Quantifying the Impact of Video Quality and Hardware/Software Stabilization on Facial Detection and Recognition in a Mobile Robot System

James Aloysius Murphy IV

****

Computer and Economics Science Department

Advisors: M. Fuat Sener and Nicholas Webb

UNION COLLEGE

Abstract

This paper empirically explores the relationship between the performance of real time facial detection/recognition and software/hardware camera stabilization techniques in mobile robots. Since the 1960's, computer vision (CV) has grown in a number of ways including but not limited to, transportation, security systems and mobile service robots. As the demand for CV has increased, so too have the capabilities of these complex systems to include advanced functions like facial detection and recognition. However, without a stable image for these systems to process, new developments in CV will not be as applicable for mobile robots.

While facial recognition systems that rely on stationary cameras are already in use, there is a drop in their performance in CV applications when implemented in moving systems (Jung et al. 2004). One issue that we have experienced during experiments at Union College (N.Y.) is the inability to maintain a "smooth" video feed required for detecting and identifying individuals. In CSC-325: Introduction to Robotics, my group's robot relied on facial recognition for movement and would lose track of its target each time it began to move due to camera shake. In response to this problem, this study explored the benefit of camera stabilization hardware and software solutions. For the test four solutions were used:

1. No solution
2. Camera stabilizing *hardware*
3. Camera stabilizing *software*
4. *Both* camera stabilizing *hardware and software*

The hardware, a device commonly used in independent film production, reduces unwanted camera shaking by using counter weights to ease the motion of a camera. The software, Deshaker, is an open source system that removes the same camera shake but through frame by frame processing. These four solutions were tested on Union College's Turtlebot (Figure 1).

Figure 1*: Turtlebot*

In order to see if these results were consistent, each of the four solutions were tested 30 times. 15 of the trials were recorded using a Samsung S7 (13 megapixels) and 15 were recorded using a Logitech USB camera (3 megapixels). Two cameras were used to provide more data regarding the four solutions effectiveness and provide insight into the benefit of image qualities effectiveness in CV applications. After quantifying the results, the hardware stabilization solutions showed a reduction in facial detection rate while all three stabilized solutions appeared to cause an increase in accurate recognition of the target. Camera quality was the most impactful for detection and recognition with the 13 megapixel camera far outperforming the 3 megapixel camera.

Table of Contents

I. Introduction

A. Inspiration for the project and the founding of facial recognition

Computer Vision (CV), like many other applications of artificial intelligence, has rapidly expanded in the past decade. CV use has expanded across many industries including but not limited to the development of autonomous vehicles, industrial robots, medical image analysis and mobile robots able to map their surrounding environment. The field evolved from a summer project Marvin Minsky, a MIT professor regarded by some as the father of artificial intelligence, assigned to an undergraduate student in 1966. Minsky wanted the student to simply "spend the summer linking a camera to a computer and getting the computer to describe what it saw" (Boden 2006, pg 781). He believed that this task could be accomplished in just one summer It was quickly discovered that this would not be an easy two month undergraduate project and computer vision became a separate branch of research for artificial intelligence.

In that same year, Woodrow Bledsoe developed a system that helped identify faces using a RAND tablet, one of the first low cost digital graphic computers. This system allowed subjects to manually mark particular facial features like the eyes, nose, hairline and mouth with a stylus. He then used these points to extract geometrical features of the face to compare them to a known database of faces with these measured features. Bledsoe then selected the face that most resembled the geometrical features of the individual subject. While Bledsoe was limited severely by the processing power of computers, it was an important step in understanding the difficulties of the field. Even he described the process of facial recognition as “difficult… [due to] the great variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc” (Bledsoe 1966). While computing power has increased exponentially since 1966, Bledsoe’s observed difficulties and techniques still apply today.

CV has been rapidly growing due to “cheaper and more capable cameras… affordable processing power and because vision algorithms are starting to mature” (Bradski, 2008). These open source software libraries like OpenCV allow a jump-start to research projects. There free libraries allow for any user to explore questions regarding computer vision applications without developing an entirely new system all together. For example, in the fall of 2017 I was able to successfully use one of OpenCV’s facial recognition softwares to create a robot which used facial recognition. It then used the location of the identified target to move towards them.

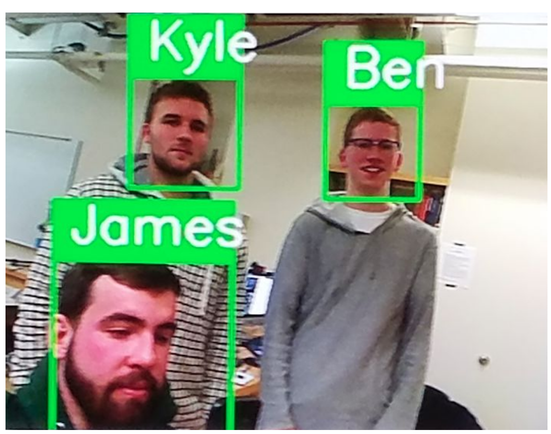
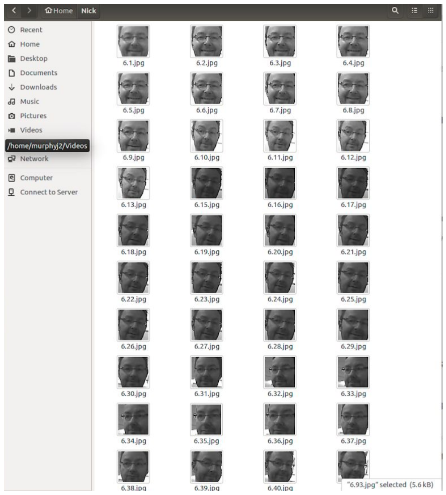


Figure 2*: Functioning facial recognition*

During our project, Kyle and Ben, my group members, and I discovered a major issue regarding our tracking robot. Because our system was designed to not move until it identified the target we chose, when it did identify the target it began to move in a jolting manner. This spastic movement caused our Logitech Quickcam Pro 9000 webcam to shake, thus disrupting the smooth video feed required for a high functioning computer vision application. Through trial and error we were able to discover other characteristics that caused this behavior such as bumps and variance in speed. This inconsistent video feed raided the following issue: How can you reduce this shake? What is the most cost efficient way to do this? Are solutions to this shaking worth it? Before I get to these questions, it is essential to explain the fundamental concept of facial detection and recognition and how they work.

B. Explanation of Facial Detection and Recognition

The most integral part to understanding this study is distinguish the difference between detection and recognition. *Detection* is the process that allows the system to confidently say that it has found a face. It does not know who that person is, it just knows that it is a face based off of techniques I will explain in the next paragraph. *recognition* is the process of extracting the features from the detection (See Fig 4.) and comparing them to a database of classified images (See Figure 3).



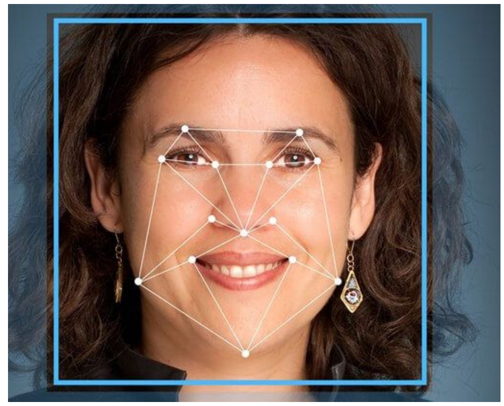


Figure 3*: Database of classified images*

Figure 4*: Feature Extraction*

While Bledsoe was able to manually compare these features, Paul Viola and Michael Jones had a breakthrough in their work. This breakthrough allowed them to automate facial detection to real-time. They accomplished this by implementing Haar features, integral image processing, adaboost machine learning and cascading. In short, Haar features funnel an image by constantly eliminating regions that it has identified not to be a face. It does this by scoring the average intensity of each pixel in relation to its four neighboring pixels into an integral representation image. Since the image is now extracted into numbers, it can process the image faster. Fast-enough for real-time video. It then takes this scoring to extract features that resemble lines, edges, and four-rectangle features. Because faces have different shaded regions like the cheeks, nose and forehead which are brighter, the system is able to look for these difference and determine if a face is contained within the image. The system then extracts facial features (see Figure 4) that Bledsoe found by hand and compare them to a known database in real-time (Viola and Jones, 2004).

Viola and Jones' discovery is the leading facial detection method, especially on the OpenCV platform. But what use is such a system if it can't extract features due to a shaky image? As mobile robots grow in potential for security, military and civilian service, they will require a reliable video feed that will allow the robot to interact with their environment while moving and performing other sophisticated tasks.

Alongside these fast developments in CV, software has been developed to combat unwanted camera shake caused by motion of the camera or videographer. The software is able to shift the electronic image by performing frame by frame analysis of the camera motion to create a smooth image (Matsushita et al. 2006). This has become popular in the film industry along with other hardware solutions which provide a similar outcome. These camera stabilizers accomplish the same task by using a mechanism which mounts the camera on to the apparatus and counters camera shake through motors or counterweights. The focus of this paper will be quantifying the benefits of these camera stabilizing techniques on facial detection and recognition in a mobile robot.

In Chapter 2, I plan to provide a brief overview of the literature on computer vision performance evaluation in mobile robots as well as engineering economics evaluation methods. I will compare the paper's methods and relate them to my future analysis. I will then evaluate these methods and their feasibility in a multitude of computer vision testing environments. In Chapter 3, I will specify my exact methods for data collection and variance in control variables. In Chapter 4, I will discuss the results of the experiments, including any unexpected challenges or observations that I discovered through experimentation. Chapter 5 I will conclude the study by explaining the implications of my data for future computer vision projects in mobile systems.

C. Software and Hardware

This project used both hardware and software stabilization in an attempt to reduce the shake of the camera during the recording of the experiments. For the hardware solution, a Roxant Pro video stabilizer was used. This system uses counterweights to reduce the instability a camera during movement. For the software solution, this study used Deshaker, an open source software solution, which analyses a video frame by frame to cut out inconsistent video within the live feed. This solution's stabilization is most prominent at the edge of the screen.

D. Estimating Impact of Online Product Reviews

Along with this facial detection/recognition analysis, I will also explore the relationship between camera qualities and online customer reviews to gain insight into customer expectations and product satisfaction. With the rapid growth of the internet, many stores and applications have been developed to allow consumers the convenience of ordering products from anywhere through just a few clicks. These platforms have also provided a forum for customers to leave reviews and communicate their satisfaction with a product through ratings and reviews. This online product presence creates ample data to mine and understand what impact these reviews have on overall product perceptions. Using this data can help understand consumer preferences.

II. Literature Review

A. Facial Detection and Recognition

The problem of camera stabilization in the CV field has gained a lot of attention alongside the advancement of robotics due to its necessity for map building and object detection. There are many proposed solutions to image stabilization. One solution is object tracking, demonstrated in the work of Censi et. al (1999) and Zoghlami et al (1997). In these studies both works focused on a process to estimate the transformation between two image coordinate systems. Censi et al. used feature tracking to create a mosaic gray scale images from a collection of frames. In this study they quantified their effectiveness by calculating the root mean square (RMS) error between their reference frame and gray scale image (Censi et. al,1999). Zoghlami et al. choose to focus only on geometrical edges. These edges help with the processing efficiency because it allows for less overall pixels to be tracked while still carrying the significance about the motion of an object. The evaluation of their success was measured in percentage of pixels that matched and how far the incorrect pixels were located (Zoghlami et. al,1999). However, the motions of moving objects were not considered which limited the estimation ability of their algorithms.

Other approaches focused on motion tracking but only used pan/tilt cameras like Murray and Basu (1994) and Foresti and Micheloni (2003). However, the most common motion in a mobile robot is forwards and backwards. Nordlund and Uhlin in 1995 were able to create an integrated system that allowed them to focus on a moving object using a camera on a mobile robot. Their algorithm allowed for real-time implementation of object tracking. Jung and Sukhatme in 2004 were able to detect moving objects in three different robots, a helicopter, a segway robot and a small Pioneer2 AT bot, in an outdoor environment using a single camera. Their "process is performed in two steps: the ego-motion compensation of camera images, and the position estimation of moving objects in the image space" (Jung and Sukhatme, 2004). They manually tracked objects and compared them to results of the image processing. They then calculated the success rate of detection and recorded any true and false positive recordings. Finally, they recorded the "Avg. Error.. [which] is the average Euclidean distance in pixels between the ground truth and the output of tracking algorithm" (Jung and Sukhatme, 2004).

This work has inspired me to use the previous research methods evaluation process for my own experiment. While Jung and Sukhatme were focusing on a similar motion that I will work with, they are evaluating an algorithm rather than solutions being implemented by a system. In order to accomplish this I will record frame by frame facial detection and recognition of the OpenCV system.

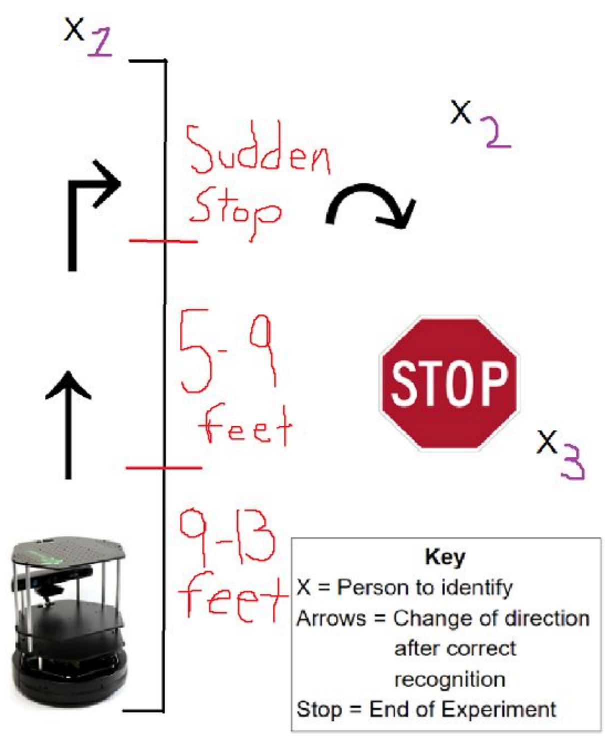
B. Customer Review Analysis

Ghose et al. (2011) exemplifies the insight that customer reviews can provide by implementing machine learning and training a system through supervised exercises. In doing so, Ghose created a system that could rate the helpfulness of each individual review. From this analysis Ghose was able to explore the importance of these reviews and their impact on the product sales. In an effort to explore these reviews, I will use Amazon reviews and explore the impact of the frequency of words within each review and camera qualities have on the consumer rating.

III. Methodology

A. Experiment Design

There are many factors that can affect the ability of a facial recognition system which Bledsoe and many others have noted in their research. These factors include environmental factors, tilt of the face, lighting, facial expression, etc. Because the focus of this study was the benefit of image stabilization solutions and picture quality, I designed an experiment that aimed at eliminating as many of these external factors as possible. Thus, a uniform course was created for the turtlebot to navigate and record its video on a camera mounted to the robot (see Figure 5). The goal of this path was to create a consistent video feed to compare different solutions detection/recognition results .



Start

***Figure 5**: Sketch of Experiment Course*

In order to navigate this course, the robot used Simultaneous Localization and Mapping (SLAM) to determine its location. In order to use this autonomous navigation, a map was created of the testing area, then four points on this map were stored in the navigation program. These four points allowed the robot to drive towards X1, rotate past X2, and drive straight past X3. SLAM navigation is a common solution to mobile robot navigation in practice which is why the method was chosen.

The outline of the experiment is as follows:

1. Robot starts at a predetermined distance away from X1, depicted by the robot in Figure 5.
2. It begins to move directly towards X1.
3. After successfully traveling to the first marked point, the turtlebot rotated in place with X2 in its field of view.
4. It then drove past X3 until the final subject was out of its field of view.

After the turtlebot was able to consistently travel the course,the OpenCV facial detection/recognition program was trained on three subjects. In an attempt to eliminate differences in facial expressions, printed images were used for subjects. The subjects were: X1-**Donald Trump**, X2-**Hillary Clinton**, X3-**Barack Obama**. The detection/recognition system was trained to identify these three subjects on four separate images per subject. Because the faces were printed, a fifth untrained image was used for the experiment because the system was unable to train itself on new faces, only detect them and compare them to known ones it had previously been trained to identify.

Next, a solution was needed to mount the hardware on top of the PVC pipe. In collaboration with Union College's 3-D printing lab, a mold was create to slide firmly on the PVC pipe and secure the hardware through a bolt (see Figure 6).

Figure 6*: Hardware Solution Attached to a 3-D Printed Mount*

With the navigation and mount in operation, the experiment was now set up for trials. Each trial was tested under one of the following four conditions:

1.No solution
2.Hardware solution
3.Software solution
4.Hardware and Software solutions combined

For each four conditions 15 trials were recorded and results averaged. In total, this provided 60 recordings. However, in order to test for camera quality impact on detection/recognition, these 60 recordings were captured for both a Samsung Galaxy S7 (13 megapixel camera) and a Logitech Quickcam Pro (3 megapixel camera) Thus, when the trials were all recorded there were a total of 120 individual trials.

B. Video/Data Processing

After all 120 trials were recorded their videos were separated into three smaller videos (creating 360 videos instead of 120) based off of the subject in its field of view i.e. Donald Trump, Hillary Clinton, or Barack Obama. This was done to cut out time where a subject was not on the screen and to allow for the automation of analysis; because only one known subject was now in each video the target could be compared to the detected face. These 360 videos were then fed through the facial detection/recognition program. This program was modified to record and output (see Figure 7) frame by frame (frameNum) who it thought was the face detected (Id) and at what confidence it had for the prediction (conf). This confidence ranged from 1-232 and was later normalized to 1-100 for simplicity. Every processed video was then written to a comma-separated values (csv) file in the form of Figure 7.

frameNum	Id	conf
1	NA	NA
2	NA	NA
3	NA	NA
4	Obama	144.59819
5	Obama	160.00052

***Figure** 7: Example of The First Five Frames of an Individual Trial*

After all 360 csv files were saved, they were read into R Studio, a popular data analysis programming platform. and evaluated using R's dplyr library. New columns were created to hold boolean values for each frame detection and recognition ability. If the Id was not NA, then it received a 1 for that frame, 0 otherwise. If the Id was equivalent to the known target subject, it received a 1 for recognition, 0 otherwise.

Due to the noticeable difference in performance of the detection/recognition from ranging distances in the first leg with Trump as a subject, these trials were divided up into three equal groups based on their frame numbers. The first third of the frames was approximately 9-13 feet away, the second was 5-9 feet, and the last captured the remaining video along with an extra 1.5 seconds of video after the turtlebot had reached its first destination point. For later reference, these segments will be referred to only by their distances. The in place rotational segments that had to identify Hillary Clinton will be referred to as rotational and Barack Obama as face on the edge of screen.

After all the videos were separated and processed according to the method above, detection and recognition rates were calculated by adding the boolean values for the respective boolean columns and dividing them by the number of frames accordingly. This produced a percentage for both detection and recognition for all frames in each video segment.

C. Online Product Reviews

In order to extract Amazon reviews I used WebHarvy, a free web scraping software. This software allowed me to extract contents and ratings for 100 reviews for ten individual cameras. Along with this review extraction, the cameras qualities i.e. resolution, size, optical zoom etc., product online ranking, and average customer ratings were scraped as well. After I extracted all 1,000 reviews and camera specifications, I combined all of the csv files into R. Within R a tally was created for occurance of particular words for each individual comment including: quality, great, zoom, size, cost, price, resolution, broken, and returned. This data was then combined to include the product details that each comment was directed towards.

IV. Results

In this section all solutions use the no solution condition as a baseline comparison.

A. 9-13 Feet

As previously mentioned in the methodology chapter, the Trump subject section of each trial was broken into three distances. This section discuss the first section which is approximately 9-13 feet away from the image of Trump (see Figure 5 for clarification). The Samsung S7 did not show any signs of statistical difference in detection rate for any of the solutions (See Figure 8). Meanwhile, the hardware and hardware+software solution showed a -22.6% and -22.4% drop in detection rate which was significant at the .1% level. This significant drop in detection for the USB camera is alarming because the Samsung S7 showed no negative effect. This large variance between cameras that used the same solution can be explained by complications with the hardware. Because the USB cam is wired, it may have distorted the balance of the hardware. Despite providing extra slack in the cameras wire before the trial, the balance of the hardware could have still been tampered due to its sensitivity to weight change. The lack of negative effect in Samsung S7 could be because the phone recorded the trials wirelessly. Thus the hardware did not experience this imbalance.

The recognition rate showed positive results during the Samsung S7 trials for the hardware + software and software solutions. Interestingly again, the hardware caused a statistically significant drop in recognition of -2.3%.

	Dependent variable:			
	detection		recognition	
	Samsung S7	USB	Samsung S7	USB
	(1)	(2)	(3)	(4)
hardware+software	-0.016	-0.214***	0.054***	0.033*
	(0.013)	(0.024)	(0.011)	(0.018)
hardware	0.020	-0.226***	-0.023**	0.018
	(0.013)	(0.024)	(0.011)	(0.018)
software	0.011	-0.014	0.030***	0.016
	(0.013)	(0.025)	(0.011)	(0.019)
constant	0.846***	0.379***	0.092***	0.092***
	(0.007)	(0.018)	(0.006)	(0.013)
Observations	6,042	2,398	6,042	2,398
R2	0.001	0.059	0.008	0.001
Adjusted R2	0.001	0.058	0.007	0.0002
Residual Std. Error	0.358 (df = 6038)	0.425 (df = 2394)	0.303 (df = 6038)	0.311 (df = 2394)
F Statistic	2.249* (df = 3; 6038)	50.267*** (df = 3; 2394)	15.519*** (df = 3; 6038)	1.122 (df = 3; 2394)
Note:				*p<0.1; **p<0.05; ***p<0.01

Figure 8*: 9-13 Feet Regression Results*

Another explanation for the drastic difference for the USB camera that utilized the hardware could be the inability of the camera to identify the image of Trump from a distance. If we compare the detection rate between the cameras we notice a drastic difference (see Figure 9). In this graph the identical solutions for each camera are compared side by side. From a 9-13 feet distance it is clear that the megapixel difference caused a drastic drop in detection rate. The USB camera was roughly 33% of the Samsung S7 detection rate.

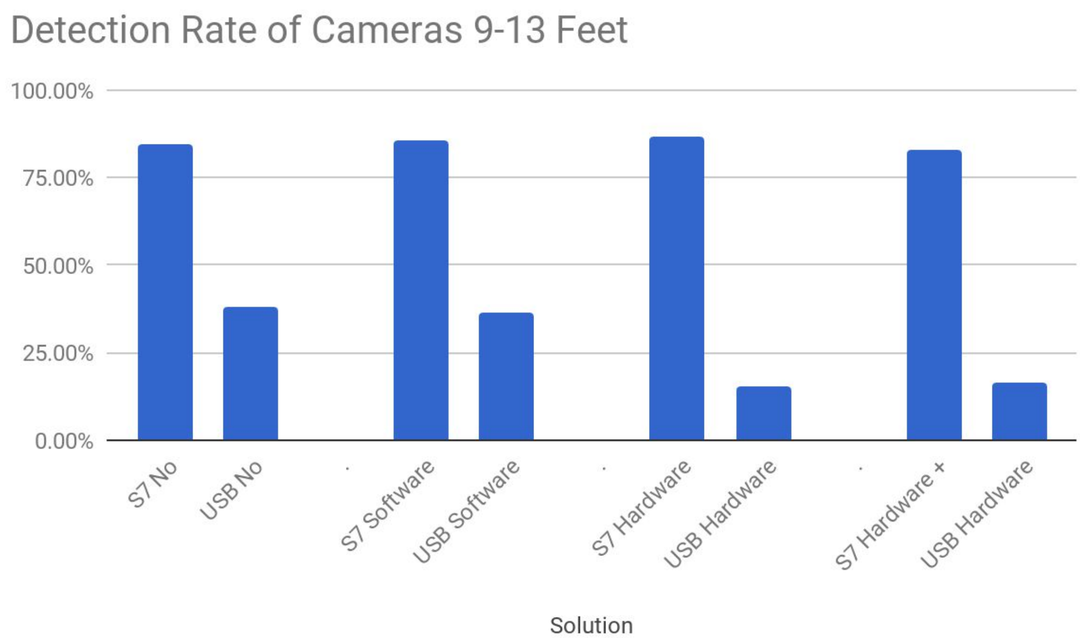


Figure 9*: 9-13 Feet Detection Rate by Phone*

B. 5-9 Feet

At the next distance of 5-9 feet a similar trend in section I detection rate was found. The hardware's influence on the USB camera was not as strong however. The hardware and hardware+software caused a -13.9% and -15.1% statistically significant drop.

	Dependent variable:			
	detection		recognition	
	Samsung S7	USB	Samsung S7	USB
	(1)	(2)	(3)	(4)
hardware+software	-0.001	-0.151***	0.027***	0.279***
	(0.001)	(0.019)	(0.008)	(0.026)
hardware	-0.001	-0.139***	-0.021***	0.147***
	(0.001)	(0.019)	(0.007)	(0.026)
software	-0.000	-0.006	0.011	0.113***
	(0.001)	(0.020)	(0.007)	(0.027)
constant	1.000***	0.941***	0.044***	0.170***
	(0.0004)	(0.014)	(0.004)	(0.019)
Observations	6,134	2,487	6,134	2,487
R2	0.001	0.042	0.005	0.047
Adjusted R2	0.00004	0.041	0.005	0.046
Residual Std. Error	0.018 (df = 6130)	0.339 (df = 2483)	0.211 (df = 6130)	0.452 (df = 2483)
F Statistic	1.072 (df = 3; 6130)	36.445*** (df = 3; 2483)	10.898*** (df = 3; 6130)	40.604*** (df = 3; 2483)
Note:				*p<0.1; **p<0.05; ***p<0.01

***Figure 10**: 5-9 Feet Regression Results*

While the detection rate showed similar results, the recognition mirrors the hardware's improvement in detection 5-9 feet away. The USB experience over 10% increase in recognition rate. The hardware + software solution even gained 27.9% accuracy. This increase may be explained by the effectiveness of the USB within closer range. Figure 11 shows how much the USB camera improved at a closer distance only being outperformed by the Samsung S7 be approximately 13%. The remainder of the trials shows a similar trend between the cameras detection abilities.

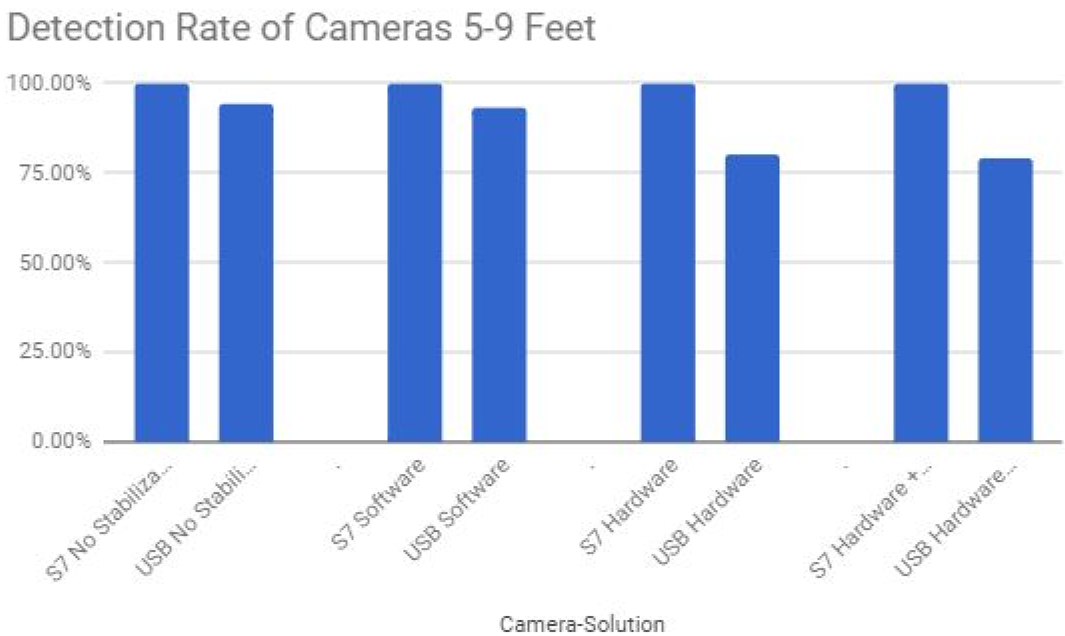


__Figure 11__:5-9 Feet Detection Rate by Phone

C. Sudden Stop

For the last distance of the Trump subject segment the turtlebot reached its first location and stopped rather abruptly. This sudden stop created noticeable sway of the PVC pipe which held the camera mount. Not only did this sway the PVC pipe but the hardware suffered greatly in trying to stabilize such a drastic shift in movement. The USB hardware suffered a -50% drop in detection rate during this unique observation.

	Dependent variable:			
	detection		recognition	
	Samsung S7	USB	Samsung S7	USB
	(1)	(2)	(3)	(4)
hardware+software	0.000	-0.497***	-0.004*	0.121***
	(0.000)	(0.023)	(0.002)	(0.021)
hardware	0.000	-0.505***	-0.004*	0.077***
	(0.000)	(0.023)	(0.002)	(0.021)
software	-0.000*	-0.008	0.003	0.013
	(0.000)	(0.024)	(0.002)	(0.022)
constant	1.000***	0.941***	0.004***	0.115***
	(0.000)	(0.017)	(0.001)	(0.016)
Observations	6,027	2,429	6,027	2,429
R2	0.500	0.279	0.002	0.017
Adjusted R2	0.500	0.278	0.001	0.016
Residual Std. Error	0.000 (df = 6023)	0.399 (df = 2425)	0.060 (df = 6023)	0.374 (df = 2425)
F Statistic	2,007.718*** (df = 3; 6023)	312.466*** (df = 3; 2425)	3.366** (df = 3; 6023)	13.851*** (df = 3; 2425)
Note:				*p<0.1; **p<0.05; ***p<0.01

***Figure 12**: Sudden Stop Regression Results*

Interestingly, at a very short distance, the phone did not detect the picture of Donald Trump while the lower quality USB camera was able to. This outlier in the data is unexplainably attributed to the Samsung S7 believing the image of Trump was Obama. When looking just at the USB data the hardware recognition rate interestingly benefitted from the hardware despite the drastic shaking. This leads to the conclusion that while it was unable to detect Trump well during the extreme shaking, the hardware was able to correctly identify the few frames that it did detect someone. At the same time, the software was unable to handle these episodes which provides insight into the benefit of the hardware for extreme circumstances.

D. Rotation

The second part of the course involved the robot rotating in place while attempting to identify the image of Hillary Clinton. Figure 11 shows that for both cameras the hardware negatively impacted it ability to detecting and recognize Clinton. This is surprising due to the robot rotating in place and there was little to no sway in the PVC pipe leading. This leads me to the conclusion again that the hardware caused the camera to sway side to side during the rotation due to the USB's wire.

	Dependent variable:			
	detection		recognition	
	Samsung S7	USB	Samsung S7	USB
	(1)	(2)	(3)	(4)
hardware+software	-0.032***	-0.285***	-0.048***	0.146***
	(0.012)	(0.015)	(0.013)	(0.013)
hardware	-0.015	-0.287***	-0.036***	0.082***
	(0.012)	(0.015)	(0.013)	(0.013)
software	-0.013	-0.007	-0.006	0.049***
	(0.011)	(0.016)	(0.012)	(0.014)
constant	0.768***	0.755***	0.411***	0.126***
	(0.008)	(0.011)	(0.009)	(0.009)
Observations	10,929	7,314	10,929	7,314
R2	0.001	0.083	0.002	0.018
Adjusted R2	0.0004	0.082	0.001	0.018
Residual Std. Error	0.430 (df = 10925)	0.469 (df = 7310)	0.488 (df = 10925)	0.395 (df = 7310)
F Statistic	2.394* (df = 3; 10925)	219.989*** (df = 3; 7310)	6.057*** (df = 3; 10925)	44.601*** (df = 3; 7310)
Note:				*p<0.1; **p<0.05; ***p<0.01

Figure 13*: Rotational Regressional Results*

E. Face at the Edge of the Video

The final leg of each trial consisted of the robot driving past the final image of Barack Obama. This part of experiment aimed to quantify the solutions benefit to what is comparable to human peripheral vision. Figure 14 shows us that this attempt to capture a significant moment is unpredictable with no consistent results for any solutions. This may be attributed to the unpredictable nature of the software as it trims off part of the video to create a stable image along with the hardware's demonstrated unpredictability.

	Dependent variable:			
	detection		recognition	
	Samsung S7	USB	Samsung S7	USB
	(1)	(2)	(3)	(4)
hardware+software	0.017	-0.002	0.013	-0.064***
	(0.013)	(0.015)	(0.014)	(0.013)
hardware	0.032***	0.285***	0.048***	-0.146***
	(0.012)	(0.015)	(0.013)	(0.013)
software	0.018	0.278***	0.042***	-0.097***
	(0.012)	(0.016)	(0.013)	(0.013)
constant	0.737***	0.470***	0.363***	0.272***
	(0.009)	(0.011)	(0.010)	(0.009)
Observations	10,929	7,314	10,929	7,314
R2	0.001	0.083	0.002	0.018
Adjusted R2	0.0004	0.082	0.001	0.018
Residual Std. Error	0.430 (df = 10925)	0.469 (df = 7310)	0.488 (df = 10925)	0.395 (df = 7310)
F Statistic	2.394* (df = 3; 10925)	219.989*** (df = 3; 7310)	6.057*** (df = 3; 10925)	44.601*** (df = 3; 7310)
Note:				*p<0.1; **p<0.05; ***p<0.01

***Figure 14**: Face on Edge of Screen Regressional Results*

F. Recognition Confidence

Throughout all the trials, one thing was consistent. Regardless of the solution or camera being used, the confidence of correctly recognized faces was consistent. (See Figures 15 and 16). This is surprising because even with the large fluctuations in detection and recognition rate, the confidence levels showed no fluctuations based on the solutions.

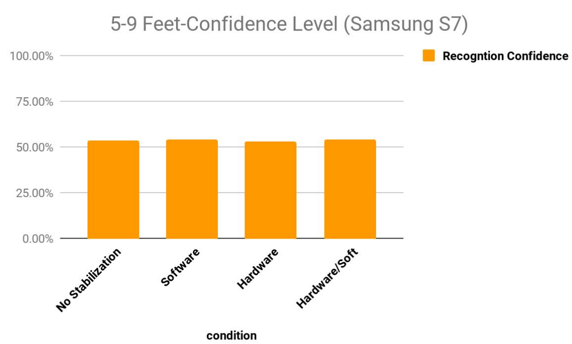


***Figure 15**: 5-9 Feet Confidence Levels-Phone Data*

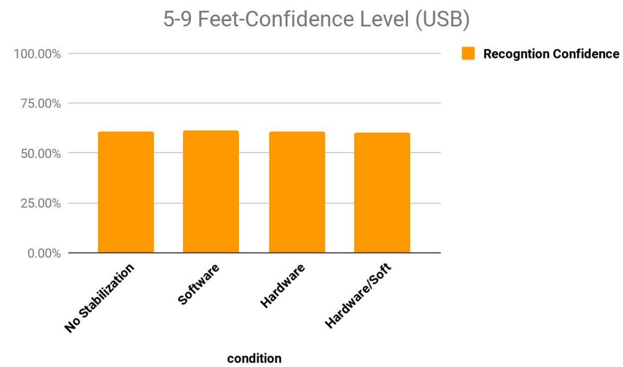


Figure 16: *5-9 Feet Confidence Levels-USB Data*

G. Overall Impact

If we perform the same analysis for all 52,000 frames of data collected, the results are consistent with our observations above. Ultimately, it appears that the solutions negatively affect detection rate while benefiting recognition rate. The USB camera saw a great decrease in detection most likely due to wire issue previously mentioned.

	Dependent variable:			
	detection		recognition	
	Samsung S7	USB	Samsung S7	USB
	(1)	(2)	(3)	(4)
hardware+software	-0.017***	-0.219***	0.117***	-0.006
	(0.005)	(0.010)	(0.007)	(0.009)
hardware	-0.004	-0.210***	0.107***	-0.014
	(0.005)	(0.010)	(0.007)	(0.009)
software	-0.031***	-0.005	0.084***	-0.006
	(0.005)	(0.011)	(0.006)	(0.009)
constant	0.886***	0.698***	0.239***	0.209***
	(0.003)	(0.007)	(0.004)	(0.006)
Observations	36,782	16,678	36,782	16,678
R2	0.001	0.046	0.012	0.0002
Adjusted R2	0.001	0.046	0.011	-0.00002
Residual Std. Error	0.331 (df = 36778)	0.481 (df = 16674)	0.459 (df = 36778)	0.402 (df = 16674)
F Statistic	16.618*** (df = 3; 36778)	270.784*** (df = 3; 16674)	142.725*** (df = 3; 36778)	0.910 (df = 3; 16674)
Note:				*p<0.1; **p<0.05; ***p<0.01

Figure 17: *Overall Regression Results by Camera*

The solutions did provide another major insight however; quality of image matters significantly. Intuitively that is a true statement, however, Figure 18 shows that the USB camera suffered a -28.1% and -11.1% loss in detection and recognition rates respectively to its Samsung S7 counterpart.

	Dependent variable:	
	detection (1)	recognition (2)
hardware+software	-0.079***	0.081***
	(0.005)	(0.005)
hardware	-0.066***	0.073***
	(0.005)	(0.005)
software	-0.016***	0.060***
	(0.005)	(0.005)
USB	-0.281***	-0.111***
	(0.004)	(0.004)
Constant	0.909***	0.260***
	(0.003)	(0.004)
Observations	53,460	53,460
R2	0.111	0.018
Adjusted R2	0.111	0.017
Residual Std. Error (df = 53455)	0.387	0.442
F Statistic (df = 4; 53455)	1,670.066***	238.542***
Note:	*p<0.1; **p<0.05; ***p<0.01	

Figure 18*: Overall Recognition Results*

H. Amazon Reviews

Turning our attention back to amazon reviews, only select words and word count proved to be correlated with the customer rating of a product. These results show that at a .1% confidence level if a review is 100 words longer it decreases the rating of a consumer by .2 for a five point scale (See Figure 19). This I believe is explained by longer reviews being submitted by frustrated consumers who explain in detail the faults they found in a product. If someone is satisfied with a product it appeared from a subjective glance that their reviews are a quick statement about their satisfaction. Although, each product typically had one in depth positive review that customers rated as helpful. Thus, typically the longer a comment was, the more disappointed the consumer was.

	Dependent variable:
	rating
wordCount	-0.002*** (0.001)
quality	0.077 (0.069)
great	0.245*** (0.068)
zoom	0.156** (0.077)
size	0.159 (0.148)
cost	-0.305 (0.208)
price.x	0.045 (0.110)
resolution	-0.139 (0.253)
broken	-0.048 (0.282)
returned	-1.118*** (0.271)
numberReturned	-0.097 (0.146)
price	0.002 (0.001)
cameraAndPhotoRank	-0.002 (0.002)
pointAndShootRank	0.001 (0.001)
opticalZoom	0.047 (0.045)
screenSize	-0.095 (0.104)
foundHelpful	0.00000 (0.001)
Constant	4.431*** (0.344)
Observations	800
R2	0.129
Adjusted R2	0.110
Residual Std. Error	1.212 (df = 782)
F Statistic	6.829*** (df = 17; 782)
Note:	*p<0.1; **p<0.05; ***p<0.01

***Figure 19**: Amazon Review Results*

Not surprisingly the use of the word “great” had a positive impact on rating and “returned” had a strongly negative impact. However, the only other statistical significant word found in the 1000 comments was the word “zoom”. This word had a positive relationship of .15 rating per use of the word. Thus, every use of the word zoom is correlated with a .15 increase to the 5 point rating. Before the data was analysed, I believed “resolution” and other qualities would be more positively related, however it is interesting to see the significance of camera zoom.

IV. Conclusions

It is clear through this empirical approach that these video stabilization solutions have both positive and negative effects to CV. All solutions proved to decrease the ability of the system to detect a face, which is alarming for any project that is using object detection in their research. Software seemed the most beneficial despite a -1.6% impact on detection rate. This minimal loss was compensated by a 6.0% benefit to accuracy. The hardware solution was unexpectedly unable to handle large movements which was where it was expected to thrive. It inability to benefit the image should be attributed to the sensitivity of the hardware along with the complication of attaching it to the robot which may not allow the system to perform as designed. When the video was still enough to detect a face, the solutions proved to increase the rate of recognition. I attribute this to the ability to absorb subtle motions while creating distortions to video during significant shaking.

By and far the most significant discovery of this study was the impact that camera quality has on CV. The 13 megapixel Samsung S7 far out performed the 3 megapixel Logitech USB in all scenarios. Due to these observations, this data suggests that any application using computer vision for objects further than nine feet absolutely need to use a high quality camera. If the designed system is only used on objects within nine feet there still is a reduction but would be practical for use. Along with this camera quality, using a software solution would benefit a robot due to its minimal impact on detection and increasing recognition ability. Hardware creates many complications and increases costs, software solutions are free, can be run in real time, and can be interchanged through code alone.

While resolution certainly has a positive effect on detection/recognition, its significance is not prominent within our extracted Amazon camera reviews. Zoom proved to be an important quality that when mentioned had a positive correlation with rating. Future work on this topic should explore neural networks ability to understand these comments in greater depth. Because this study only tallied the frequency of words it couldn't determine if a word was being used in a positive or negative context. For example, a consumer could write "great quality" or "poor quality" and this method would not be able to extract that meaning. Splicing the ratings into coherent positive or negative uses of these words would be interesting to create a more in depth understanding of camera qualities impact on consumer demand for particular product qualities.

V. Bibliography

Boden, M. A. (2006). Mind As Machine: A History of Cognitive Science. Oxford University Press, Oxford, England.

Bledsoe,W.W.: Man-Machine Facial Recognition: Report oBoden, M. A. (2006). Mind As Machine: A History of Cognitive Science. Oxford University Press, Oxford, England.

Bledsoe,W.W.: Man-Machine Facial Recognition: Report on a Large-Scale Experiment. Technical Report PRI-22. Panoramic Research Inc. California (1966)

Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

Censi, Alberto, Andrea Fusiello, and Vito Roberto. "Image stabilization by features tracking." Image Analysis and Processing, 1999. Proceedings. International Conference on. IEEE, 1999.

Ghose, Anindya, and Panagiotis G. Ipeirotis. "Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics." *IEEE Transactions on Knowledge and Data Engineering* 23.10 (2011): 1498-1512.

Jung, Boyoon, and Gaurav S. Sukhatme. "Detecting moving objects using a single camera on a mobile robot in an outdoor environment." In International Conference on Intelligent Autonomous Systems, pp. 980-987. 2004.

Matsushita Yasuyuki , Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(7):1150–1163, 2006.

Murray, Don, and Anup Basu. "Motion tracking with an active camera." IEEE transactions on pattern analysis and machine intelligence 16.5 (1994): 449-459.

Nordlund, Peter, and Tomas Uhlin. "Closing the loop: Detection and pursuit of a moving object by a moving observer." Image and Vision Computing 14.4 (1996): 265-275.

Rousso, Michal Irani Benny, and Shmuel Peleg. "Recovery of ego-motion using image stabilization." (1993).

Srinivasan, Sridhar, and Rama Chellappa. "Image stabilization and mosaicking using the overlapped basis optical flow field." Image Processing, 1997. Proceedings., International Conference on. Vol. 3. IEEE, 1997.

Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57.2 (2004): 137-154.

Zoghlami, Imad, Olivier Faugeras, and Rachid Deriche. "Using geometric corners to build a 2D mosaic from a set of images." *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997.