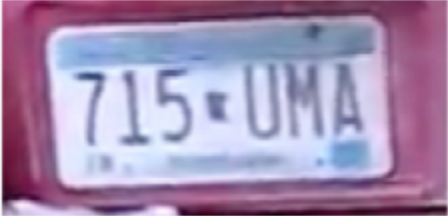
Some license plates are too dirty to get any sensible reading. It is illegal to obstruct license plates and it would be up to the parking lot operator how to deal with such cases. The OpenALPR does not read dirty license plates and it simply report no vehicle.

3.13 GUI

The current user interface simply overlays the video obtained from the camera. It would be better to provide a user with a simple non-destructive user interface as proposed below.

Session #332324 3/6/2018 4:42 PM Lot 03 (Howe)

ROI:



Reads:

715UMA

COLOR: RED

STATUS: VIOLATION

CHECKED: 23

VIOLATIONS: 2

Generate Report

Figure 2: Concept In-Progress Operation

Figure 2 as shown above is what the driver will see while navigating through the parking lot. The picture of the license plate will show the last license plate read. It will also show the status, color of the car, and the textual representation for the last license plate read. On this screen, it will also show the total number of license plates read and how many violations have been identified. Once the driver has finished driving through the parking lot, they will be able to generate a report for the entire parking lot.

The report generated will look similar to Figure 3. For each session, a unique identifier will be generated and used for history purposes. It will show which parking lot was scanned for the report by using the enforcement vehicles GPS location when launching a new session. It will display in a table format the license plate number, color of the vehicle, violation status, and action taken for each license plate scanned. The enforcer then has the option to print the report or start a new session.

Session #332324 3/6/2018 4:42 PM Lot 03 (Howe)
 Total: 24, Violations: 4 Status: In Progress

Registration	Color	Status	Action
715UMA	Red	Violation	CHECK <input type="checkbox"/>
442EEH	White	OK	
521TWL	Pink	OK	
GOCYCLS	Black	Violation	ISSUED <input type="checkbox"/>
DAW916	Gray	Violation	ISSUED <input type="checkbox"/>

New Session

Print

Figure 3: Concept Operation Report

4 Closure Materials

4.1 CONCLUSION

The purpose of this project was to create a more efficient method for Iowa State University's parking violation system. We accomplished this through the use of computer vision in the form of licence plate recognition. By doing so our group was able to extract information on the plates and car to gain insight about the vehicle. We then compared that information with a database to see if a car was allowed in a specific lot.

To verify our system, we put it through a number of tests with specific specifications to be met. These test included correctly identifying the licence plate characters, determining a accurate representation of the cars location. These are discussed in detail above in the test section. After that the group implemented a friendly user interface to provide simple operation feedback.

4.2 REFERENCES

- [1] Hill, V., & UW La Crosse. (n.d.). Enhancing Parking Services with AMS. Retrieved March 06, 2018, from <https://www.genetec.com/solutions/resources/uwl-parking-enforcement>

- [2] Williams Edward, Aviation Formulary V1.46. Accessed on April 17, 2018, Retrieved from <http://www.edwilliams.org/avform.htm>
- [3] Howe Parking Lot Map. Provided by Mr. Miller, ISU Parking Director.
- [4] Aboura, K., & Al-hmouz, R. (2017, August 3). An Overview of Image Analysis Algorithms for License Plate Recognition. Retrieved March 8, 2018, from <https://www.degruyter.com/downloadpdf/j/orga.2017.50.issue-3/orga-2017-0014/orga-2017-0014.pdf>
- [5] Palmer, C. (2012, December 26). OpenCV: How-to calculate distance between camera and object using image? Retrieved March 09, 2018, from <https://stackoverflow.com/questions/14038002/opencv-how-to-calculate-distance-between-camera-and-object-using-image>
- [6] OpenALPR. OpenALPR - Automatic License Plate Recognition Library. Retrieved March 09, 2018, from <http://www.openalpr.com/>
- [7] OpenCV. OpenCV (Open Source Computer Vision Library). Retrieved March 09, 2018, from <http://www.opencv.org>

4.3 APPENDICES

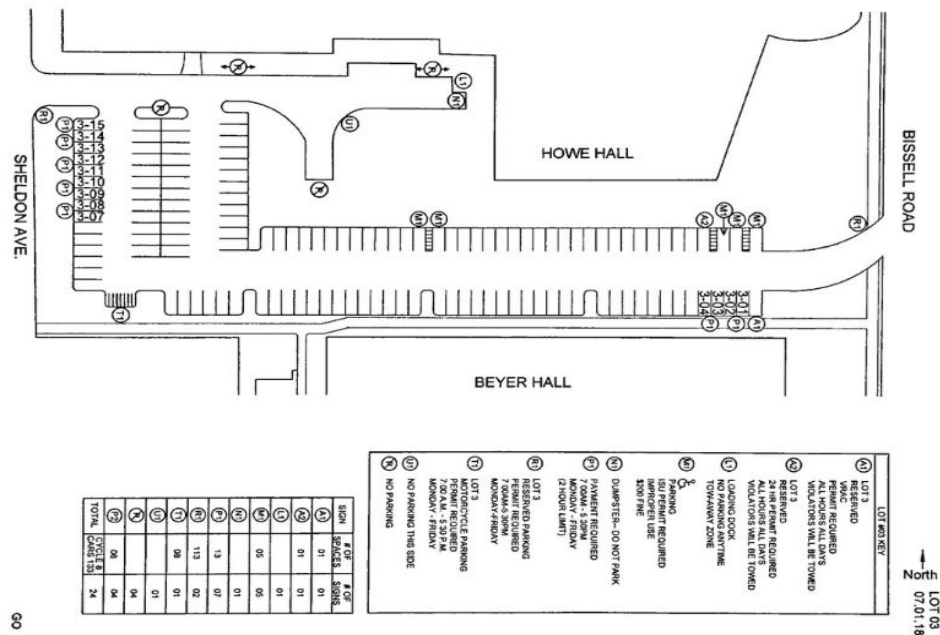


Figure 6: Map of Howe Parking Lot