Image Comparison for Finding the Lowest Priced Unique Commodities in an Online Retail Store

Miguel Nicolas Alcantara, Stefen Genoa Decena, John Lenard Guevarra
College of Computer Studies
Angeles University Foundation
Angeles City, Philippines
alcantara.miguelnicolas@auf.edu.ph, decena.stefengenoa@auf.edu.ph,
guevarra.johnlenard@auf.edu.ph

Ray Nicolas
College of Computer Studies, Faculty
Angeles University Foundation
Angeles City, Philippines
ray.nicolas@auf.edu.ph

Abstract

The impact of e-commerce and the ever-increasing integration of the online world to the daily lives of citizens are the driving reasons for the creation of this thesis. To help streamline the user's shopping experience, the E-Commerce Cheapest Clusterer (ECCC) achieves this goal by doing its namesake: to group similar products into corresponding clusters and select the most affordable product in each of the clusters. The result is then displayed in an easy-to-browse interface that directly connects to the project's target website, Lazada. The images are compared using the Structural Similarity Index Measure (SSIM). If the value returned by the SSIM function is greater than a certain threshold, the image enters the same cluster as the one it was compared to. To ensure efficiency, this paper also investigates the results of different threshold values. Setting a lower threshold value resulted in the algorithm finishing faster at the cost of several misplaced images in clusters, whereas a higher threshold value gave the program a considerable runtime but was less prone to misidentification. Additionally, the testing also revealed that setting the threshold value too high can hinder the program's ability to cluster similar images even if the difference is slight.

Keywords
E-Commerce, image comparison, SSIM, cheapest selection, online store.

1. Introduction
The rise of the Internet era and the gradual integration of technology into every aspect of the daily lives of people has led to an ever more digital lifestyle for global citizens. The Philippines and its citizens are no stranger to the cyberspace; in fact, by January 2022, there were 76.01 million internet users in the country (Kemp 2022). This is helped by the fact that mobile phones have become ubiquitous in the life of people, no matter the class or social status. In the growing integration of society into the digital world, we come across new developments in different fields that previously were grounded outside of the digital world.

One of the facets of such a development is the advent of e-commerce, a relatively new branch of computer science that subsists on an online level (Blazewicz et al. 2016). According to a report on 2020 by Philippine Daily Inquirer, Lazada enjoys a monthly visit of over 34 million, followed by Shopee at over 19 million monthly visits. Looking at the same source that the report had cited (https://iprice.ph/insights/mapofecommerce/en/), the numbers have shifted even greater with its latest report, with Shopee enjoying 71 million monthly visits and Lazada at 36 million, as of Q2 of 2022. The surprising shift in monthly visits indicate a growing popularity of e-commerce as an aspect of daily life, in the case of Shopee's explosive growth; as well as consistency of the trend, in the case of Lazada's monthly visit numbers.

This great popularity of e-commerce necessitates the attempts to streamline and ensure a better experience for its customers. The most popular form of e-commerce in the Philippines takes the form of a B2C (business-to-consumer) model. The most used e-commerce services in the Philippines, Lazada and Shopee, fall under the B2C category as they act as online intermediaries for online sellers and prospective customers. The platform is open to various marketing tactics by the sellers it accommodates, and the buyers on the platform are free to choose from which seller they wish to acquire their goods from. E-commerce sites also feature flexible payment options and product delivery/retrieval methods, which further helps their popularity because of sheer convenience.

As such, some of the more novel technologies are being explored to bring innovations to this new field. For example, the model article for this study attempted to use the K-Means algorithm on different e-commerce websites in order to determine the cheapest version of the item (Prasetyo 2018). In this regard, data science crossed over with the field of e-commerce to bring valuable results. In a similar vein, the research team wishes to use adjacent technologies in the field of data science with the similar goal of finding the cheapest products, this time with a different scope and methodology for achieving that objective. The researchers are interested in applying the concepts of image processing and clustering to create a software that would be beneficial to prospective customers of the leading e-commerce services in the Philippines.

Because of the convenience and accessibility that e-commerce provides, more people are opting to switch to online shopping. Purchasing items and products online, and filtering prices from low to high are simple to do, but seeing several duplicate products can affect the user shopping experience as duplicate products just fill up the result page and makes it harder for the shopper to find the item they're supposed to find. As a proposed solution, this study worked on eliminating duplicate products and finding the cheapest product as well as the contributing factors of image processing in E-Commerce websites. Furthermore, this paper intends to simplify the shopping experience by making it user-friendly.

1.1 Objectives

The goal of this research is to provide a more efficient and productive way of buying the cheapest product on an e-commerce website and eliminating duplicate products. Image comparison and a clustering algorithm are used to cluster the same or similar images together and the lowest priced item are picked in each cluster. The specific objectives of the research are as follows:

1. To extract datasets from a selected e-commerce platform (Lazada) using Selenium, a tool that is used to automate web browser interaction.
2. To apply the Structured Similarity Index Measure (SSIM) algorithm to cluster together similar looking images
3. To output an interface where the duplicate products have been eliminated by clustering them and selecting the cheapest product

2. Literature Review

Guo et al. (2017) argued that for a recommendation system for e-commerce platforms to be truly of use, the said system should be based on a tailored recommendation algorithm for consumers. It is urged that consumers need to interact with the system in order to bring about relevant and accurate recommendations. The e-commerce platform, according to Rui (2021), is the carrier of cross-border e-commerce, and product classification in the e-commerce platform is critical. This also provides an opportunity to create a new classification technology to address the shortcomings of existing product classification.

A study by Qing, et al. (2020) looks at how to identify objects from ordinary images, a challenging task in real-world applications due to category diversity and other aspects. In a study by Yang et al. (2020), it was stated that it is an apparent behavior for consumers to compare information of the same product across multiple e-commerce platforms. It was found that in e-commerce transactions, images play a significant role in determining the quality of the user experience and the users' decision-making (Chaudhuri 2018). Images provide precise product information that aids the client in developing trust in the product's quality and ability to deliver on its promises.

Metrics for comparing images are used in objectives such as determining image quality (Umme Sara 2019). In the cited study, different metrics were compared; these were the MSE (Mean Square Error), PSNR (Peak Signal to Noise Ratio), and SSIM (Structured Similarity Index Measure). The Mean Square Error (MSE) is one of the most popular image quality assessment metrics because of its mathematical simplicity. It is also considerably fast compared to other more recent image quality metrics; this however comes at the tradeoff that MSE can become quite unreliable for detecting perceived visual quality when images are not clear (Sung-Ho Bae 2020). Despite its drawback, MSE is still popularly used as a benchmark for comparison with other similarity metrics. The formula of Peak Signal to Noise Ratio (PSNR) is also derived from the MSE formula, again touted as easy to comprehend and calculate (Setiadi, 2020). Therefore, both the advantages and drawbacks of the aforementioned metric are also seemingly present in PSNR. It can be sensitive to distortions via Gaussian noise, which are distortions produced by sensor limitations especially in lower lighting conditions. Structured Similarity Index Measure (SSIM) is a metrics proposed by Zhou Wang which became nearly as popular as MSE in comparison because of its effectivity (Zhou Wang 2004). Many of the modern papers cite this article, including the documentation of scikit-image on their implementation of the SSIM function. Furthermore, it also draws comparison in the paper cited for PSNR. Setiadi's article also cites a finding that SSIM, in comparison, can be sensitive to the loss of quality that happens with JPEG format compressions.

Automating tasks using Python is one of the popular features of the programming language. Libraries have been developed as a way to extend the flexibility and power of Python. Widely used automation framework Selenium may be utilized to scrape web pages in a variety of ways (Gudavalli & JayaLakshmi 2022). The study exhibited the possibility of information being able to be scraped from static web pages as well as analyze the major advantages and hurdles of web scraping in building functional web applications. The framework is deemed fully functional and can be used by novice and advanced users alike to automate the testing of web sites.

3. Methods

The researchers programmed a series of steps using Python to build the application that would implement the solution. A keyword is inputted to be searched on Lazada's search function. The resulting webpage is then scraped using Selenium, acquiring each of the products' image, name, price, and their corresponding internet link/address. This step is repeated for the consequent pages in the search result until all results have been scraped, or a certain limit is given. The output from the previous step is stored in a dataframe, without losing their association from each other. The images are extracted and run through an image processing algorithm. The algorithm selects the same/similar looking images and cluster or label them together. After being clustered, the program goes through every cluster and selects the cheapest product from each cluster. The program outputs all of the selected items, displaying their image, name, price, and the web address for the user's perusal. These items are also sorted from lowest to highest in their price.

In order to make sense of the data, the researchers bundle the information into a single row, associating each item with its corresponding name, image, web address, and price. The program also generates a unique ID for each row, which is then used as the file name for the downloaded images to maintain association. Despite the input of the model being a string of text, the process mainly involves the usage of images and as such there is an emphasis on the preparation of images for the model to process. The following figure is a visual illustration of the process.

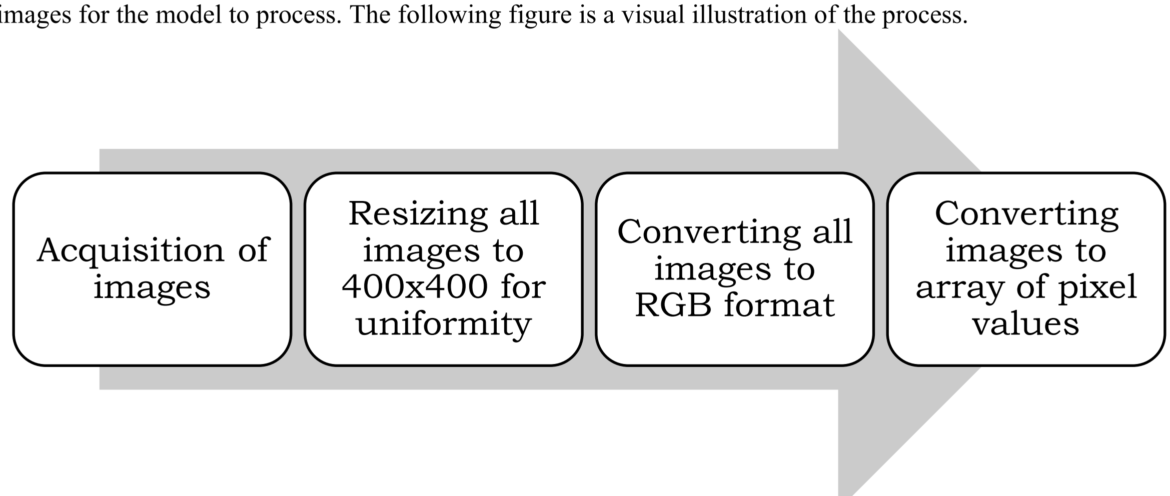

Figure 1. Data Preprocessing Flow

Once the images have been preprocessed as outlined, the building of clusters according to the model can proceed. By default, the program starts with cluster 0 and assigns the first image (based on item ID) as the image to compare to with the rest of the dataset. This will be referred to as the "candidate image" for the cluster. Using SSIM, the candidate image is compared to the next image that does not yet belong to a cluster in the dataset.

For the algorithm, Structured Similarity Index Measure (SSIM) is used. The formula for SSIM is as follows:

$$SSIM(x, y) = l(x, y)^{\alpha} \cdot c(x, y)^{\beta} \cdot s(x, y)^{\gamma}$$

The metric considers three key features of an image: luminance, contrast, and structure. These features are compared using three corresponding comparison functions, as denoted by $l(x, y)^{\alpha}, c(x, y)^{\beta}, and\ s(x, y)^{\gamma}$. While the variables α, β, and γ can be adjusted depending on which feature one wants to give emphasis to, they are often left all equal to 1. This formula then returns a value ranging from -1 to 1, with lower values indicating that the images are more different and higher values indicating similarity. Implementations often normalize the values to range between 0 and 1. It is also worth noting that rather than applying the SSIM globally to the entire image, it is instead applied regionally and the resulting output is then averaged. This can then be called the MSSIM, or the Mean Structural Similarity of two images being compared.

If the SSIM value between the two images reaches a certain threshold, the image being compared to would be assigned to the same cluster as the candidate image, and would no longer be compared to on succeeding clusters. If the SSIM value is below the threshold amount, nothing happens and the program continues comparing with the rest. Once all non-clustered images have been compared with the candidate image, the program creates a new cluster and assigns the next non-clustered image as the new cluster's candidate image. This loop is executed until all images have been assigned their own clusters.

In order to test the actual performance of the prototype, three different search prompts were used, and three pages worth of items (120 items total) were included for each test case. In addition, three thresholds of similarity were used for the test cases so that the researchers would be able to determine what value should be used in detecting similar products. The search prompts were the following items:

- Earphones
- Gaming mouse
- Face mask

The thresholds used were 0.5, 0.7 and 0.9. This means that in the test case for 0.5, the program was more lenient and inclusive in assigning images to clusters whereas a threshold of 0.9 means that the program is stricter and would only cluster images that reach a score of 0.9 or above. The value of 1 was not used as a threshold because, intuitively, this would cause the program to look only for pixel-perfect correlations between images, thus not fulfilling the intended feature of clustering similar (but not same) images.

4. Data Collection

To simulate a search query in Lazada's database, the researchers followed the pattern that the target website uses in its query. In Lazada's case, it follows a certain pattern: *"/catalog/?page={page number}&q={keywords}"*. For example, if one is searching for "male shirts" and has already reached the 3rd page, the following pattern would show up in the web address: *"https://www.lazada.com.ph/catalog/?page=3&q=male+shirts"*, and then followed by additional string which can be omitted from the query. Using this, the program generates a series of search query using the keyword given and the number of pages specified, which will then be automated by the program to access each individual page and scrape the data. Figure 2 demonstrates this pattern.

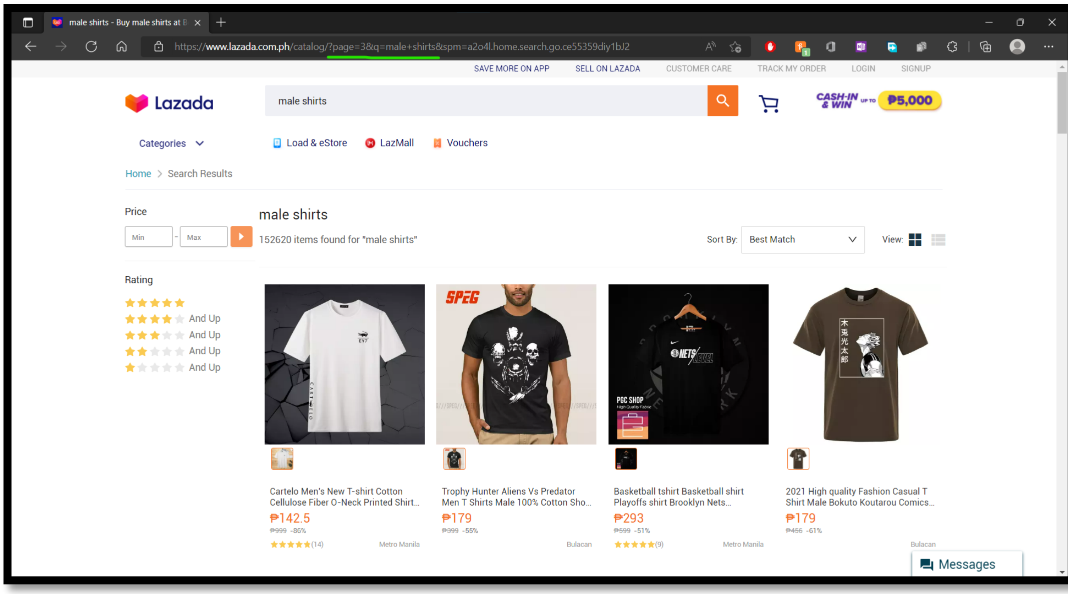

Figure 2. A sample search query

The program then scrolls down in order to load the image, as not doing so would cause the scraper to miss some image links, especially at the bottom of the page. The following data are then taken from this list of items:

- name of the product
- URL to the image/icon of the product
- URL to the product, or its web address
- price of the product

Additionally, the program also accessed each of the URL to the image of the product, downloads the image, and then renames the image as *"{id}.png"* so as to retain the association. Once the data has been collected from all necessary pages, it is then stored in a .csv file, ready for processing.

5. Results and Discussion

To assign a numerical value to the application's effectivity, a simple accuracy formula was followed:

$$Error\ Rate = \frac{Wrongly\ assigned\ items}{Total\ number\ of\ items} \cdot 100$$

The output of this formula is expressed as a percentage, rounded to 2 decimals. Notable exceptions and errors are also pointed out and discussed in the upcoming sections.

Because the dataset was acquired as a set of 3 pages, the total number of items was consistently at 120. The number of wrongly assigned items are identified visually by manually checking each cluster. The wrongly assigned item is distinguished by comparing the outlier from the majority of the items in the cluster. In cases of huge clusters with no clearly defined majority, all of the items in the cluster would be considered wrongly assigned. These cases are prominent in threshold 0.5 test cases. The following figure demonstrates such a case.

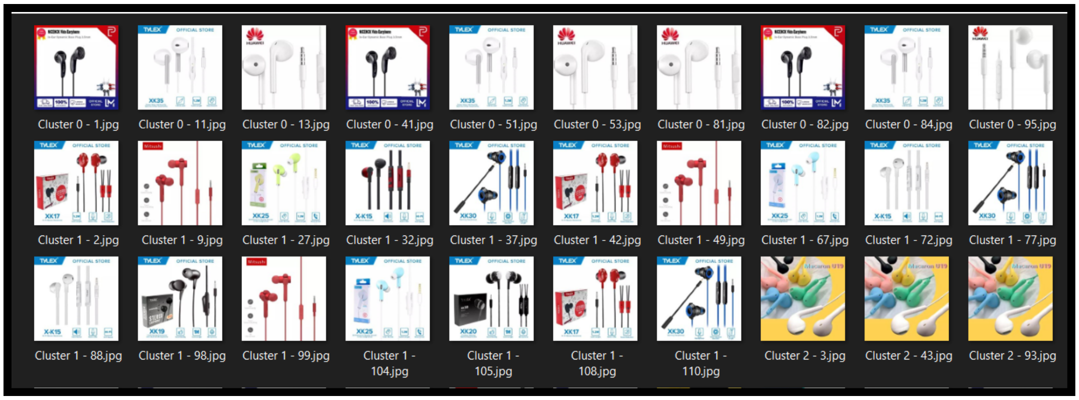

Figure 3. Clusters 0 and 1 demonstrating no clear majority

Conversely, if an item was assigned its own cluster but was very similar to an existing cluster, it would also be counted as a wrongly assigned item. Figure 3 reflects this scenario.

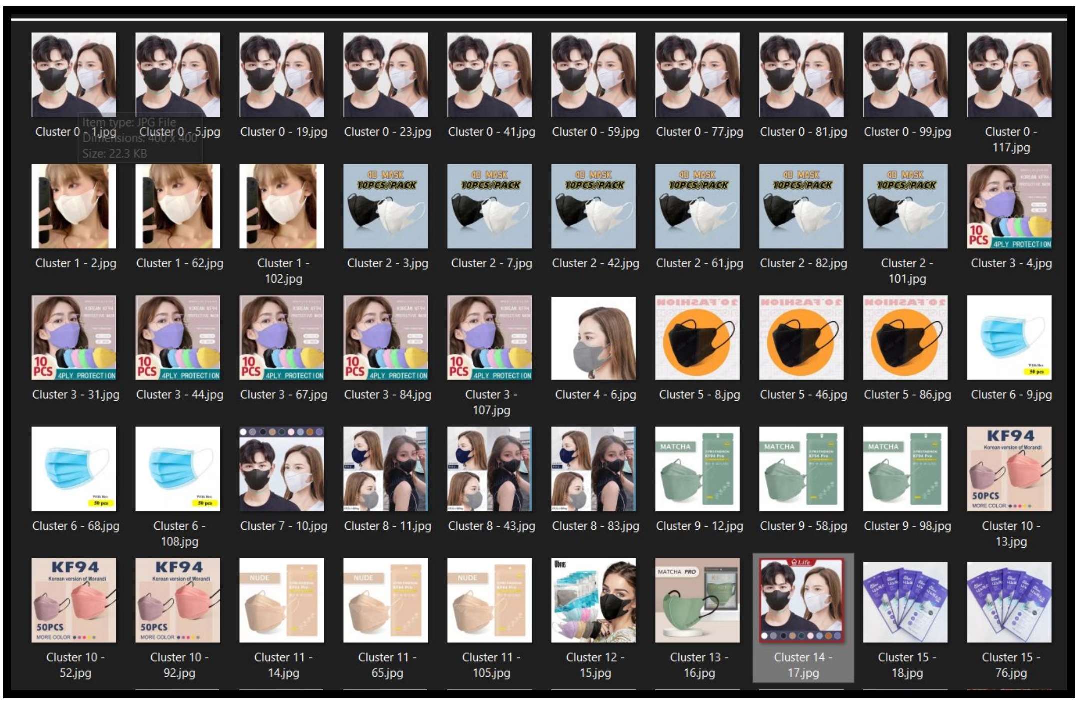

Figure 4. Lone item in cluster 7 and cluster 14 match cluster 0

The dataset for each test cases (earphones, gaming mouse, face mask) were all extracted within the same day: October 2, 2022.

5.1 Numerical Results

For the "Earphones" test case (as well as its corresponding thresholds), the following information describes the result: When set to 0.5 threshold, a total of 28 clusters were formed. Four clusters were found to be erroneous. Of those, three clusters – #0, #1, and #6 possessing 10, 17, and 9 items, respectively – had completely mixed images and thus had to be invalidated completely, while cluster #3 (possessing 31 items) had one defined majority and thus only 20 of its items were invalidated. Table 1 shows the information regarding erroneous clusters, as well as those for 0.7 and 0.9 thresholds. Subsequently, Tables 2 and 3 also describe the same information, but for "Gaming Mouse" and "Face Mask", respectively.

Table 1. Earphones test cases

Threshold: 0.5			
Cluster No.	**Items in Cluster**	**Wrongly Assigned**	**Remarks**
0	10	10	No strongly defined majority, entire cluster invalid
1	17	17	No strongly defined majority, entire cluster invalid
3	31	20	1 strongly defined majority
6	9	9	3 defined majority, entire cluster invalid
Threshold: 0.7			
9	9	9	No strongly defined majority, entire cluster invalid
11	5	1	Outlier is very similar but can be distinguished manually
20	2	0	Same models but different color, refer to cluster 26
26	2	0	Same models but different color, refer to cluster 20
Threshold: 0.9			
20	1	1	Same model but different color, refer to cluster 37
27	2	0	Same model but different color
37	1	1	Same model but different color, refer to cluster 20

Table 2. Gaming mouse test cases

Threshold: 0.5			
Cluster No.	**Items in Cluster**	**Wrongly Assigned**	**Remarks**
0	37	37	No strongly defined majority, entire cluster invalid
1	7	1	Outlier is very different from the majority
2	12	12	3 defined majority, entire cluster invalid
3	5	2	2 outliers different from the majority, and each other
7	5	2	2 outliers different from the majority, and each other
15	3	1	Outlier is very different from the majority
29	3	3	No strongly defined majority, entire cluster invalid
Threshold: 0.7			
2	4	1	Outlier has similar image borders w/ the majority
4	16	16	No strongly defined majority, entire cluster invalid
6	6	3	2 strongly defined majority, should have been 2 clusters
7	6	3	1 strongly defined majority
28	2	2	Two different models
41	3	3	No strongly defined majority, entire cluster invalid
Threshold: 0.9			
N/A	N/A	N/A	No errors observed; all clusters properly assigned and unique

Table 3. Face mask test case

Threshold: 0.5			
Cluster No.	**Items in Cluster**	**Wrongly Assigned**	**Remarks**
2	29	29	No strongly defined majority, entire cluster invalid
6	1	1	Similar images in other clusters
10	1	1	Similar images in other clusters
23	2	2	Similar image model with cluster 28
28	2	2	Similar image model with Cluster 23 but with more masks on the image
Threshold: 0.7			
6	6	6	Two defined majority, one outlier, entire cluster invalid
7	1	1	A lone image that is similar to other clusters
9	3	3	Same image, different "flavors"
13	1	1	Similar image in different clusters
15	3	3	Similar image in different clusters
16	1	1	Similar image in Cluster 14
19	1	1	Similar image model in different background
36	2	2	Similar image model with Cluster 31, but with more masks in the image
Threshold: 0.9			
7	1	1	A lone image that is similar to other clusters
11	3	0	Different mask flavor but same mask with Cluster 9
14	1	1	A lone image that is similar to other clusters
16	3	3	Images are similar to that of Cluster 0
17	1	1	A lone image that is similar to Cluster 15
20	1	1	Similar image model in different background

Table 4 summarizes information gathered from the test cases. Note that the total number of items for all test cases for each threshold is 120.

Table 4. Summary of test cases

Search Query	**Thresholds**	**Number of Clusters**	**Total Wrongly Assigned Items**	**Error Rate**
Earphones	0.5	28	56	46.67%
	0.7	43	10	8.33%
	0.9	48	2	1.67%
Gaming Mouse	0.5	33	58	48.33%
	0.7	47	28	23.33%
	0.9	61	0	0%
Face Mask	0.5	40	35	29.17%
	0.7	49	18	15%
	0.9	53	9	7.5%

5.2 Graphical Results

Figures 4 and 5 visualize the number of clusters and error rate correspondingly, arranged by the items in the search query. The colors denote the threshold value used.

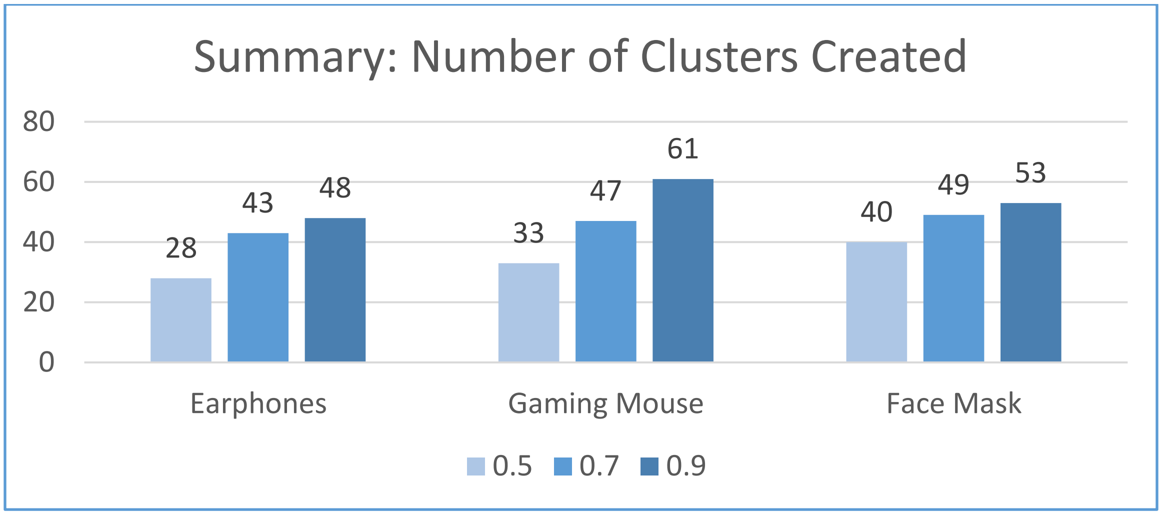

Figure 5. Number of clusters created

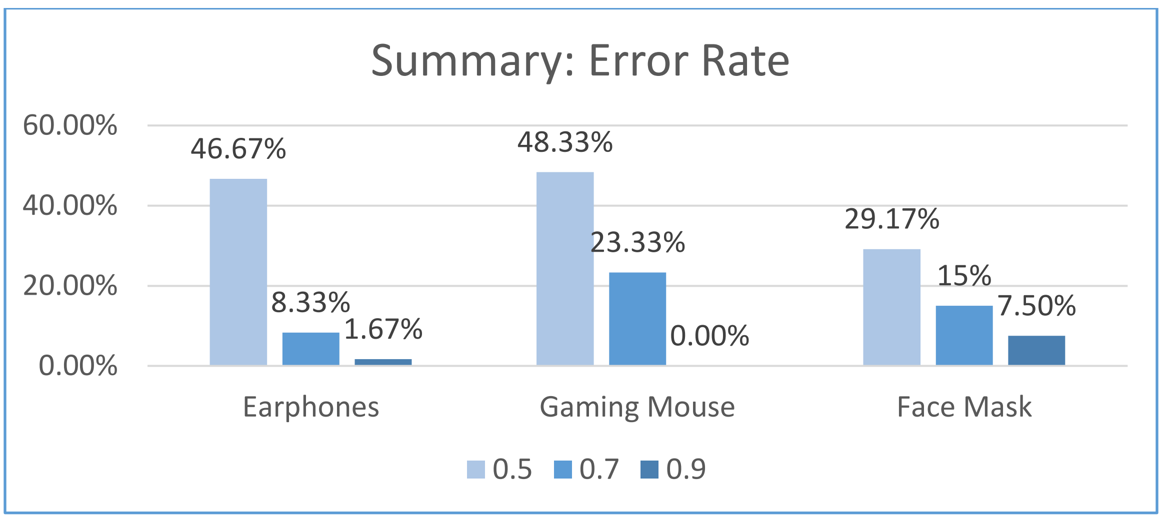

Figure 6. Error rate

It can be noted that a higher number of thresholds improves the results given by the application. This comes at the cost of having more clusters created, translating to a slower overall runtime as the algorithm runs through the dataset more for each cluster.

After having grouped together the images, the application selects the cheapest offering from each of the cluster and the result becomes the selection that is offered to the prospective user. Figure 6 is a screenshot of the expected output of the application.

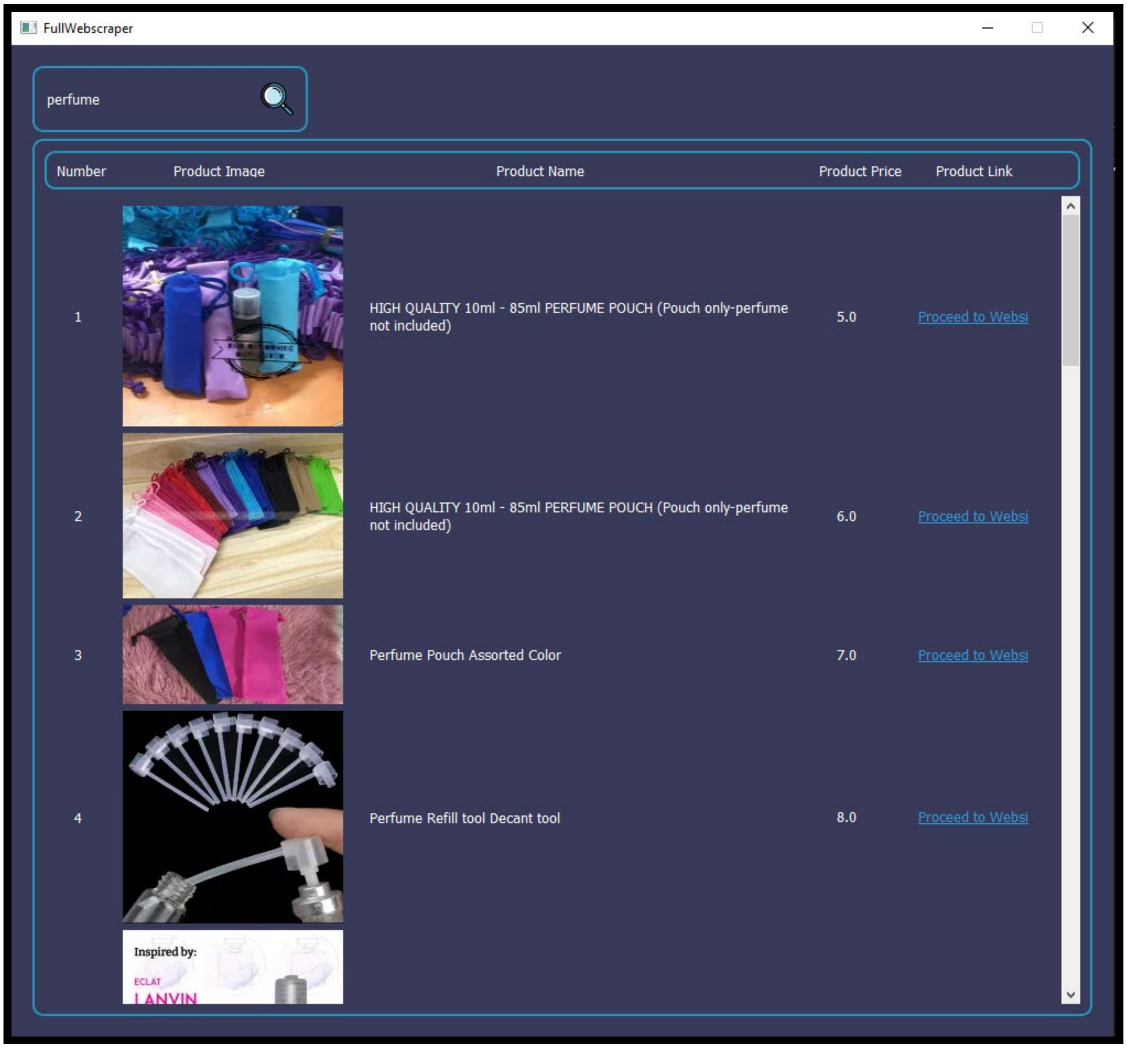

Figure 7. Testing the output with "perfume" as search query

5.3 Proposed Improvements

The system could potentially benefit from using hybrid algorithms, such as using both MSE and SSIM together and then applying different weighted values to both outputs when considering what images fit in a cluster. It is also worth considering to develop a mobile version of the system, as a huge portion of online shopping is done through smartphone applications of the e-commerce websites. For such an undertaking, the future researchers or developers must consider the feasibility of the following scenarios: whether the algorithm running the image comparison can be done on the customer's own phone (client-side), or if it should be deployed in the cloud and the customer must connect to the system that way. Lastly, another avenue to explore is to try achieving the same objectives this paper did with different parameters instead such as a product's title, or their description page, essentially touching on the field of natural language processing.

6. Conclusion

At the beginning of the paper, the problem of convenience regarding online shopping was brought up and has been the central objective of the project. The system was able to accomplish each step as outlined and produces an output within the expectations of the research. In addition, the system was also tested on different parameters and test cases in order to help identify the matters that could be finetuned. As demonstrated in the discussion of results, there was a clear pattern that emerged when setting up the threshold of the image processing algorithm. Because a lower threshold meant that the compared images could be less similar to each other, the system would only generate fewer clusters. While the number of clusters itself was not a problem, it was instead the number of misidentified images within each of the cluster, rendering many of the clusters incoherent and unable to represent a majority. On the other hand, setting a threshold of perfect similarity – meaning 1.0 – meant that the program would only accept perfect comparisons between images. This would run the risk of not clustering images that are actually the same product but have been classified otherwise because of small imperfections in the image file.

With the test cases conducted and observed through different datasets (earphones, gaming mouse, face mask), the numbers suggest that the optimal threshold to set for the system is 0.9. All of these were achieved and written using the programming language Python and external libraries that were designed for Python. As such it can be concluded that the project has answer all of the problem statements and has met all of its original objectives that were stated at the first chapter.

References

Bhardwaj, H., Tomar, P., Sakalle, A., & Sharma, U., *Artificial Intelligence to Solve Pervasive Internet of Things Issues.* Greater Noida: Elsevier, 2021.

Blazewicz, J., Cheriere, N., Dutot, P.-F., Musial, J., & Trystam, D., Novel dual discounting functions for the Internet shopping optimization problem: new algorithms. *Journal of Scheduling*, 245-255, 2016

Chaudhuri, A., *A Smart System for Selection of Optimal Product Images in E-Commerce*. Retrieved from IEEE International Conference on Big Data (Big Data), 2016

Chunmei Zheng, G. H. A Study of Web Information Extraction Technology Based on Beautiful Soup. *Journal of Computers*, 381-387, 2015

Datta, P., All about Structural Similarity Index (SSIM): Theory + Code in PyTorch. Available: https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e, September 3, 2020.

Gudavalli, A., & JayaLakshmi, G., Implementation of Test Automation with Selenium Webdriver. *Grenze International Journal of Engineering & Technology*, 347-353, 2022

Guo, Y., Wang, M., & Li, X., An Interactive Personalized Recommendation System Using the Hybrid Algorithm Model. *symmetry*, 2017

Kemp, S., *DIGITAL 2022: THE PHILIPPINES.* DataReportal. Available: https://datareportal.com/reports/digital-2022-philippines, 2022

Kubat, M., *An Introduction to Machine,* 3rd Edition, Cham: Springer Nature, Switzerland, 2017

Prasetyo, V. R., Searching Cheapest Product on Three Different E-Commerce Using K-Means Algorithm. *2018 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 239-244, 2018

Qing, L., Xiaojiang , P., Liangliang, C., Wenbin, D., Hao, X., Yu, Q., & Qiang, P., Product image recognition with guidance learning and noisy supervision, *Computer Vision and Image Understanding*, vol. 196, 2020

Rui, C., Research on Classification of Cross-Border E-Commerce Products Based on Image Recognition and Deep Learning, *IEEE Access*, vol. 9, pp. 108083-108090, 2021

Setiadi, D. R., PSNR vs SSIM: imperceptibility quality assessment for image steganography. *Multimedia Tools and Applications*, vol.80, pp. 8423–8444, 2020

Sung-Ho Bae, S.-B. P., A Very Fast and Accurate Image Quality Assessment Method based on Mean Squared Error with Difference of Gaussians. *Journal of Imaging Science and Technology*, vol.64, pp. 1-5, 2020

Tayao-Juego, A., Lazada, Shopee, ZALORA top list of most visited online stores in PH*,* April 03, 2020, https://business.inquirer.net/293997/lazada-shopee-zalora-top-list-of-most-visited-online-stores-in-ph, Accessed March 3, 2022

Umme Sara, M. A., Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications*, vol. 7, pp. 8-18, 2019

Yang, Z., Ouyang, T., Fu, X., & Peng, X., A decision-making algorithm for online shopping using deep-learning–based opinion pairs mining and q-rung orthopair fuzzy interaction Heronian mean operators. *International Journal of Intelligent Systems*, vol. 35, pp. 783-825, 2020

Wang, Z., Bovik, A.C., Sheikh, H. R., Simoncelli, E. P., Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions On Image Processing*, vol. 13, pp. 600-612, 2004

Biographies

Stefen Genoa Decena is an undergraduate student from Angeles University Foundation, taking up B.S. in Computer Science under the specialization of Data Science. His research interest covers the field of data science and its potential applications to daily life.

Miguel Nicolas Alcantara is an undergraduate student of Angeles University Foundation, currently undertaking the B.S. in Computer Science program. His main research interest is machine learning.

John Lenard Guevarra is an undergraduate student of Angeles University Foundation under the B.S. in Computer Science program. His research interest includes software usability and user experience.

Ray Nicolas is an instructor at the College of Computer Studies at Angeles University Foundation, Angeles City. He graduated in the year 1995 under B.S. in Computer Science degree from La Consolacion University Philippines (formerly University of Regina Carmeli). He earned his degree in Master of Science in Information Technology at Hannam University, South Korea, in 2003. He has over 25 years of experience teaching in the field of computer science and information technology, as well as handling administrative duties. He is affiliated with the Philippine Society of I.T. Educators – Region III (PSITE R3).