Is Now the Time for a Cobot Burger Chef?

Richard Hooper, PhD, PE
Safemachines, PLLC
Austin, Texas

***Abstract*— This paper examines the potential use of robotic manipulators to automate the assembly of burgers at fast food restaurants. More specifically, the paper looks at three basic questions. Is this application feasible with currently available, commercial technology? If not, what is missing? And finally, does it make economic sense to pursue this application? The paper begins by overviewing the work and the technologies required to automate the work. The paper then details an implementation of machine learning (CNN-based image segmentation) applied to the problem. Next, the paper describes a novel end-effector for handling sliced vegetables that was specifically developed for the burger assembly application. Finally, the paper concludes with an examination of the economic considerations around implementing a cobot burger chef.**

Keywords – Image Segmentation, Fast R-CNN, Mask R-CNN, Residual Learning, Cobot, End-effector, burgers.

I. INTRODUCTION

The job "flipping burgers" has been around as long as there have been hamburger joints and it is often used as the classic example of a low-skill job. The jobs of cooking and assembling burgers are ubiquitous, however, they have largely resisted automation and are still being performed almost exclusively by human workers.

This paper considers whether now is the time to automate the work of cooking and assembling burgers. Of course, there are different varieties of automation. This paper focuses on the use of robotic manipulators. Thus, the question is whether now is the time to use robotic manipulators to automate this work? This question, in turn, breaks into three more basic questions. Is it feasible with currently available, commercial technology? If not, what is missing? And finally, does it make economic sense to do it?

To understand whether the cobot chef is technically feasible, this paper considers three main subsystems:

1. Robotic manipulator - Traditional robotic manipulators with the performance specifications (payload, speed, dexterity, etc.) to do the work have been around for decades. The work of assembling burgers is not, however, safe to automate with traditional robotic manipulators because it needs to be performed side-by-side with people. Fortunately, collaborative robotic manipulators (cobots) have been developed to address many of the safety issues with traditional robotic manipulators. Cobots have become a relatively mature technology and are commercially available with performance specifications appropriate for the tasks of cooking and assembling burgers. [6]

2. Image recognition - The cobot burger chef needs to identify and locate objects in an image. For example, it needs to identify a specific portion of sliced vegetable, say a sliced tomato, to pick out of a bin of sliced tomatoes. This task is an obvious fit for AI-based vision systems. Most of us have seen examples of this type of image recognition in the context of autonomous cars, but these cars have extremely powerful (and expensive) computers inside them. Is this task also feasible in the cobot burger chef application using more modest computing hardware? This paper presents a test case implementation.

3. End-effector - People are currently using their hands to do the burger assembly work. Robotic hands have been under development for at least forty years, and they are still many years from being commercially available. [9] If the cobot burger chef is going to be viable in the near term, low-cost end-effectors specialized for the burger application will need to be developed. This paper overviews the development of one such end-effector. This end-effector focuses on the task of picking sliced vegetables from a bin and then placing them onto a burger.

The final section of this paper addresses questions of whether it currently makes economic sense to deploy a cobot burger chef. The potential market size is always a part of economic considerations, and this section presents data around the market size for cobot burger chefs. This section also develops and presents analyses of the Payback Period, Return On Investment (ROI), and Compound Annual Growth Rate (CAGR) of a capital investment in a cobot burger chef. Since the comparison is to human labor, the results of these analyses are tabulated as a function of wage.

II. WORK TO BE AUTOMATED

Figure 1. Assembling burgers is the work to be automated

Figure 1 shows a line of three workstations where burgers are cooked and assembled. At the far right station condiments are added to the burger. Theses condiments include sliced

vegetables such as pickles, tomatoes, onions, and lettuce. Sauces such as catsup, mustard, and mayonnaise are also applied at this station. Though a cobot burger chef could potentially work all three stations in this line, it is the far right station that is the specific subject of this paper.

III. Applied Machine Learning

To assemble the burger, the cobot burger chef needs to identify a single portion of sliced vegetable to pick out of a bin of sliced vegetables. Convolutional Neural Networks (CNNs) have demonstrated great success in image identification and segmentation tasks like this. The discussion below details the application of publicly available software (torchvision [10]) for implementing CNNs within the cobot burger chef application.

The torchvision software, and all the foundational software it is built upon, represents hundreds of thousands of hours of development and research work by many smart and talented people. These smart and talented people have packaged this functionality into pre-trained and publicly available CNN models that are well documented and readily deployable by competent software engineers.

They though they are pre-trained, the CNN models need a final layer of training specific to the application at hand. Developing these training images is a laborious process that involves capturing and labelling (typically by hand) many images, though it is orders of magnitude easier than what would be required if pre-trained models were not available.

A. Image Capture, Preparation and Labeling

This test case utilized four training and verification sets. Beginning with the images shown in Figure 2, a human operator sequentially removed a single, exposed slice (as if they were taking a slice to put on a burger) and then captured another image. This process repeated until all the slices were removed from the trays. In all, 60 images were taken of the oblong plate, 60 images were taken of the round plate, 169 images were taken of the rectangular flower plate, and 38 images were taken of the square dish. The images were 629 x 472 pixels and 24 bit color with a size on disk of about 500 kb per image.

Figure 2. Four training and verification sets

After the images were acquired, the work of labelling began. A human operator needed to identify the tomato slices in each image that were free and clear to be picked from the plate.

In this implementation we simply used PowerPoint to allow a person to place and size ovals over the tomato slices that were free and clear. We allowed up to seven slices to be identified per image and used seven different discrete gray scale values to differentiate each instance. There are many publicly available labelling packages available for more complex applications, but PowerPoint was familiar and the oval shape matched the shape of the tomato slice well.

After the images were labelled, an image processing program was written using the Python programming language to replace the background in the labelled images with pure black and draw bounding boxes around the ovals. This created the masks that were required for final training of the CNN. As with the original images, the masks were 629 x 472 pixels, but with only 8 discrete values stored in an 8 bit format and a size on disk of less than 1 kb.

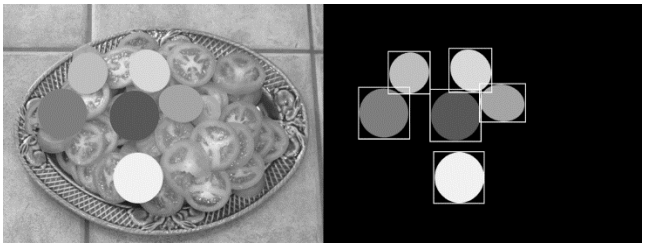

Figure 3. Masks used for training

The image on the left in Figure 3 shows the tray of tomatoes after it had been labelled by the human operator. The image on the right shows the corresponding mask and bounding boxes generated by the image processing software.

B. Convolutional Neural Network (CNN)

This example used the ResNet-50 CNN model pre-trained on the CoCo database as the starting point. These are publicly available at no cost.

ResNet-50 is a CNN that is 50 layers deep. It has 48 convolution layers, 1 Max Pool layer and 1 Average Pool layer. It is a type of CNN known as a Residual Network model. [3]

"CoCo" stands for "Common Objects in Context." This dataset was developed by Microsoft and includes photos of 91 object types with a total of 2.5 million labeled instances in 328k images. The object types include items such as person, dog, sheep, sunglasses, cell phone, chair, and other items that would be easily recognizable by a 4 year old. [4]

The software developed for the sliced tomatoes application replaced the pre-trained head with a Fast R-CNN predictor head and replaced the mask predictor with a Mask R-CNN Predictor. [1][2] Processing the model in this way simply required calling pre-packaged, publicly available software modules.

C. Model Training and Results

As is common practice, the images and masks were divided into two equal groups: training and verification. The hardware used for the training and verification was quite modest. The processor was an Intel (R) Core (TM) i7 3770 CPU @ 3.4 GHz

with 16 MB of RAM and the GPU was a GeForce GTX 1660 6GB GDDR5.

Training using the images and hardware described above took approximately 7 minutes. Once the model was trained, new images were processed at a rate of approximately 2 images per second. Without exception, the model worked properly on all the verification images, including some that seemed especially challenging because the tomato slices were stacked, or located on a challenging background. Figure 4 shows a test case that seemed challenging.

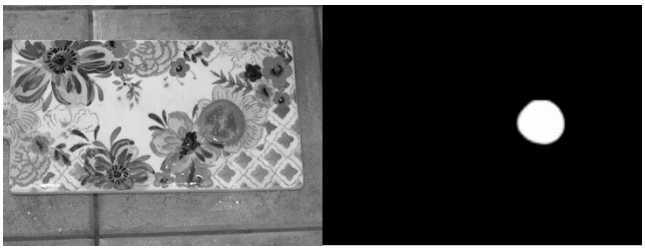

Figure 4. Image on flower background

Figure 4 is a composite of two images. The image on the left shows a tomato slice located directly over a camouflaging background on the tray. The image on the right was automatically produced by the CNN. It is a segmentation mask that identifies the location of the tomato slice on the tray. In this test case we see the trained CNN properly found the tomato slice even though it was located over a challenging background.

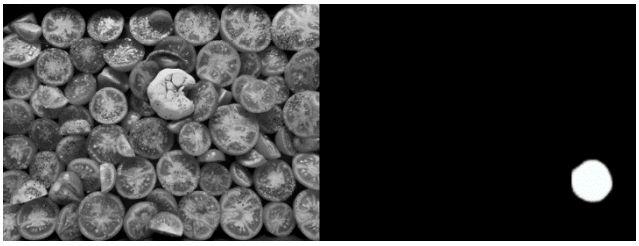

Figure 5. Image from the web where the model passed

Figure 5 shows the results of test where the CNN was presented with an image that was found on the web. In this case the CNN correctly identified a good tomato slice for the robot to pick.

Figure 6. Images from the web where the segmentation failed

Figure 6 shows two images found on the web where the CNN did not properly segment a tomato slice. Training with a larger and more varied collection of labelled images should improve the segmentation performance of the CNN, but at the expense of more time spent acquiring and labelling training images. Though it is certainly feasible to acquire and label all the training and verification images that will be required to deploy the cobot burger chef, it should be acknowledged that this will be a considerable effort.

IV. End-Effector

The end-effectors used by robotic manipulators are typically quite specific to a given task and the cobot burger chef application is no different. These end-effectors will need to be developed if the cobot burger chef is to be realized. One of the goals of the current work is to begin developing the end-effectors that will be required to deploy the cobot burger chef.

The basic requirements for the end-effector are familiar to us. The cobot burger chef needs to pick slices of pickles, tomatoes, onions, and lettuce out of bins and place them onto buns. The cobot chef also needs to dispense ketchup, mustard, and other sauces onto the burger at this station. This will likely require an end-effector that is different than the end-effector that handles sliced vegetables so some tool change capability will ultimately be required.

Figures 7 through 10 illustrate a novel end-effector for handling sliced vegetable toppings that was developed as part of this work. Note this end-effector design requires no vacuum to capture the vegetable slices. This improves both the cleanliness and speed of operation when compared to end-effectors that require vacuum. Speed, reliability, cost, and safety are important criteria for evaluating any automation solution, but the requirements for washing and cleaning are notable in food applications. As will be discussed, the end-effector has other attributes that make it suitable for this cobot burger chef application.

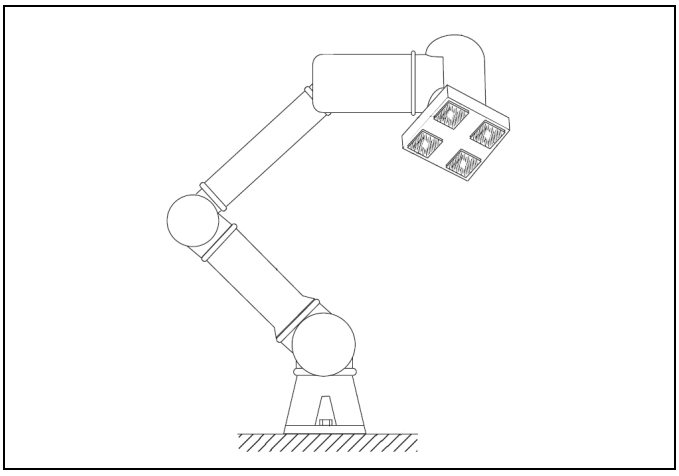

Figure 7. Depiction of end-effector on UR5e (TM) cobot

Figure 7 depicts the end-effector mounted to a Universal Robots (R) UR5e (TM) cobot. Though the end-effector is largely agnostic regarding the manipulator it is attached to, the UR5e is a reasonable choice in this application for a number of reasons. To start with, the UR5e is a cobot. As discussed, this is a requirement for the manipulator. Next, the workspace of the UR5e robot extends radially 850 mm from the base joint. [5]

This would allow the robot to be placed in the same workspace that was previously occupied by a person and reach the entire work volume. Finally, the UR5e has a 5 kg payload. [5] This is sufficient for the robot to perform its tasks with some headroom (the end-effector weighs less than a pound).

There are four locations on the end-effector that are each individually controllable to capture and place a sliced vegetable. Each location includes an array of short and stiff stainless steel wires that project out from the bottom of the end-effector. The robot presses this array into the sliced vegetable to pierce and capture it.

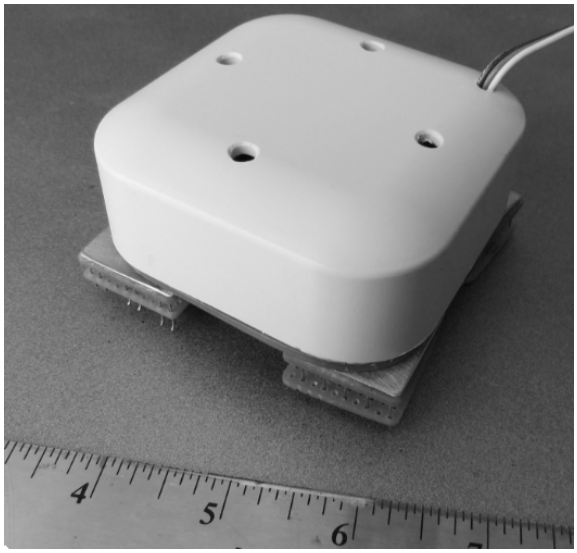

Figure 8. End-effector for picking and placing sliced vegetables

Figure 8 shows a prototype end-effector for the cobot chef. The generally square shape allows it to reach sliced vegetables in the corners of bins and containers.

The hole pattern at the top of the end-effector is used for attaching to the robot's mounting plate and conforms to the ISO 9409-1-50-4-M6 50 mm pitch mounting plate specification. This is also the end-effector mounting interface of the UR5e cobot.

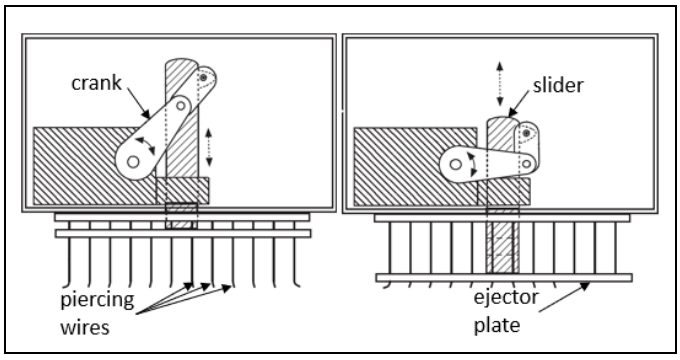

Figure 9. End-effector capture and eject sequence

Figure 9. shows a side view of the servo mechanism that captures and ejects the sliced vegetables. The drawing on the left shows the capture position with the ejector plate fully retracted. By adjusting the retraction distance, the mechanism can accommodate different thicknesses of sliced vegetables. The drawing on the right shows the mechanism with the ejector plate in the fully extended position, as it would be after it has ejected the sliced vegetable. The end-effector employs slider-crank mechanisms to convert the rotational motion of the servos into translational motions of the ejector plates.

Note the ends of the piercing wires have small bends in them that help hold the sliced vegetables in place after they have been pierced by the wires and then lifted by the robot.

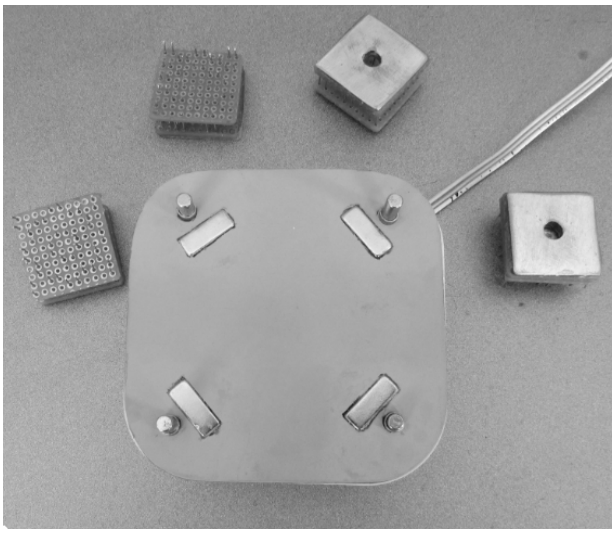

Figure 10. Capture mechanisms are held in place with magnetic forces

Figure 10 shows the four capture mechanisms removed from the main end-effector body. Note the capture mechanisms are aligned and held in place entirely by magnetic forces. This allows them to be easily removed for cleaning in a dishwashing machine. It also allows the bottom of the end-effector to be sealed so that it can be washed down. Also note the three-wire cable coming from the end-effector body. This supplies power, ground, and a communications link to energize and control the four servo mechanisms inside the end-effector.

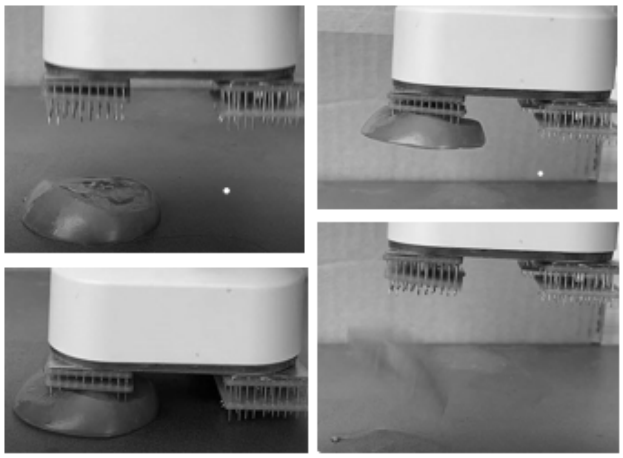

Figure 11. End-effector capture, move and eject of tomato slice

Figure 11 shows a capture, move, and eject sequence of the end-effector. In the top left image, the robot has positioned the end-effector over a tomato slice. The bottom left image shows

the robot has pushed the end-effector onto the tomato slice to pierce and capture it with the array of wires. The top right image shows the robot has picked up the tomato slice and positioned it to be ejected. The bottom right image shows the slice immediately after ejection by the end-effector.

V. Economic Analysis

As shown in Table 1, the potential market for cobot burger chefs is quite large. Assuming a viable cobot burger chef were available for purchase, the question becomes one of economic justification. This section evaluates the cobot burger chef using three common metrics of economic motivations for deploying automation: payback period, Return On Investment (ROI) and Compound Annual Growth Rate (CAGR).

The payback period is the most elementary of these metrics. It is the amount of time when the investment in the automation "pays for itself" when compared to the same task being performed by a human. The ROI expresses how the investment pays off over time as a simple interest rate. The CAGR is similar to ROI but facilitates comparison to alternative investments that compound over time.

Table 1. below shows fifteen of the biggest burger restaurant chains in the United States. This table was created by gathering data from each of the restaurants' websites and from other publicly available sources. It is not meant to be a definitive list, but rather to show the magnitude of this automation opportunity.

Table 1. Burger restaurant locations in the United States

Restaurant chain	Num US locations
McDonalds	14000
Burger King	7750
Wendys/Arbys	6576
Dairy Queen	5050
Sonic	3500
Jack in the Box	2100
Hardees	1687
Carl's Jr	1104
Checkers	800
Five Guys	735
Whataburger	575
Culver's	438
White Castle	421
A&W	359
In-N-Out	258
Total	**45,353**

As discussed, a common economic analysis applied when deploying automation is the computation of the payback time. This is the amount of time for the investment in the automation to "pay for itself" when compared to the same task being performed by a human.

The calculation of the cost of the human is relatively easy to calculate. It includes the hourly wage and a multiplier to account for taxes and benefits. This total hourly rate is multiplied by the hours worked (payback time) to get the total cost of the person. This may be expressed as follows:

$$\textit{total cost of person} = T_P * C_P,$$

where T_P = payback time and C_P = total hourly rate of person.

The calculation of the cost of the robot system is slightly more complicated. Of course, it includes the purchase price of the robot system, but it also includes the recurring costs of tending the robot, maintaining the robot and a lifecycle cost to account for the robot system's eventual obsolescence.

The recurring cost to tend the robot is represented by a tending multiplier multiplied by the total hourly rate of a human multiplied by the hours worked (payback time). This may be expressed as follows:

$$\textit{recurring cost of tending robot system} = M_T * C_P * T_p,$$

where T_P = payback time, C_P = total hourly rate of person and M_T is the tending multiplier.

The recurring cost to maintain the robot is represented by a maintenance multiplier multiplied by the cost to purchase the robot system multiplied by the hours worked (payback time). Note the maintenance multiplier must include not only parts and labor to repair any issues with the system, but also downtime of the system associated with the repairs. This total cost may be expressed as follows:

$$\textit{recurring cost of maintaining robot system} = M_M * C_R * T_p,$$

where T_P = payback time and C_R = cost to purchase robot system and M_M is the maintenance multiplier.

The lifecycle cost to account for the robot system's eventual obsolescence is represented by a lifecycle cost multiplier multiplied by the cost to purchase the robot system multiplied by the hours worked (payback time). This may be expressed as follows:

$$\textit{lifecycle cost to account for obsolescence} = M_L * C_R * T_p,$$

where T_P = payback time and C_R = cost to purchase robot system and M_L is the lifecycle cost multiplier.

Setting the cost of the human equal to the cost of the robot gives:

$$T_P * C_P = C_R + M_T * C_P * T_p + M_M * C_R * T_p + M_L * C_R * T_p,$$

and rearranging for payback time gives:

$$T_P = C_R / (C_P - M_T * C_P - M_M * C_R - M_L * C_R).$$

Actually solving for payback time requires values to be substituted into the equation above. Table 2 below shows the values used in the calculation results shown in Table 3.

Table 2. Factors used in calculations

Value	Factor
10 years	Useful lifespan of system
.1 years	Maintenance multiplier
0.25 persons	Tending multiplier
$85,000	Cost of system
4,284 hours	Hours worked per year
1.283 persons	Benefits multiplier

The following discussion details the rationale behind the values shown in Table 2. Because no cobot burger chef has been built or deployed, some of the factors in Table 2 are necessarily estimates. Where estimates are required, the rationale behind the estimated values is discussed.

Ten to fifteen years is the range of typical lifecycles for a robotic system deployment. [5] The conservative end of this range (10 years) was used in the calculation of payback time for the cobot burger chef.

The maintenance multiplier of 0.1 implies that every year maintenance and downtime will represent a recurring cost equal to 10% of the purchase price of the system. The cobots themselves will have an MTBF of at least 40,000 hours, but the end-effectors, wiring, sensors and the like often reduce the MTBF to the range of 500-1000 hours. [5] Experience with automatic milking machines in the dairy industry suggests the annual maintenance may be closer to 3% ($7,000 yearly maintenance cost for a $220,000 system [8]), but 10% was used as a more conservative estimate.

As discussed, the tending multiplier acknowledges the fact that the robot system will need to be tended by a human worker for addressing issues where the robot gets stuck, mishandles condiments, etc. This is analogous to the grocery store clerk that supports a number of shoppers using automated checkout stations. The tending multiplier of 0.25 persons implies a single human worker could support four robot burger chefs, or similarly, spend 25% of their time supporting one robot chef.

The purchase price of the robot system includes the cost of the robot and controller ($35,000), vision system ($3,000), end-effectors ($3,000), cables ($1,000), and stand ($500). This gives a cost of $42,500 for the system and this is doubled to arrive at the $85,000 purchase price shown in Table 2.

The hours worked was calculated as follows:

4,284 hrs/yr = 12 hrs/day * 7 days/week * 51 weeks/ yr.

Table 2 shows the multiplier to account for the taxes and benefits associated with a human employee to be 28.3%. This number is taken from a U.S. Bureau of Labor Statistics report. [7] Specifically, the number is for a civilian worker in the service industry.

Table 3. Calculation results summary

Wage ($$$/hour)	Payback (months)	5 yr. ROI	5 yr. CAGR
9	50.7	18.2%	3.4%
10	42.1	42.5%	7.4%
11	36.0	66.7%	10.9%
12	31.4	91.0%	13.8%
13	27.9	115.2%	16.6%
14	25.1	139.5%	19.1%
15	22.8	163.7%	21.4%

Table 3 shows the results of the payback period, ROI and CAGR analyses using the factors listed in Table 2. These results are tabulated as a function of human employee wage in dollars per hour. Based on the author's experience, the calculations at $9/hr do not make a good case for deploying the automation while the calculations at $15/hr make a strong case for the automation.

VI. Conclusion

This paper looked at whether now is the time for robots to begin taking on the job of assembling burgers at fast food restaurants. More specifically, the paper looks at three basic questions. Is this application feasible with currently available, commercial technology? If not, what is missing? And finally, does it make economic sense to pursue this application?

The past decade has seen two of the key technologies required to implement the cobot burger chef mature and become commercially available. The first is the cobot that can safely work alongside humans. The second is AI-based scene processing and image recognition. Both of these were developed as general purpose technologies with a wide range of applications. The end-effectors needed by the cobot burger chef, however, are quite specific. This paper discussed an end-effector designed to handle sliced vegetables, but substantial work developing a suite of end-effectors remains before the technology is in place to deploy the cobot burger chef.

From a business perspective, the paper presented analyses of payback period, ROI, and CAGR that showed the cobot burger chef is not an attractive proposition, even if it were technologically feasible, when compared to human labor at $9/hr. When the human labor was priced at $15/hr, the cobot burger chef became the more economically attractive alternative.

References

[1] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

[2] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[3] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[4] Lin TY, et al. "Microsoft COCO: Common Objects in Context." In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham.

[5] A. Kampa, " The Review of Reliability Factors Related to Industrial Robots," Robotics and Automation Engineering Journal, Volume 3, Issue 5, November 2018.

[6] Universal Robots e-Series User Manual, UR5e, Version 5.0.2 by Universal Robots A/S, 2018.

[7] "Employer Costs for Employee Compensation Summary," U.S. Bureau of Labor Statistics, USDL-21-0437, March 2021

[8] K. Schulte and L. Tranel, "The Economics of Automatic Milking Systems," ISU Extension Publication LT-KS-2013-1.

[9] 2013J. Szkopek and G. Redlarski, "Artificial-Hand Technology—Current State of Knowledge in Designing and Forecasting Changes," Appl. Sci. 2019, 9, 4090.

[10] S. Marcel, et al, "Torchvision the machine-vision package of torch," Proceedings of the 18th ACM international conference on MultimediaOctober 2010.