BAREKENG: Journal of Mathematics and Its Applications
December 2024 Volume 18 Issue 4 Page 2703–2712
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss4pp2703-2712

SIMILARITY CHECKING OF CCTV IMAGES USING PEARSON CORRELATION: IMPLEMENTATION WITH PYTHON

Angga Dwi Mulyanto[1], Bambang Widjanarko Otok[2*], Hasri Wiji Aqsari[3], Sri Harini[4], Cindy Cahyaning Astuti[5]

[1,4] Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Jl Gajayana 50 Malang, 65144, Indonesia

[1,2,3]Departement of Statistics, Faculty of Science and Data Analytics, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo Surabaya, 60111, Indonesia

[5]Information Technology Education Study Program, Faculty of Psychology and Education, Universitas Muhammadiyah Sidoarjo, Jl Majapahit 666B Sidoarjo, 61215, Indonesia

*Corresponding author's e-mail: *prof.bw.otok@gmail.com*

ABSTRACT

Video surveillance technology, such as CCTV, is increasingly common in various applications, including public safety and business surveillance. Analyzing and comparing images from CCTV systems is essential for ensuring safety and security. This research implements the Pearson Correlation method in Python to measure the similarity of CCTV images. Pearson Correlation, which assesses the linear relationship between two variables, is employed to compare the pixel values of two images, resulting in a coefficient that indicates the degree of similarity. We used a quantitative approach with experiments on two scenarios to test the program's effectiveness in measuring image similarity. The results demonstrate that Pearson Correlation is highly effective in distinguishing between identical and other images, providing a more accurate and comprehensive assessment of image similarity compared to histogram analysis. However, the findings are constrained by the specific scenarios and dataset utilized. Further research with more diverse empirical data is required to generalize these results across a broader range of CCTV conditions.

Article History:

Received: 2nd, June 2024
Revised: 4th, August 2024
Accepted: 2nd,September 2024
Published:14th, October 2024

Keywords:

Pearson Correlation;
Image Similarity;
Python.

How to cite this article:

A. D. Mulyanto, B. W. Otok, H. W. Aqsari, S. Harini, and C. C. Astuti., "SIMILARITY CHECKING OF CCTV IMAGES USING PEARSON CORRELATION: IMPLEMENTATION WITH PYTHON," *BAREKENG: J. Math. & App.*, vol. 18, iss. 4, pp. 2703-2712, December, 2024.

1. INTRODUCTION

The use of video surveillance technologies such as CCTV [1] has become increasingly common in a variety of applications, ranging from public safety [2] to business surveillance[3]. Analyzing and comparing images generated by CCTV systems is essential in ensuring safety and security [4]. One method that can be used to measure image similarity is the Pearson correlation. Pearson correlation measures the linear relationship between two variables, which can be applied to image data to identify how similar two images are.

Pearson correlation, first introduced by Karl Pearson, is used extensively in statistical analysis to measure the strength and direction of a linear relationship between two variables [5], [6]. In the context of image analysis, Pearson correlation can be used to compare the pixel values of two images, resulting in a coefficient that indicates the degree of similarity between the images. This method is beneficial in CCTV applications, where identification and identity verification from video footage is crucial, especially in cases of forensics or criminal investigations [7], [8], [9], [10].

Previous research has shown that Pearson correlation can be used to identify similarities between different types of images, including medical images such as CT scans and MRIs. In a study by [11], a significant relationship was found between features extracted from CT scans and MRI images, with the correlation coefficient reaching 0.93 for texture features, demonstrating the potential application of Pearson correlation in broader image analysis. The use of Pearson correlation in the context of CCTV has been tested in various studies that demonstrate its effectiveness in improving face identification accuracy from low-quality footage. For example, research has shown that a gradual decrease in image resolution affects face identification accuracy. Still, using averaged images from multiple low-quality images can improve identification performance by humans and computer systems. Furthermore, research by [12] found that providing numerous photos of the search target consistently improved performance in searching for specific individuals in CCTV footage. This study confirms that variability in the appearance of individuals in images helps improve search efficiency in the context of CCTV surveillance. In addition, another study by [13] showed that using averaged images consisting of multiple poor-quality images can significantly improve face identification accuracy in forensic settings. This finding has important implications for forensic settings where faces are identified from low-quality images such as CCTV footage.

Using Pearson correlation to compare images, we can quantitatively measure the similarity between images, which is useful in various practical applications. For example, security surveillance needs to compare images from CCTV footage with photos from a database to identify specific individuals. Implementing the Pearson correlation algorithm in Python also allows for broader applicability and automation in image analysis. Implementing this algorithm in Python can be done quickly using libraries such as OpenCV [14] for image processing and SciPy [15] or statistical calculations. The steps in this implementation involve reading and converting images into arrays, normalizing pixel values, and calculating the Pearson correlation coefficient between two images. This process improves efficiency and accuracy in image-based identity recognition and verification.

In this research, we aim to implement the Pearson correlation method in Python to measure the similarity of CCTV images. By doing this, we hope to provide an effective and efficient solution for security surveillance applications that require fast and accurate image analysis and verification. This implementation is expected to significantly contribute to the security surveillance and forensics field, as well as open up opportunities for further research in image analysis using other statistical techniques. This research will further examine the effectiveness of the Pearson correlation method in a broader context and explore the potential for developing more sophisticated algorithms to improve accuracy and efficiency in image analysis. Thus, the results of this research can provide a solid foundation for further applications in various fields that require accurate and efficient image analysis.

2. RESEARCH METHODS

This research uses a quantitative approach with experiments to develop and test the functionality of a Python program capable of analyzing the similarity between CCTV-captured images with correlation analysis. We selected several experimental scenarios to ensure systematic testing of the effectiveness of the

Pearson correlation method in measuring image similarity and evaluate the program's ability to process and analyze image data..

The primary data source in this research is a collection of CCTV images stored in a digital folder. The images were obtained from CCTV surveillance systems installed in the researcher's house (the living room and the front). The CCTV camera used is the Xiaomi Smart Camera C300, with the resulting image measuring 2304 x 1296. The data collection technique collects the image files and stores them in a particular directory, which the developed Python program will then access.

Data analysis was conducted through several steps. First, the CCTV-captured images are converted into numerical data using pixel intensity representation [16], [17]. There are many ways to convert numerical data, but this study will first convert the image into grayscale form. Converting an image to grayscale before the conversion process to numeric form is an essential step in image analysis for several reasons. First, it reduces the dimensionality of the data by summarizing information from three color channels (red, green, blue, or RGB) into a single intensity channel, making analysis more straightforward and efficient [18]. Secondly, in many applications, what matters more is the pixel intensity or light-dark value rather than the color information, and grayscale captures this intensity more clearly. It also ensures consistency in analysis, where each pixel has only one value, which facilitates direct comparison between images. In addition, reducing data channels from three to one reduces the computational burden, making the analysis process faster and resource-efficient. Therefore, conversion to grayscale before analysis is a wise move to simplify the process and focus on the most relevant aspects of the image [19]. Each pixel is converted into numerical data after the image is converted to grayscale. For example, if the image is 1980 x 1080 pixels, then the converted variable's amount of data (n) is 2138400. The value ranges from 0 to 255, where 0 represents black and 255 represents white.

This conversion allows the images to be quantitatively analyzed using Pearson correlation [20]. Once the image data has been converted into numerical form, the Python program developed then calculates the Pearson correlation coefficient between pairs of images to measure their degree of similarity. The Pearson Correlation formula is written as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

Where:

- r the Pearson correlation coefficient,
- n the amount of data,
- $x_i$is the value at the i-th data point of the first variable,
- y_i is the value at the i-th data point of the second variable,
- $\bar{x}$ is the average value of the first variable,
- $\bar{y}$ is the average value of the second variable.

The Pearson correlation coefficient (r) measures the strength and direction of the linear relationship between two variables. The value of r can range from -1 to 1 [21], where:

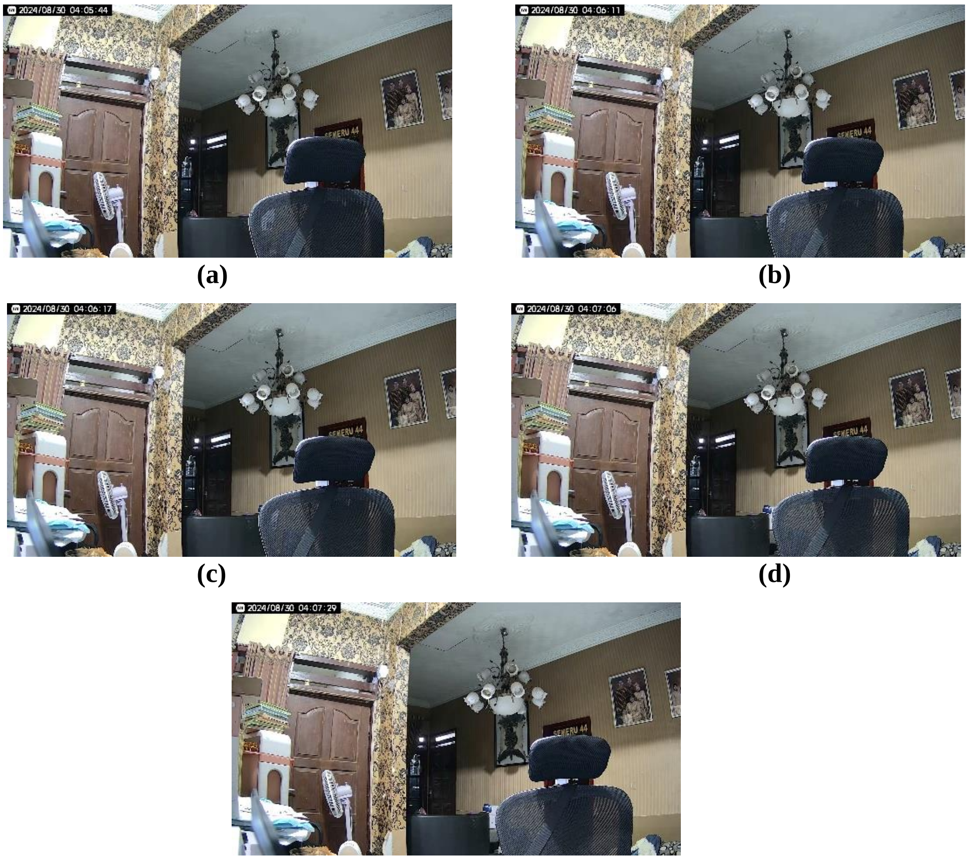
- Values close to 1 indicate a strong positive linear relationship between two variables. That is, as one variable's value increases, the other variable's value also increases.
- A value close to -1 indicates a strong negative linear relationship between two variables. This means that as one variable's value rises, the other variable's value decreases.
- Values close to 0 indicate no strong linear relationship between two variables.

In the context of images and Pearson correlation, each image after being flattened can be considered one "variable," with each flattened pixel being one "observation" or data point in that variable. A Pearson correlation is then calculated between the pair of images to assess how similar (or different) the pixel intensity distributions are between the two images. A correlation value close to +1 indicates high similarity between the two images, while a value close to 0 indicates no linear relationship. This research method is expected to generate a deep understanding of the application of Pearson correlation in CCTV image similarity analysis and develop an effective tool to assist in CCTV image-based security surveillance systems.

3. RESULTS AND DISCUSSION

3.1 Scenario 1: Static Observation in Workspace

In the first scenario, we collected CCTV images from the living room. These images were taken with the same lighting conditions and viewing angles to ensure that the differences between the pictures were minimal, the only slight difference being the chair being shifted slightly. The purpose of this scenario is to measure the level of similarity between relatively similar images. The center room CCTV images can be seen in **Figure 1**.



(a) **(b)** **(c)** **(d)** **(e)**

Figure 1. Living room
(a) 04:05:44 (b) 04:06:11 (c) 04:06:17 (d) 04:07:06 (e) 04:07:29

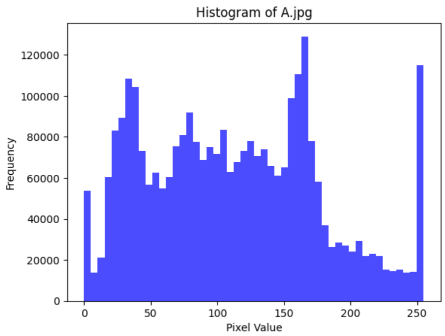
The five images in **Figure 1** presented show the same viewpoint of a room that appears to be a workspace or study with classic decor. In each image, the camera position, viewpoint, and main elements of the room do not change significantly, but some minor differences can be observed as time passes. The results of converting images into numbers can be seen in **Table 1**.

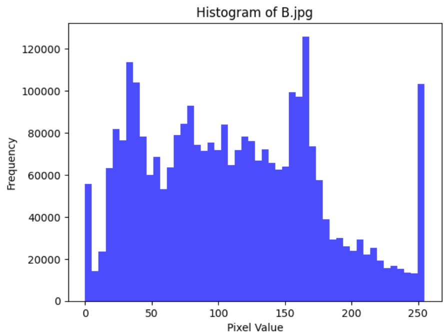
Table 1. Convert Images to Numbers

No	A.jpg	B.jpg	C.jpg	D.jpg	E.jpg
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

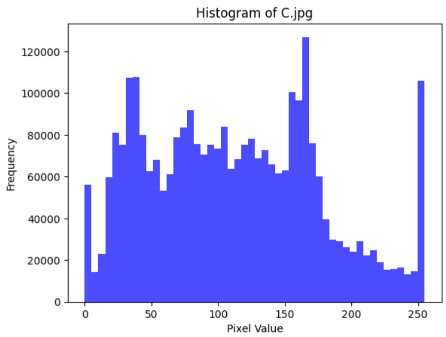
No	A.jpg	B.jpg	C.jpg	D.jpg	E.jpg
...	...	...	...	...	...
2985979	157	158	157	168	160
2985980	159	158	157	168	161
2985981	160	158	157	162	162
2985982	160	159	158	155	162
2985983	161	159	160	151	163

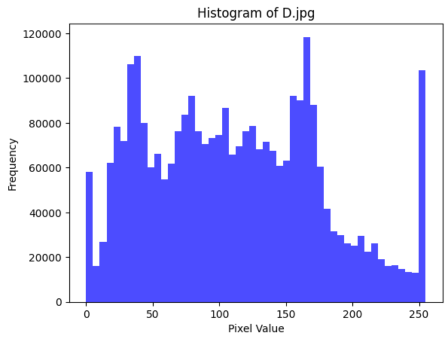
The results in **Table 1** if packaged in the form of a histogram will produce the histogram in **Figure 2**.



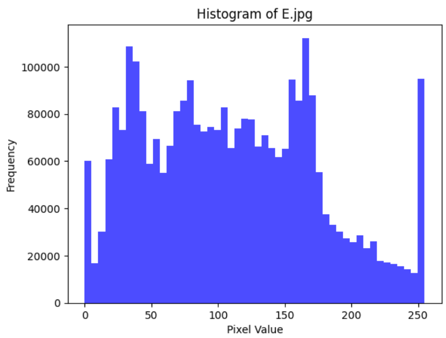


(a) **(b)**





(c) **(d)**



(e)

Figure 2. Histogram of living room images

The histogram results of the five images in **Figure 2** show the distribution of pixel intensities in grayscale form, providing a more detailed picture of the visual characteristics of each image. These histograms generally show a uniform distribution with peaks at certain pixel intensities, reflecting the lighting and contrast conditions in the photos. In such histograms, there are minor variations from one image to another, which may result from changes in the intensity of natural or artificial light in the room or even small shifts in the position of objects that affect the light distribution in the image. The peaks of the distribution tend to be in the center of the spectrum, indicating that the majority of pixels in the image are of medium intensity, without too many very bright or dark areas. In addition, the consistent shape of the histogram also indicates that the photos were taken under relatively stable lighting conditions, with little change that might be caused by slight movements of the object or minor changes in the light source. Overall, analysis of these histograms shows that the images have similar visual characteristics, with the distribution of pixel intensities showing no drastic changes from one image to another.

After running the program for scenario 1, the correlation results of the five images in Figure 1 are presented in Table 2.

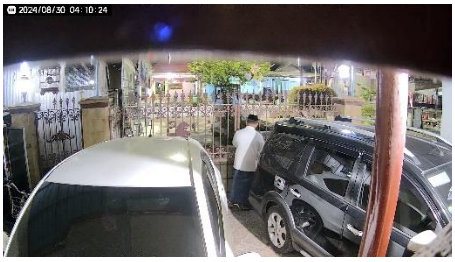
Table 2. Pearson Correlation Matrix in Scenario 1

Image	a	b	c	d	e
a	1.000	0.934	0.963	0.947	0.947
b	0.934	1.000	0.924	0.959	0.960
c	0.963	0.924	1.000	0.937	0.937
d	0.947	0.959	0.937	1.000	0.995
e	0.947	0.960	0.937	0.995	1.000

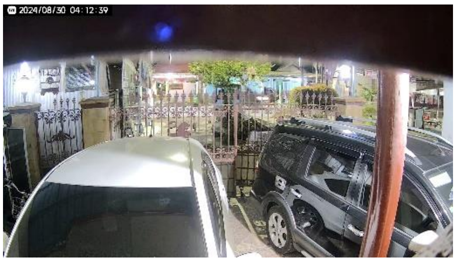
The correlation matrix results in Table 2 show the degree of similarity between the five images represented by the labels a, b, c, d, and e. The correlation between each pair of images ranges from 0.924 to 0.995, indicating that these images have a very high degree of similarity. Starting from Figure 1a, we see that this image has the highest correlation with Figure 1c (0.963), indicating that pixel distribution is almost identical. Figure 1a also has a reasonably high correlation with Figures 1d and Figure 1e (0.947), as well as a slightly lower one with Figure 1b (0.934). This suggests that Figures 1a and Figure 1c may have the most similar visual content, while Figures 1a and Figure 1b, while still very similar, show slightly more differences. Furthermore, Figure 1b has the highest correlation with Figures 1e (0.960) and Figure 1d (0.959), indicating that Figure 1b is more similar to Figure 1e and Figure 1d compared to Figure 1c (0.924). Figure 1d and Figure 1e, show higher similarity than other pairs, for example, between Figure 1d and Figure 1e, show a higher degree of similarity compared to other pairs, for example between Figure 1b and Figure 1c. From the perspective of similarity, these images were taken under almost the same conditions, with a few minor variations occurring due to insignificant changes in object movement.

3.2 Scenario 2: Monitoring Gate Activity in the Morning

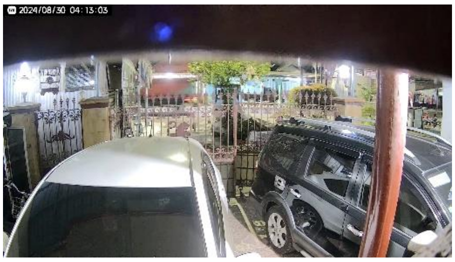
In the second scenario, as seen in Figure 3, the five images taken in front of the house show a similar point of view, focusing on the area in front of the house that includes the iron fence, two parked cars, and a small portion of the street and neighborhood. These images show a moment that occurred in the early morning hours, as indicated by the time on the image, around 04:16. To the naked eye, these images show activity around the house's gate. Figure 3a shows a man seen from behind moving around the gate area. In Figure 3d, another man can also be seen near the entrance, while in Figure 3e, the figure appears to be closer to the CCTV. While Figure 3c shows a vehicle passing by on the road, Figure 3b does not show any such vehicle. The neighborhood around the house looks quiet, with the classically ornate iron fence and the small tree near the gate as prominent visual elements. The cars parked in front of the house also appear consistent in each image, creating the impression that the shooting location has not changed. These images capture a quiet morning with minimal human activity around the house area.



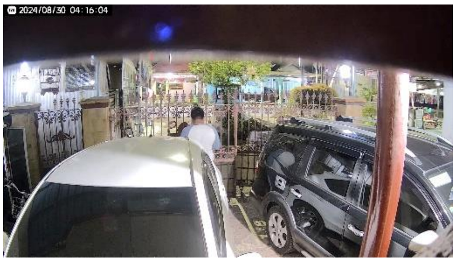
(a)



(b)



(c)



(d)

(e)

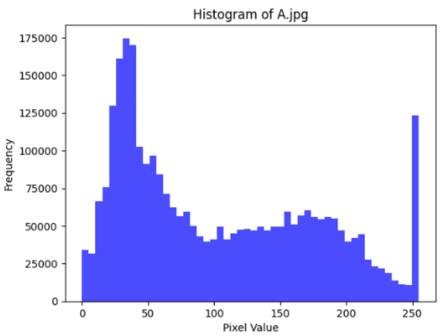
Figure 3. Mushola
(a) 04:10:24 (b) 04:12:39 (c) 04:13:03 (d) 04:16:04 (e) 04:16:36

Like before, if **Figure 3** is converted into numbers, it will produce data as in **Table 3**.

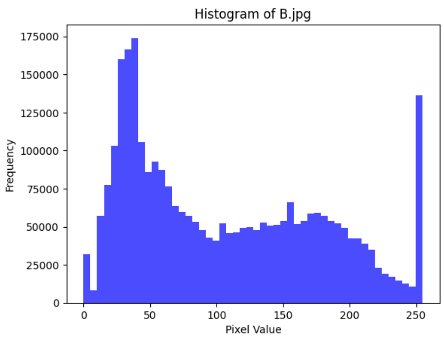
Table 3. Pearson Correlation Matrix in Scenario 1

No	A.jpg	B.jpg	C.jpg	D.jpg	E.jpg
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
...	...	...	...	...	...
2985979	177	175	176	178	138
2985980	176	174	176	178	130
2985981	176	175	176	178	128
2985982	176	175	176	178	132
2985983	176	175	176	178	135

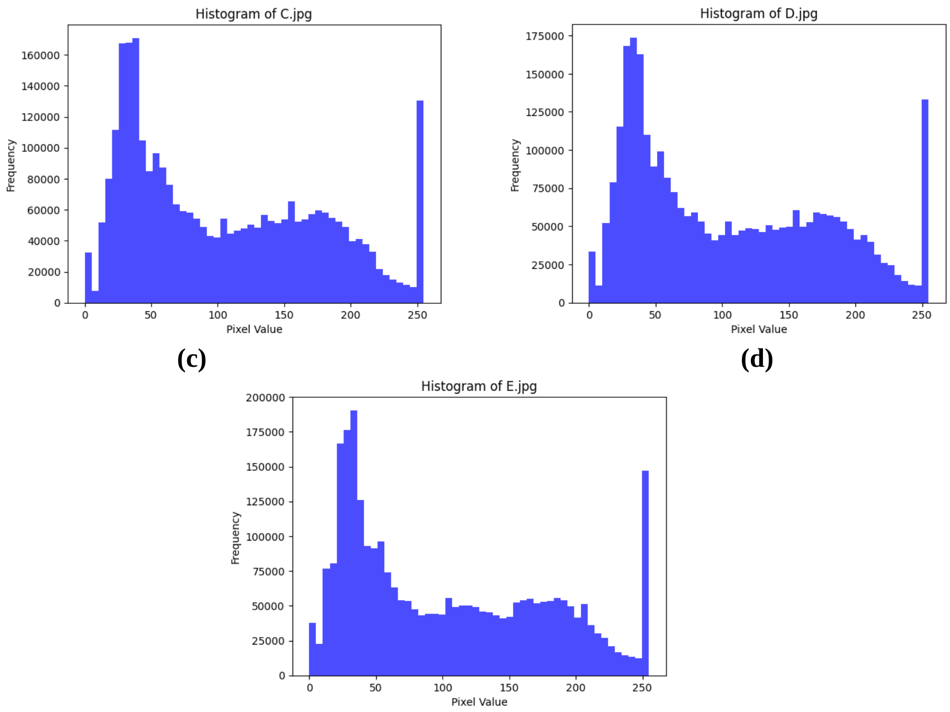
If the data in **Table 3** is made into a histogram, it will produce a histogram that can be seen in **Figure 4**.



(a)



(b)



(c) (d) (e)

Figure 4. Histogram of living room images

The histogram results in **Figure 4** derived from **Figure 3** taken in front of the house, give an idea of the intensity distribution of the pixels in each image. These histograms generally show a uniform distribution, with most pixels in the middle-intensity range. This reflects the fairly consistent lighting conditions in the neighborhood when we took the images due to the illumination from the lights around the house area in the early morning hours. Some minor differences can be observed between the histograms of the images. Images that show the male figure closer to the light source tend to have slight peaks at higher intensities, reflecting the brighter areas in the image. In contrast, images with the male figure farther away from the light source or when the gate is open may show a slight dip in the high intensities due to the light distribution becoming more even or the presence of additional shadows. In addition, these subtle variations in pixel intensity distribution may also indicate slight differences in camera position, changes in viewing angle, or shifts in objects within the frame. Overall, however, these histograms show that the images were taken under similar lighting conditions, with the pixel intensity distribution showing no significant changes from one image to another. This indicates that the photos are highly comparable regarding exposure and contrast, with variations occurring mainly in the mid-intensity area.

After running the program for scenario 2, the correlation results of the five images in **Figure 3** are presented in **Table 4**.

Table 4. Pearson Correlation Matrix in Scenario 2

Image	a	b	c	d	e
a	1.000	0.909	0.918	0.916	0.889
b	0.909	1.000	0.971	0.980	0.954
c	0.918	0.971	1.000	0.977	0.949
d	0.916	0.980	0.977	1.000	0.962
e	0.889	0.954	0.949	0.962	1.000

The Pearson correlation results of the five new images taken in front of the house showed a very high degree of similarity between the images, with correlation values ranging from 0.892 to 0.982. This high correlation indicates that the images have similar pixel distributions, which means that the visuals of these images are almost identical to each other. Images B and C have the highest correlation value (0.982), indicating that pixel distribution is nearly similar. This suggests that only a slight change has occurred between these two images regarding lighting, object position, and camera viewpoint. In contrast, images A and E show the lowest correlation value (0.892). While this value is still very high, it indicates a difference between the two images. This is because image E changes due to the presence of a male figure positioned close to the CCTV. The slightly lower correlation compared to the other image pairs indicates that minor differences can be observed, although they are insignificant. Overall, these Pearson correlation results confirm that the five images are very similar, with only slight variations between them. This indicates that these images were taken under almost the same conditions, with some minor differences that may be due to the movement of objects or the presence of new objects. This high correlation also indicates that the images can be considered very similar in the context of visual analysis.

3.3 Discussion

Regarding a more effective method for assessing image similarity, looking at the results, Pearson correlation is considered superior to histograms for several vital reasons. Pearson correlation measures the strength of the linear relationship between two data sets, thus providing a global picture of the similarity between two images. This means that the correlation considers the entire distribution of pixels within the image, allowing the detection of small changes or shifts in the image that may not be visible through histogram analysis. Meanwhile, a histogram only gives the frequency distribution of pixel intensities without considering the spatial position of the pixels. This makes the histogram less sensitive to minor differences scattered throughout the image. By considering the linear relationship between each pixel in two images, Pearson correlation provides a more accurate and comprehensive measure of how similar two images are. Therefore, in image analysis requiring detailed similarity assessment and sensitivity to slight variations, Pearson correlation is considered a more robust and informative than histogram.

4. CONCLUSIONS

In conclusion, Pearson correlation offers a superior method to histograms in assessing image similarity. Pearson correlation provides a comprehensive and accurate measurement by considering the linear relationship between all pixels in two images, thus being able to detect slight differences that may not be detected through histograms. While histograms only provide the frequency distribution of pixel intensities regardless of their spatial position, Pearson correlation considers the entire pixel distribution globally, making it more sensitive to slight variations significant in image analysis. Therefore, for analysis purposes that require in-depth and accurate similarity assessment, Pearson correlation is a better choice.

ACKNOWLEDGMENT

Thank you to the Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang, for funding the publication of this article.

REFERENCES

[1] P. Vennam, P. T. C., T. B. M., Y.-G. Kim, and P. B. N., “Attacks and Preventive Measures on Video Surveillance Systems: A Review,” *Applied Sciences*, 2021, doi: 10.3390/APP11125571.

[2] S. S. Priya and R. Minu, “Abnormal Activity Detection Techniques in Intelligent Video Surveillance: A Survey,” *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1608–1613, 2023, doi: 10.1109/ICOEI56765.2023.10125671.

[3] J.-Y. Kim and H. Kim, “A Study on the Application Model of AI Convergence Services Using CCTV Video for the Advancement of Retail Marketing,” *Journal of Digital Convergence*, vol. 19, pp. 197–205, 2021, doi: 10.14400/JDC.2021.19.5.197.

[4] S. Arora, K. Bhatia, and V. Amit, “Storage optimization of video surveillance from CCTV camera,” *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 710–713, 2016, doi: 10.1109/NGCT.2016.7877503.

[5] P. Schober, C. Boer, and L. Schwarte, “Correlation Coefficients: Appropriate Use and Interpretation,” *Anesth Analg*, vol. 126, pp. 1763–1768, 2018, doi: 10.1213/ANE.0000000000002864.

[6] J. D. de Winter, S. Gosling, and J. Potter, “Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data.,” *Psychol Methods*, vol. 21 3, pp. 273–90, 2016, doi: 10.1037/met0000079.

[7] M. F. E. Md. Senan, S. Abdullah, W. M. Kharudin, and N. A. M. Saupi, "CCTV quality assessment for forensics facial recognition analysis," *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*, pp. 649–655, 2017, doi: 10.1109/CONFLUENCE.2017.7943232.

[8] P. Pramkeaw, P. Ngamrungsiri, and M. Ketcham, "CCTV Face Detection Criminals and Tracking System Using Data Analysis Algorithm," *Advances in Intelligent Systems and Computing*, 2017, doi: 10.1007/978-3-319-94703-7_10.

[9] K. Maksymowicz, A. Kuzan, Ł. Szleszkowski, and W. Tunikowski, "Anthropological Comparative Analysis of CCTV Footage in a 3D Virtual Environment," *Applied Sciences*, 2023, doi: 10.3390/app132111879.

[10] J. Liu, Y. Zhang, and Q. Zhao, "Video stabilization algorithm based on Pearson correlation coefficient," *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pp. 289–293, 2019, doi: 10.1109/ICAMechS.2019.8861649.

[11] F. F. Birgani and D. Fatehi, "Measurement of the correlation coefficients between extracted features from CT-scan and MRI images," *Journal of Shahrekord University of Medical Sciences*, vol. 20, pp. 87–99, 2018.

[12] M. Mileva and A. Burton, "Face search in CCTV surveillance," *Cogn Res Princ Implic*, vol. 4, 2019, doi: 10.1186/s41235-019-0193-0.

[13] K. Ritchie, D. White, R. Kramer, E. Noyes, R. Jenkins, and A. Burton, "Enhancing CCTV: Averages improve face identification from poor-quality images," *Appl Cogn Psychol*, 2018, doi: 10.1002/ACP.3449.

[14] R. T. H. Hasan and A. B. Sallow, "Face Detection and Recognition Using OpenCV," *Journal of Soft Computing and Data Mining*, 2021, doi: 10.30880/jscdm.2021.02.02.008.

[15] T. J. Sargent and J. Stachurski, "SciPy," *Learning Scientific Programming with Python*, 2020, doi: 10.1017/9781108778039.009.

[16] R. Gulve *et al.*, "39 000-Subexposures/s Dual-ADC CMOS Image Sensor With Dual-Tap Coded-Exposure Pixels for Single-Shot HDR and 3-D Computational Imaging," *IEEE J Solid-State Circuits*, vol. 58, pp. 3150–3163, 2023, doi: 10.1109/JSSC.2023.3275271.

[17] H. Shan, L. Feng, Y. Zhang, and Z. Zhu, "An Interpretable Pixel Intensity Reconstruction Model for Asynchronous Event Camera," *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 1–4, 2023, doi: 10.1109/AICAS57966.2023.10168635.

[18] Z. N. Khudhair *et al.*, "Color to Grayscale Image Conversion Based on Singular Value Decomposition," *IEEE Access*, vol. 11, pp. 54629–54638, 2023, doi: 10.1109/ACCESS.2023.3279734.

[19] Z. Wu and J. Robinson, "Edge-preserving colour-to-greyscale conversion," *IET Image Process.*, vol. 8, pp. 252–260, 2014, doi: 10.1049/iet-ipr.2013.0348.

[20] J. D. de Winter, S. Gosling, and J. Potter, "Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data.," *Psychol Methods*, vol. 21 3, pp. 273–90, 2016, doi: 10.1037/met0000079.

[21] T. J. Cleophas and A. H. Zwinderman, *Modern Meta-Analysis: Review and Update of Methodologies*, 1st ed. Springer, 2017. doi: https://doi.org/10.1007/978-3-319-55895-0.