Performance Analysis of Eigenface Method for Detecting Organic and Non-Organic Waste Type

Aditya Wijayanto
Fakultas Informatika, Rekayasa Perangkat Lunak
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia
*aditya.wijayanto@ittelkom-pwt.ac.id

Afifah Dwi Ramadhani
Departemen Teknik Elektro, Teknologi Rekayasa Internet
Politeknik Elektronika Negeri Surabaya
Surabaya, Indonesia
afifah@pens.ac.id

Alon Jala Tirta Segara
Fakultas Informatika, Rekayasa Perangkat Lunak
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia
alon@ittelkom-pwt.ac.id

Muhamad Azrino Gustalika
Fakultas Informatika, Teknik Informatika
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia
azrino@ittelkom-pwt.ac.id

***Abstract*— Indonesia is one of the largest countries in Asia with a very dense population. According to data from The World Bank, human population indicators in Indonesia in 2019 increased by 270 milion people. This shows that population density in Indonesia is related to world problems related to waste generated from households. The household sector contributes as the top waste producer in Indonesia. Landfilling that occurs without any waste sorting, results in waste being more difficult to decompose and difficult to recycle. Therefore, to overcome this problem, it is necessary to increase public awareness about waste sorting and processing. We propose to create a device that can help sort organic and non-organic waste with Computer Vision-based Artificial Intelligence technology using the Eigenface method and the Internet of Things. Eigenface is a method that has a working principle by using XML files in performing face recognition. The result of testing in this system can run well, where the system detects organic objects the door of the chopping machine can open and if it detects nonorganic, the machine door is closed. The accuracy result for organics is 70% and for inorganic 75%. This is due to the lack of variation in the dataset and changes in the physical condition of the object.**

Keywords— Microcontroller; Computer Vision; Eigenface

I. Introduction (*Heading 1*)

Indonesia is one of the most populous countries in the world. Population density is one of the factors that can cause problems in Indonesia. Indonesia's population density is related to global issues related to household waste. The household sector contributes as the largest waste producer in Indonesia. However, the public awareness in solving the problem is low. For solving this problem, we can do a waste separation. Accumulation of waste generated without waste separation makes it difficult to decompose and recycle waste. Moreover, it can cause air pollution.

Therefore, in order to overcome these problems, it is necessary to raise public awareness of waste separation and disposal. We propose the production of equipment that is useful for separating organic waste and non-norganic waste by using artificial intelligence technology based on computer vision using the eigenface method and the Internet of Things. Eigenface is a method that has the working principle of performing face recognition using an XML file. This study uses a Raspberry Pi camera to detect objects. The monitoring data is read by the Raspberry Pi, which acts as a single board computer and records the detected objects in real time. The Raspberry Pi then retrieves the image data stored in thedatabase and uses the eigenface image processing method to match the object's similarity [1].

Previous studies [2] used the Eigenface method to detect facial recognition. The facial recognition process was used to identify hacking visuals. Studies had found that when there was intruder movement on a computer monitor, the results locked the monitor screen. Previous studies [3] used computer vision techniques that used the method of combining the saturation values of video traffic. The proposed system was designed to automatically calculate how many vehicles were needed to overcome congestion. Previous studies [4] used computer vision techniques to detect the human in CCTV. The YOLOv3 algorithm was a technique used to prevent data from being reused. The accuracy was 91.07% with the normal YOLOv3 model. Previous research designed a plastic shredding machine [5]. The result is that the engine design is quite good because it is easy to use, plastic feed can be chopped well with a machine enumeration capacity of 0.46 kg per hour.

In this study, we analyze objects using the Eigenface method, and the objects themselves are classified as either non-organic or organic. We are using a Raspberry Pi and a NOIR camera that will connect to a chopper machine. The device is capable of detecting and classifying the type of object that is being thrown into the camera, then the system will identify and classify them. When the system has successfully identified and classified an object seen by the camera, it will tive the servo motor in chopper machine when an organic sample is found. The shopper machine will continuously chop the garbage.

II. Methodology

In this study, we designed garbage detection using a NOIR camera and a Raspberry Pi. The tool is then placed on the chopping machine created earlier, as shown in Fig.1. The diagram block of this research is described as shown in Fig.2 First, we collect datasets from multiple photographs for the categories of organic and inorganic waste. Data sets help you train your program to recognize data and recognize specific objects. The dataset used is a collection of photos for each object. Then use the Eigenface method to perform the training process. Then analyze based on the accuracy score and draw conclusions. Python is used as the programming language.

The expected result is the classification of organic and inorganic waste types.

TABLE I. DATASET SAMPLING

No	Picture	Note
1		organic
2		Organic
3		Organic
4		Non-norganic
5		Non-norganic

During the research phase, we will collect a dataset of 50 photos for one photo type to enable the program to recognize objects with high accuracy when viewed from the camera. An example record table is shown in Table I. In this research, the process of image recognition takes several steps as shown in Fig.3. First, the camre will capture the object then we normalize the image. The normalization process is converting the RGB image format to grayscale then grayscale format convert to matrix. After that, we calculate using eigenface then matching the input eigenface value with the eigenface in the database, if it doesn't match it will returns to the photo process from the camera. Whereas if it matches, it will look for eigenvalues that are close to database.

Fig.1. The modul embed with chopping machine

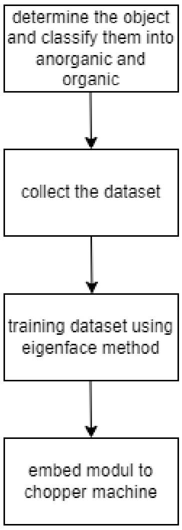

Fig. 2. Diagram Block

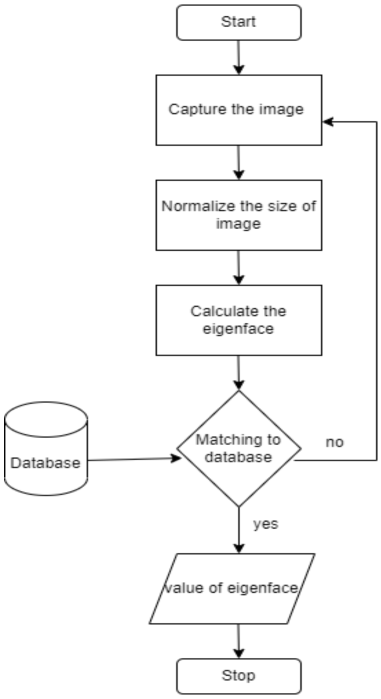

Fig. 3. Flowchart Eigenface

we use the obtained data to carry out the tests performed andaa we use the obtained data to carry out the tests performed and train the program on different data. Then the program can give the result of what type of object is seen by the camera according to Fig.4 [6]. Next we try to combine the Raspberry Pi module into the chopper as in Fig.5.

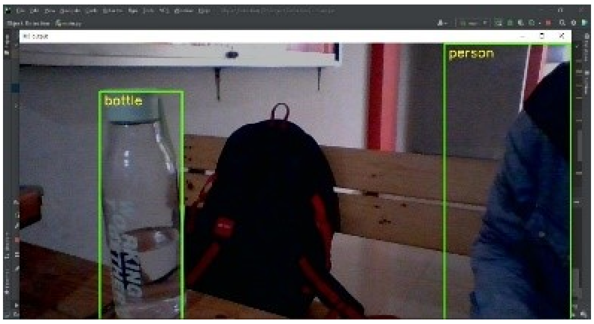

Fig.4. Traning Dataset

Fig.5. Embed the raspberry Pi into chopper machine

A. Eigenface Method

The experiment recognizes a new photo image (Γnew), first applying the Eigenface calculation (μ) to get the eigenvalue of the image as in equation (1)[7].

$$\mu_{new} = v \times (\Gamma_{new} - \Psi),\ \Omega = [\mu_1, \mu_2, ..., \mu_M] \quad (1)$$

Where the new Eigenface (μnew) is generated from the eigenvector (v) times the result of the middle value (ψ) of the new image (Γnew). And omega (Ω) is a set of Eigenface matrix (μ). Then by using the Euclidean distance method to find the shortest distance between the eigenvalues of the training data and the eigenvalues of the new face image as in equation (2.2).

$$\varepsilon_K = \|\Omega - \Omega_K\| .... \quad (2)$$

Where the shortest distance using Euclidean distance is obtained by means of the range of Eigenface training image values (μ) reduced by the Eigenface test image values (μnew).

The first step in the eigenface calculation is compiling a Flatvector Image Matrix by compiling a set of S matrix consisting of all training images ($\Gamma\ 1,\ \Gamma\ 2,\ \ldots\ \Gamma\ m$). Examples of calculations in the first stage are shown in Fig.6 and Fig.7.

Then we calculate the middle value by adding up the matrix values of image 1 and image 2 and then we divide by the number of image data in the database according to equation (2.3). In this study, the number of image data in the database there are 2 image data. After obtaining the value of , we look for the difference (ϕ) between the training image (Γ) and the mean value (Ψ) according to equation (2.4).

$$C_1 = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix}$$

Fig.6. First Citra

$$C_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix}$$

Fig.7. Second Citra

$$\Psi = \frac{1}{2}\sum_{n=1}^{2}\Gamma_n = \frac{1}{2}\left[\begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix} \right]$$

$$\Psi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix} \quad (3)$$

$$\phi_1 = \Gamma_1 - \Psi = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix}$$

$$\phi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\phi_2 = \Gamma_2 - \Psi = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix}$$

$$\phi_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (4)$$

Then we calculate the covariance matrix values. The covariance matrix is used to calculate the eigenvalue (λ) and eigenvector (v) according to equation (2.5). Next we look for the values of and according to equation (2.6).

$$L = \left[\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} x \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right]$$

$$L = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad (5)$$

$$L\ x\ v = \lambda\ x\ v,$$

$$L\ x\ v = \lambda l\ x\ v$$

$$(L - \lambda l) = 0\ atau\ (\lambda l - L) = 0 \quad (6)$$

$$0 = det \begin{bmatrix} \lambda - 2 & -1 & 0 \\ -1 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 3 \end{bmatrix}$$

So the resulting eigenface is $\lambda_1 = 3$, $\lambda_2 = 1$, $\lambda_3 = 3$.. The eigenvector (v) is generated by substituting the eigenvalue (λ) into the equation $L = 0$. The eigenvector of each eigenvalue is obtained based on each column of eigenvalues and then reassembled into one matrix. The next process is the identification process. In this process the image will be matched with the existing image in the database to find out which image matches the newly entered image with the image in the database. To recognize a new face that is entered (test face), the steps taken are the same as the image data in the database, to get the eigenface value of the new face. First find the difference (Φ) between the test face and the mean (Ψ). The test face matrix value of these coordinates is shown in Fig.8.

$$= \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 2 & 4 \end{bmatrix}$$

Fig.8. Testface

In the calculation results, the eigenface distance of the face image one has the smallest distance. Because the eigenface distance of face one with the testface is the smallest, the identification results conclude that the testface is more similar to face one than face two.

III. Result and Discussion

Figure 9 shows the source code for implementing the eigenface method at the dataset reading stage. At this stage, the camera will save images of both organic and inorganic waste, then the data will be processed to determine whether it is in the organic or inorganic category. Fig.10 shows the source code for implementing the eigenface method. In this stage, output will display whether the stored image is organic or inorganic.

In the first test, we tried to create a 1:1 dataset labeled 1a, where this initial training used 1 photo sample dataset from organic and inorganic waste labels. Both of them used a sample of 100 photos. Table II is an example of a confusion matrix resulting from training data. Based on the table above, the predicted test data is 50 photo samples. In the first test, an accuracy of 40% was obtained for inorganic by 46%. The second test uses a dataset labeled 1b, in this second experiment using a dataset with 1:10, where the organic and inorganic labels with 100 photos and the Unknown label with 1000 photos. Table III below is the confusion matrix of the training data results. Based on the table above, the predicted test data is 662. In this second test, the accuracy is 75% for organic and 70% for inorganic.

```
pickle_in = open("pca1.dat","rb")
pca = pickle.load(pickle_in)

#X_train_pca = pca.transform(X_train)
#X_test_pca = pca.transform(X_test)

pickle_in = open("clf1.dat","rb")
clf= pickle.load(pickle_in)


nama = {0: 'anorganik', 1: 'organik'}

#proses face recogn
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
sampah = cv2.CascadeClassifier('sampah1.xml')

cap.set(3,640)
cap.set(4,480)

identitas=[]
ketemu = False
```

Fig.9. Image capture source code

```
trash = sampah.detectMultiScale(gray, 6.5, 17)
test = []
for (x,y,wf,hf) in trash:
    # create box on detected face
    frame = cv2.rectangle(frame,(x,y),(x+wf,y+hf),(255,0,0),1)

    wajah = frame[y:y+hf,x:x+wf]
    dim = (320, 320)

    # resize image
    resized = cv2.resize(wajah, dim, interpolation = cv2.INTER_AREA)
    resized=cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)
    testImageFeatureVector=numpy.array(resized).flatten()
    test.append(testImageFeatureVector)
    testImagePCA = pca.transform(test)
    testImagePredict=clf.predict(testImagePCA)

    cv2.imshow('wframe', resized)
    print('Jenis Sampah', nama[testImagePredict[0]], type(testImagePredict))
    cv2.putText(frame, "Name : " + nama[testImagePredict[0]], (x + x//10, y+hf+20), \
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255, 255), 1)
```

Fig.10. Source code displays image detection results

TABLE II. Data Testing using label 1A

Test	Organik	Unorganik	Unknown
Organik	20	0	30
Unorganik	0	23	27
Unknown	0	0	57

TABLE III. Data Testing using label 1B

Test	Organik	Unorganik	Unknown
Organik	75	0	267
Unorganik	0	70	250
Unknown	0	0	517

IV. Conclusion

Based on the experiment, it can be concluded that the Eigenface method can work well, where the accuracy for organic garbage is 70% and for unorganic garbage is 75%. This is due to the lack of variation in the dataset and a change in the physical condition of the object. On the counting machine, when the object detected is organic or inorganic waste, the door of the enumerator is open.

Acknowledgment *(Heading 5)*

Thank you to the Ministry of Education, Culture, Research, and Technology for providing funding for this research through Penelitian Dosen Pemula (PDP) funding scheme in 2022.

References

[1] T. S. Gunawan, M. H. H. Gani, F. F. A. Rahman, M. Kartiwi, “Development of Face Recognition on Raspberry Pi for Security Enhancement of Smart Home System,” *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 5, no. 4, pp. 317-325, Dec. 2017.

[2] N. Anwar, R. A. Pratama, Sunardi, et.al, “Implementation of Face Recognition Eigenface Method Approach to Detect Visual Hacking,” in *The Second Internatiinal Symposium on Future Information and Communication Technology*, 2021

[3] F. Liu, Z. Zeng, R. Jiang, “A video-based real-time adaptive vehicle-counting system for urban roads” *PLOS*, pp. 1-16, Nov. 2017.

[4] H. Mokayed, T. Z. Quan, L. Alkhaled, V. Sivakumar, “Real-Time Human Detection and Counting System Using Deep Learning Computer Vision Techniques,” *Artificial Intelligence and Application*, vol. 00, no 00, Oct. 2022.

[5] W. Alvarado-Diaz, J. Chicoma-Moreno, B. Meneses-Caludio, L. Nuñez-Tapia, “Design of a Plastic Shredding Machine to Obtain Small Plastic Waste,” *Int. Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.

[6] Rosdiana, V. Novalia, et.al, “Appkication and Attribute Analysis in The Model of Classifying Heart Disease”, *Multica Science and Technology*, vol. 1, no. 2, pp. 72-75, 2021

[7] S. Gurumurthy, C. Ammu, B. Sreedevi, "Age Estimation and gender Calssification Based on Face Detection and Feature Extraction," *Int. Journal of Management&Information* Technology, vol. 4, no.1, pp. 134-140, 2013.