EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Application of Python and OpenCV on industrial cycle time study

Anintaya Khamkanya*, Suphanath Promteravong and Sirapop Thongampa

Department of Industrial Engineering, Faculty of Engineering, Thammasat School of Engineering, Thammasat University, Pathum Thani 12120, Thailand

Received 10 October 2022
Revised 3 January 2023
Accepted 16 January 2023

Abstract

Motion and time study enhance business performance by improving productivity. The cycle times are collected repetitively to confirm their accuracy and precision. The time taken to complete tasks by engineers or technicians varies due to work experience and educational background, which can cause a large number of repetitions in the time observation. Recent technologies to help shorten time and motion studies as process improvement include digital stopwatches and mobile applications. However, these only reduce documentation time, not observation time. Therefore, this project integrated machine vision technology to reduce observation time in a motion and time study project. The proposed algorithm was developed using OpenCV with Python. Cycle time, measured by the proposed algorithm using work process videos, was then compared with cycle time observed by human appraisers. Results confirmed that the cycle time detected by the proposed algorithm differed from the cycle time evaluated by appraisers with lower variation, i.e., requiring less replicates of observations. Outcomes of this research can be used to shorten the time study process and facilitate remote monitoring in process improvement projects.

Keywords: Machine vision, Artificial intelligence, Process improvement, Work study

1. Introduction

Motion and time study, also called industrial work study, is a systematic method of analyzing and evaluating human work in a production process [1, 2]. The initial data required to conduct a motion and time study is cycle time, which records the time from the start of a particular working activity to the end of that working activity. The motion and time study projects are normally performed by an engineer or a technician, who hereafter will be referred to as an appraiser. The statistical foundation of the time study is the cycle time. This is observed repetitively by the appraiser to compensate for variations in different work cycles and human error. The time study process could take a week or more for time observations in a large factory due to the required number of repetitions of cycle time data and production schedules. Some recent technologies to help speed up the time study are a digital stopwatch and time study applications. However, these techniques only reduce the time of documentation and calculation steps, not the time of observation steps in the project.

Since time study plays a significant role in process improvement, there is software available in the market advertises the ability to conduct time study automatically using artificial intelligence (AI). The first example is mcframe MOTION software [3]. This software tracks workers' motions using AI and machine vision technology using a learning system that operates by a color tracking method to follow the movement of workers' hands or body in a large area. A video recorded using a 3D depth sensor camera is then processed through the software to obtain the cycle time. However, the computer connected to the software must have a high-performance processing unit and disk space to analyze the 3D video smoothly. The second example is GAO-RFID People Tracking System for Manufacturing Facilities software [4]. This software records how employees move throughout the facility by tracking the RFID signal of attached devices. RFID tags, such as lanyard badges, ID badges and helmet tags must be worn by the workers during the observation time. Adequate numbers of RFID reader equipment and antennas must be installed in the tracking zone. Both these software programs can track and measure cycle time; however, investment is required in additional equipment and the cost of the software license may be unaffordable for SMEs.

To resolve these problems and generate productive opportunities, this study integrated an open-source machine vision technology to measure cycle time and reduce labor costs in a work improvement project. The proposed AI algorithm was developed in Python using Open-Source Computer Vision Library (OpenCV) [5, 6], which is a widely used library in machine vision [7, 8]. Applications of Python and OpenCV in detecting motions or objects have been used to develop an automated motion tracking system to track the riders during the changeover [9], detect moving objects of security cameras [10], in face recognition door lock systems [11, 12] and to detect violation of personal safety distance during the Covid-19 pandemic [13, 14]. Many applications of motion tracking in a production process are presented in the literature [15, 16] but the application of Python and OpenCV in motion and time is limited.

This study focused on the development of algorithms and coding in Python and OpenCV to measure cycle time from a work activity video. The designed test scenarios were used to assess the performance of the developed algorithms. Finally, cycle times observed by the appraisers and the developed algorithm were compared using the Gage R&R study to verify cycle time accuracy obtained from the

*Corresponding author.
Email address: kanintay@engr.tu.ac.th
doi: 10.14456/easr.2023.3

proposed algorithm and appraisers. The research methodology is explained in Section 2, with experimental results, comparison study results and conclusions presented in Sections 3, 4 and 5, respectively.

2. Research methodology

The objective of this research was to develop a computer vision algorithm to analyze the cycle time of workers in a production process from a video. Test scenarios and experiments were created to evaluate the ability of the proposed algorithm. The cycle time obtained from the algorithm was then compared with the observed cycle time by engineers. Statistical analysis was also performed to confirm the similarity of the cycle time observed by AI and the engineers.

2.1 Software and hardware

Python was used as a base program. Python is a programming language that has a high-level built-in data structure, combined with dynamic typing and dynamic binding, and can be coded with a machine learning algorithm. To detect motion, the OpenCV library was used through Python. OpenCV is a library used to automate tasks that involve visualization analysis in Python. Both Python and OpenCV are open-source software, with no investment cost required.

A standard mobile phone was used as the hardware to record the work activity video, with the camera set up firmly on a tripod to avoid a blurred picture. The entire working area was covered so that the movement of the operator’s hands was recorded properly. Recording the video with or without audio did not affect the analysis. The lighting was adjusted to reduce reflection and clearly see the work motion. A cable link connecting the camera to the computer was used to transfer the video. This could also be done through a cloud storage application or Bluetooth. Finally, an entry-level gaming laptop computer was used to code and test the algorithm and run the experiments.

2.2 Test scenarios

The test scenario covers the assembly process of a product, chosen here as a Lego model. In the video, one operator was assigned to assemble the product as ten consecutive units in a laboratory. The finished product and the layout of the workstation are shown in Figure 1. In the video, the operator was asked to wear a red glove on the left hand and a black glove on the right hand during the entire filming session. This color set of gloves was pre-planned to be different from the color of the product. The color of the gloves was used as a variable to track hand movements in the algorithm. The workstation background was set to white to enhance the tracking ability of the algorithm. The workstation should have enough lighting to prevent noise in the video that can interfere with the color-tracking algorithm. The video file type was *.mp4 format.

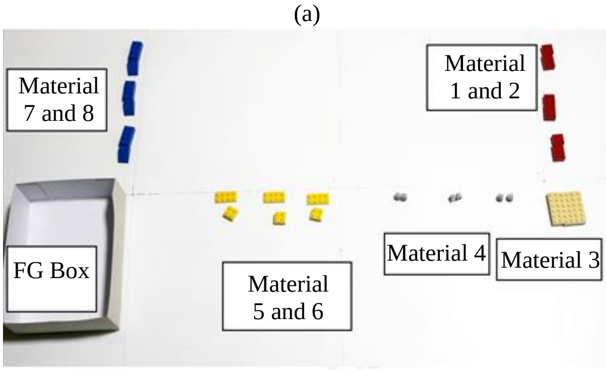
The assembly process is depicted in Figure 2 and detailed as follows.

Process Step 1: The operator places both hands over the finished product box (FG box) that is in the red rectangle. This action triggers the starting point of the cycle time.

Process Step 2: The operator moves the hands to grab the materials and starts the assembly process of material numbers 1 to 8 at the center of the desk.

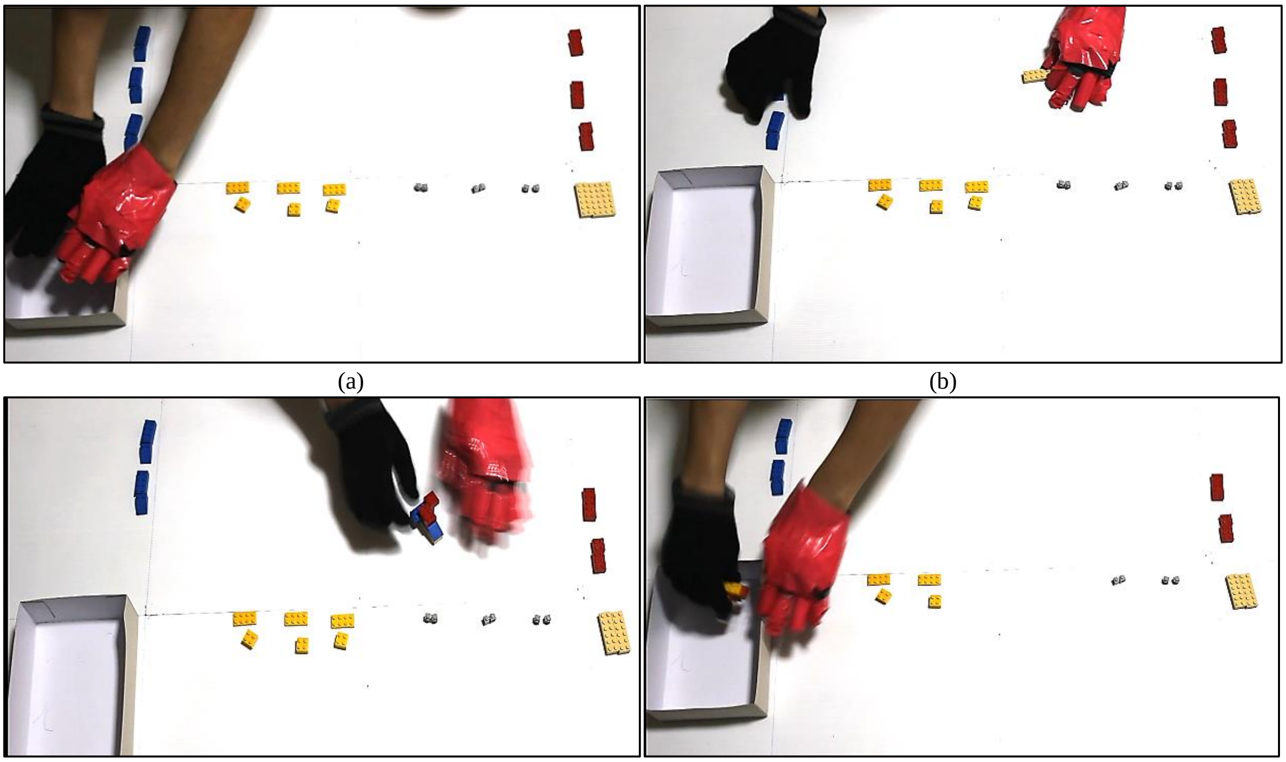
Process Step 3: After the assembly process is completed, the operator moves the hands to store the assembled product in the FG box. When the hands move over the FG box this triggers the code to end the current cycle time and set a clean start to measure the next cycle time. The operator then continues the assembly process by repeating process steps 1-3 until ten assembled products are finished.

(a)



(b)

Figure 1 (a) Finished product and (b) layout of the workstation



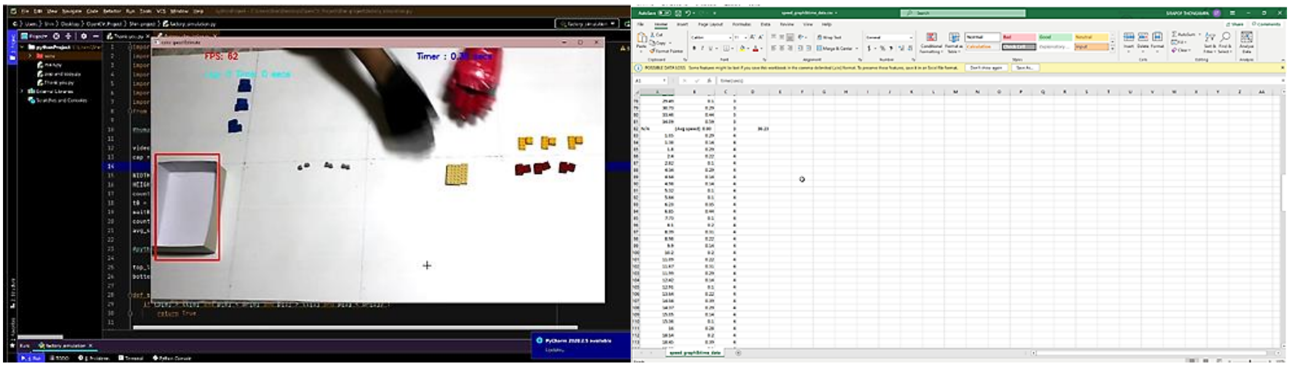
(a) (b) (c) (d)

Figure 2 Motion of (a) Step 1- Move hands from the FG box, (b) Step 2- Grab materials, (c) Step 3- Assembly and (d) Step 4- Put finished product in the FG box

2.3 Development of AI algorithm for time study

Machine vision allows computers to identify objects through digital images or videos. The proposed AI algorithm was designed to measure cycle time from a pre-recorded video of a work process. The time study process started by recording a work process video for a proper number of work cycles. The video was then imported into the program and used for analysis. The result was exported into an Excel file as ready-to-use data for the next step of the time study in a work improvement project. A screenshot of the application and an example of the Excel output are shown in Figure 3. The AI algorithm was developed according to the structure presented in Figure 4. Before running the algorithm, the process information was defined and added to the algorithm as input. The first input was a video of work laps that was then imported into the program and evaluated for the time offset value. Next, the color shade of gloves was assigned in the form of hue-saturation-value (HSV) color space, and used as a tracking variable for the movement. The HSV used as tracking color must be easily distinguishable from the environment to reduce tracking errors. The actual distance of movement in the workstation was then measured and input into the program. This was required as input to adjust the value of pixels per meter in the algorithm. Finally, the position of the area in a scene to start and stop the algorithm was assigned as a pair of corner coordinates to create a rectangle area of a starting-ending point of the work cycle.

The algorithm started the tracking cycle time when the hands came out of the FG box to start the assembly process, as a pre-defined starting-ending point. The cycle time ended when the hands returned to the FG box to place the finished product. At this point, the cycle time was recorded and counted. The display of the algorithm in Python shows FPS, lap and time display, timer and the red rectangle over the white box area. The FPS is the frame rate per second to show whether the video is running smoothly. The lap and time display show the cycle number and the previous cycle time, which was exported as output in Excel. The full video is available at https://bit.ly/TSE-SQDILAB_Vid1. The timer shows the time tracking of a current work cycle, while the red-rectangular area over the FG box presents the area of starting and ending the cycle (Figure 3).



(a) (b)

Figure 3 (a) Screenshot from the application and (b) Excel output

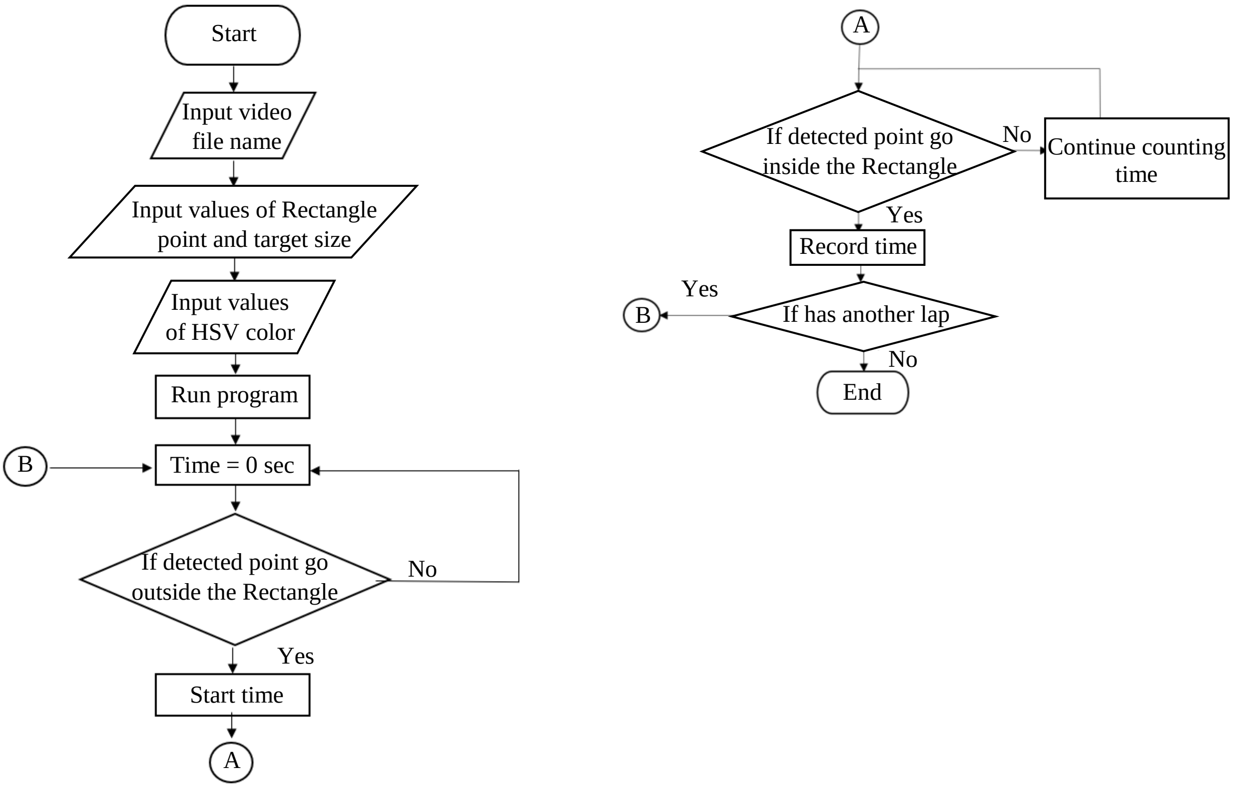


Figure 4 Structure of the AI algorithm

2.4.Experimental design and analysis method

This study employed a two-factor general factorial experimental design [17]. The first factor was the assembly time from the video, which differed across ten work cycles. Thus, the first factor was defined as *Cycle No*, with ten test levels including *Cycle Nos. 1-10*. The second factor was the *Appraisers*, with four levels including the proposed AI algorithm and three industrial engineering students in their senior year who were familiar with industrial motion and time study. The engineering students measured cycle time using a digital stopwatch and recorded the result on a time study sheet. The experiment was run for five replications. Average cycle times measured by the AI algorithm and the appraisers were then compared and the difference was computed to reach the preliminary conclusion, as presented in Section 3. Gage R&R analysis (ANOVA) was conducted to statistically confirm the similarity of cycle times measured by both AI and the appraisers and explain the source of variation in the data, as discussed in Section 4.

3. Experimental results

Average values of the cycle time in seconds obtained from the proposed algorithm (AI), Appraiser 1 (Ap1), Appraiser 2 (Ap2) and Appraiser 3 (Ap3) are presented in Table 1. The cycle time reported by AI was adjusted by subtracting with an offset of 4.4 seconds as the time that the program used to process the video. Thus, the length of the video in the program was longer than the original version. The value of offset time can vary based on the performance of the computer and the resolution of the video. Average cycle times of five replicates of time observations on each work cycle from different appraisers are summarized in Table 1. Differences in the average cycle time were recorded by the different appraisers. The standard deviation (S.D.) values calculated to present variations in cycle time were also different across the appraisers.

Percentage differences in cycle time between AI-Appraiser 1, AI-Appraiser 2 and AI-Appraiser 3 are shown in Table 2. Average percentage differences between AI-Appraiser 1, AI-Appraiser 2 and AI-Appraiser 3 were -0.14%, -2.11% and -0.41%, respectively with overall average percentage difference -0.89% and difference in S.D. 0.02%. Thus, as a primary conclusion, cycle times detected using the developed algorithm in Python and the OpenCV platform returned similar values as observed by engineers.

Table 1 Cycle time recorded from AI and appraisers 1-3

Cycle No.	**Cycle time, seconds**			
	AI adj.	**Appraiser 1**	**Appraiser 2**	**Appraiser 3**
1	26.92	26.19	26.14	26.04
2	26.85	25.35	26.08	25.33
3	23.07	23.55	23.98	23.70
4	27.23	26.96	27.83	27.14
5	20.37	21.12	21.15	21.12
6	25.21	23.99	24.83	24.00
7	31.12	29.04	29.21	29.17
8	26.21	28.46	29.01	28.51
9	28.69	29.99	30.82	30.17
10	30.82	31.85	32.79	32.05
Average	26.65	26.65	27.18	26.72
S.D.	3.138	3.139	3.306	3.194

Table 2 Percentage cycle time differences between the proposed algorithm (AI), Appraiser 1 (Ap1), Appraiser 2 (Ap2) and Appraiser 3 (Ap3)

Cycle No.	Percentage differences			
	AI - Ap1	**AI - Ap2**	**AI - Ap3**	**Average**
1	2.71%	2.88%	3.25%	2.95%
2	5.59%	2.86%	5.65%	4.70%
3	-2.06%	-3.93%	-2.75%	-2.91%
4	1.01%	-2.22%	0.32%	-0.30%
5	-3.70%	-3.83%	-3.70%	-3.74%
6	4.85%	1.51%	4.81%	3.72%
7	6.68%	6.14%	6.25%	6.36%
8	-8.58%	-10.70%	-8.79%	-9.36%
9	-4.53%	-7.42%	-5.16%	-5.70%
10	-3.35%	-6.40%	-4.00%	-4.58%
Average	-0.14%	-2.11%	-0.41%	-0.89%
S.D.	0.00%	0.05%	0.02%	0.02%

4. Comparison of cycle times observed by AI and the engineers

To better clarify the differences in average cycle time and standard deviation observed by AI and the engineers, Gage R&R analysis was used as a statistical tool to determine the result of the two-factor general factorial experimental design in Section 3. Gage R&R analysis is a technique for determining which sources of variation have significant impacts on the measurement data. This statistical technique is widely used by engineers to verify the accuracy of a measurement system before designing an experimental project [17-19]. The main factors involved in this analysis were the different parts and different measurement operators. This study had ten different parts as work cycles 1-10 and four different operators (Appraisers: AI, Ap1, Ap2 and Ap3). Measurements of each combination of the four appraisers on the time observation of ten work cycles were repeated five times, with statistical analysis conducted using Minitab® software [20].

The two-way ANOVA table with interaction of the Gage R&R study is presented in Table 3. The time observed by different appraisers was indifferent, with p-value of 0.316 less than the 95% confidence level. Thus, cycle time measured by AI was similar to the time observed by the three human appraisers. Moreover, from Table 4, the percentage contribution (%Contribution) for part-to-part variation, i.e., variation of time used in each work cycle was 95.03%, meaning that the measurement systems (AI and human appraisers) reliably distinguished between the time used in each work cycle. Thus, the proposed algorithm using Python and OpenCV captured cycle times similar to those reported by the engineers.

The average times observed by AI and the engineers were indifferent but time observations varied for each engineer. Percentage study variations (%SV) of total Gage R&R in Table 4 showed that appraisers' skills required improvement because values of percentage study variations were higher than 10%, which is the general acceptance level for measurement system variation [21]. However, the R-chart, as the appraiser graph in Figure 5, showed that variations in cycle time measures were different among the three appraisers but remained stable for AI. Therefore, large variations in the measurement system resulted from human appraisers, not AI. Appraisers AP1 and AP3 showed less skill compared to appraiser AP2 since their variations of observed time were large and higher than the control limits in the R chart. This type of human error can cause misleading information in the motion and time study. By contrast, the AI algorithm returned the same value of cycle time for all replications. Thus, the cycle time recorded by AI had less variation compared to the human appraisers. This characteristic suggested that AI minimized human error in cycle time observations, and reduced study time in a process improvement project.

Table 3 ANOVA table showing interaction of the Gage R&R study (α to remove interaction term = 0.05)

Source	**DF**	**SS**	**MS**	**F**	**p-value**
Cycle No.	9	1927.56	214.174	79.703	0.000
Appraiser	3	9.97	3.323	1.237	*0.316*
Cycle No. * Appraiser	27	72.55	2.687	766.275	0.000
Repeatability	160	0.56	0.004		
Total	199	2010.65			

Table 4 Variance components and Gage evaluation results

Source	**Variance components**	***%Contribution of variance components***	**Standard deviation (SD)**	**Study variation (6 × SD)**	***%Study variation (%SV)***
Total Gage R&R	0.5529	4.97	0.74360	4.4616	*22.29*
Repeatability	0.0035	0.03	0.05922	0.3553	1.78
Reproducibility	0.5494	4.94	0.74124	4.4475	22.22
Appraiser	0.0127	0.11	0.11276	0.6766	3.38
Appraiser*Cycle No.	0.5367	4.82	0.73262	4.3957	21.96
Part-to-part	10.5743	*95.03*	3.25182	19.5109	97.48
Total variation	11.1273	100.00	3.33576	20.0145	100.00

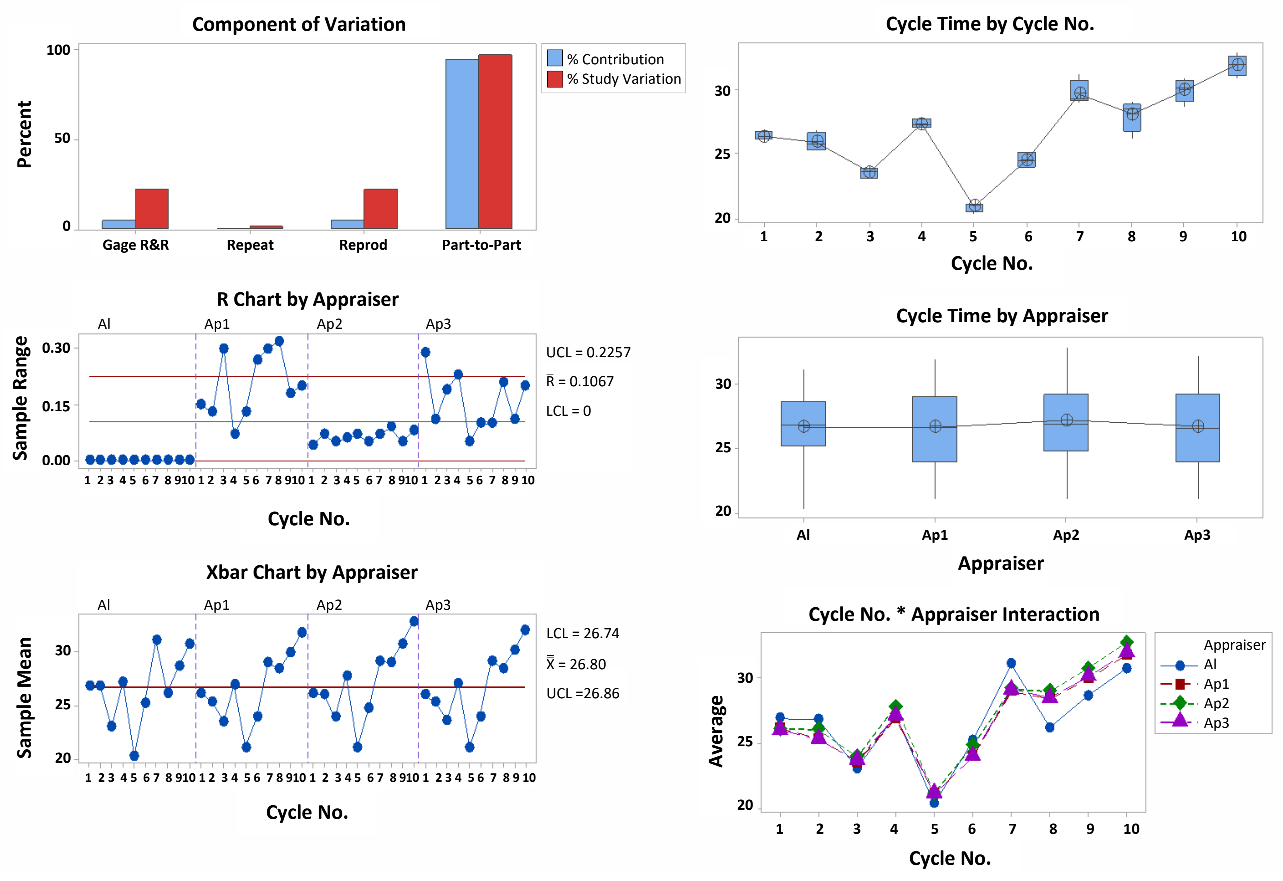


Figure 5 Gage R&R analysis results

5. Conclusions

Research results proved that using Python and OpenCV can help engineers to analyze cycle times of work processes using video recordings. The cycle time detected by AI was similar to the time observed by the engineers but had a lower variation. Thus, using AI reduced human error and also the labor cost of a process improvement project. The proposed algorithm demonstrated a convenient and affordable solution to bridge the gap between AI-Machine vision technology and industrial work improvement schemes. HSV colors used as motion-tracking mechanisms in a scene must be chosen carefully to prevent errors. The current version of this algorithm is coded in Python and the user interface should be developed to reach more groups of users. Further studies should focus on adding features to the algorithm to analyze work motions in poor visibility environments, such as when the color of the hands or parts of the body to be tracked are similar to the background color in the scene. Analysis of standard time considering working speed, fatigue rating and work environment should also be added to enhance the ability of the program.

6. Acknowledgements

This paper is supported by Faculty of Engineering, Thammasat School of Engineering, Thammasat University.

7. References

[1] Meyers FE. Motion, and time study: improving work methods and management. Upper Saddle River: Prentice Hall; 1992.
[2] Feld WM. Lean manufacturing: tools, techniques, and how to use them. Florida: CRC Press; 2000.
[3] Mcframe-Asia.Com. mcframe MOTION software [Internet]. 2022 [cited 2022 Oct 11]. Available from: https://www.mcframe-asia.com/mcframe-motion/.
[4] Gaorfid.Com. People tracking system for manufacturing facilities [Internet]. 2022 [cited 2022 Oct 11]. Available from: https://gaorfid.com/people-tracking-for-manufacturing-facilities/.
[5] Bradski G. The OpenCV library. Dr. Dobb's Journal: Software Tools for the Professional Programmer. 2000;25(11):120-3.
[6] Bradski G, Kaehler A. Learning OpenCV: computer vision with the OpenCV library. California: O'Reilly Media; 2008.
[7] Brahmbhatt S. Practical OpenCV. California: Apress Berkeley; 2013.
[8] Howse J. OpenCV computer vision with python. Birmingham: Packt Publishing; 2013.
[9] Burden J, Cleland M, Conway M, Falconer M, Green R, Geoff Chase JJ, et al. Tracking a single cyclist during a team changeover on a velodrome track with Python and OpenCV. Procedia Eng. 2010;2(2):2931-5.
[10] Parveen S, Shah J. A motion detection system in python and OpenCV. Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks; 2021 Feb 4-6; Tirunelveli, India. India: IEEE; 2021. p. 1378-82.
[11] Yedulapuram S, Arabelli R, Mahender K, Sidhardha C. Automatic door lock system by face recognition. IOP Conf Ser: Mater Sci Eng. 2020;981:032036.

[12] Azhari Halim MA, Othman MFI, Abidin AZZ, Hamid E, Harum N, Shah WM. Face recognition-based door locking system with two-factor authentication using OpenCV. Proceedings of the 6th International Conference on Informatics and Computing; 2021 Nov 3-4; Jakarta, Indonesia. Indonesia: IEEE; 2021. p. 1-7.
[13] Ahamad AH, Zaini N, Latip MFA. Person detection for social distancing and safety violation alert based on segmented ROI. Proceedings of the 10th IEEE International Conference on Control System, Computing, and Engineering; 2020 Aug 21-22; Penang, Malaysia. Malaysia: IEEE; 2020. p. 113-8.
[14] Verma S, Jain PK. COVID-19: Automatic social distancing rule violation detection using PP-Yolo & Tensorflow in OpenCV. Proceedings of the 2022 International Conference for Advancement in Technology; 2022 Jan 21-23; Goa, India. India: IEEE; 2022. p. 1-6.
[15] Terreran M, Lamon E, Michieletto S, Pagello E. Low-cost scalable people tracking system for human-robot collaboration in an industrial environment. Procedia Manuf. 2020;51:116-24.
[16] Geiselhart F, Otto M, Rukzio E. On the use of multi-depth-camera-based motion tracking systems in production planning environments. Procedia CIRP. 2016;41:759-64.
[17] Montgomery DC. Introduction to statistical quality control. 8th ed. New Jersey: John Wiley & Sons; 2020.
[18] Durivage, MA. Practical attribute and variable measurement systems analysis (MSA): a guide for conducting Gage R&R studies and test method validations. Wisconsin: ASQ Quality Press; 2015.
[19] Kappele WD, Raffaldi JD. An introduction to gage R&R: Gage R&R studies can help operators from making costly measurement errors. Quality magazine [Internet]. 2005 [cited 2022 Oct 11]. Available from: https://www.qualitymag.com/articles/83529-quality-101-an-introduction-to-gage-r-r?.
[20] Minitab.Com. Minitab: Data analysis, statistical & process improvement tools [Internet]. 2022 [cited 2022 Oct 11]. Available from: https://www.minitab.com.
[21] Automotive Industry Action Group (AIAG). Measurement Systems Analysis (MSA). 4th ed. Miami: AIAG; 2010.