International Journal of Artificial Intelligence,
ISSN 0974-0635; Int. J. Artif. Intell.
2025, Volume 23, Issue Number 1

Ovitrap IoT with Automatic Mosquito Egg Counter Based-on Digital Image Processing

T. Heryono[1] , M. I. Mandasari[2] , E. Joelianto[3] , I. Ahmad[4] , T. Anggraeni[4] , M. E. Prasetyo[5] , S. Tjahjani[6] and **Dani[6]**

[1]Instrumentation and Control Graduate Program, Institut Teknologi Bandung, Indonesia;
Email: 23820305@mahasiswa.itb.ac.id

[2]Built Environment Performance Engineering Research Group' Institut Teknologi Bandung, Indonesia;
Email: mandasari@itb.ac.id

[3]Instrumentation, Control, and Automation Research Group, Institut Teknologi Bandung, Indonesia;
Email: ejoel@itb.ac.id

[4]School of Life Sciences and Technology, Institut Teknologi Bandung, Indonesia;
Email: tjandra@sith.itb.ac.id, intan@itb.ac.id

[5]Biomedical Research Centre of Indonesia, Bandung, Indonesia;
Email: mariaprasetyo1@gmail.com

[6]Faculty of Medicine,
Maranatha Christian University, Indonesia;
Email: susy.tjahjani@med.maranatha.edu, marcello.dani.dd@gmail.com

Correspondence: Email: ejoel@itb.ac.id; Tel.: +62-22-2504424

ABSTRACT

Dengue Hemorrhagic Fever (DHF) has spread rapidly to all regions of the world in recent years and now, based on the 2023 WHO report, has become endemic in more than 100 countries in the WHO Region in Africa, the Eastern Mediterranean, the Americas, the Western Pacific and Southeast Asia. One way to anticipate further spread is to prevent and control the reproduction cycle of dengue vectors such as Aedes aegypti and Aedes albopictus. For this purpose, a prototype was designed of a mosquito egg trap, known as an ovitrap, which can automatically count the number of eggs trapped in it and can be monitored remotely using current technology called Internet of Things (IoT). The prototype was successfully constructed along with a proposed new digital image processing algorithm and an automatic egg counting system. In a real experimental environment, the IoT-ovitrap was used to count the number of eggs in a sample and achieved the best MAPE score using the proposed fusion method at 2.21% with a standard deviation of 3.93.

Keywords: Mosquito Eggs, Image Processing, Internet of Things, Dengue Hemorrhagic Fever

Mathematics Subject Classification: 92-04, 92D40, 92D99, 92-08

Journal of Economic Literature (JEL) Classification Number: I18, O31

1. INTRODUCTION

Mosquitoes are infamous for their role as a vector of various diseases, among which dengue fever (DF) and dengue hemorrhagic fever (DHF) caused by the dengue virus (Candra, 2010). The vectors of this virus come from the Aedes genus, mainly *Aedes aegypti* and to a lesser extent *Aedes*

albopictus (Jorge et al., 2019; Redaksi, 2010; WHO, 2014). *Aedes aegypti* features a small, black body with a typical white circle pattern on the legs and a silvery hue on the upper body (Jorge et al., 2019; Redaksi, 2010).

In the last few years, dengue has spread out quickly over all WHO regions, with Asian and Latin American countries as the most affected countries. This viral disease has developed into a primary cause of hospitalization and death along with children and adults (WHO, 2023). WHO also reported that the number of registered dengue cases had increased about eight-fold over the preceding two decades, with the number of deaths between the years 2000 and 2015 increasing from 960 to 4032 (WHO, 2014).

As a tropical country, Indonesia is considered one of many countries that are endemic to DF (WHO, 2014). Data from the Indonesian Ministry of Health show that the incidence rate (IR) of DHF in Indonesia from 1968 to 2020 always fluctuated but tended to increase (Ministry of Health of the Republic of Indonesia, 2016). For example, according to data from 2011 to 2020, the IR ranged from 24.8 (2016) to 78.9 (2018) cases per 100,000 people. In 2020, there were 108,303 cases. This number was a decrease compared to 2019 with 138,127 cases. In line with the number of cases, deaths due to DHF decreased from 919 to 747 in 2020. Nonetheless, the national average case fatality rate (CFR) remains alarming at 0.70% (Ministry of Health of the Republic of Indonesia, 2016). This situation outlines the need for preventive action to prevent the outbreak from becoming worse.

Vector surveillance is a preventive action that is crucial in controlling and preventing the occurrence of DF (World Health Organization, 2019). Preventive action can be used to determine some factors related to dengue transmission, such as the distribution, population density, and larval habitats of the mosquitoes (World Health Organization, 2011). Some of the vector surveillance methods that are commonly used for both research and disease prevention are ovitrap survey, larval survey, pupae survey, and adult survey (Hotez et al., 2007; VDCI Mosquito Management).

An ovitrap is a device that is widely used in mosquito surveillance to attract mosquitoes to lay their eggs inside it, either for research purposes or for vector control (CDC, 2016). Generally, it consists of a black container with a thin cloth as a medium for mosquitoes to lay their eggs, some water, and an attractant such as hay infusion to bait the mosquitoes (CDC, 2016; Velo, 2016; Polson et al., 2016). Ovitraps are inexpensive, easy to use, and offer a great deal of insight regarding mosquitoes as the disease vector by providing data for indexes such as the pupae index, ovitrap index, positive house index, and ovitrap density index (Focks, 2004; Sasmita et al., 2021). Ovitraps are used in many countries, for example, Hong Kong, Singapore, Taiwan, and Australia (Sasmita et al. 2021; Lee and Fok, 2008; Ooi et al., 2006; Ritchie et al., 2004). However, conventional ovitraps suffer from several disadvantages, one of which is the process of egg counting in the laboratory, which takes a long time and a hard effort. The ability to accurately count mosquito eggs requires skill in using a microscope

and distinguishing debris from eggs. Furthermore, trained personnel are needed to identify mosquitoes from larvae that hatch from mosquito eggs in an ovitrap in the laboratory (CDC, 2016).

These issues occur especially in areas that have limited facilities (Caputo, 2020; Vasconcelos, 2019; Townson, 2005; Hay et al., 2010). Another disadvantage of using ovitraps, is that data is commonly collected only weekly and not in real time, which may be too late if there is an outbreak. With IoT, it is possible to monitor the number of mosquito eggs in real time for a large area so that preventive measures to control mosquitoes can be carried out before a dengue outbreak. This creates the rationale to integrate conventional ovitraps with an automatic counting system and to connect them through IoT. The use of an automatic counting system will reduce the need to count the eggs manually, whilst the IoT system will enable the ovitraps to be operated as well as monitored remotely by users over the network. Furthermore, it will facilitate the ovitraps to store the counting data on a cloud server and visualize them via internet.

Several methods have been developed to automatically count mosquito eggs. In 2016, Gaburro et al. developed the ICount software using a high-resolution camera mounted on a microscope (Gaburro et al., 2016). Although in principle there is no difference with manual counting, what is different is the speed of counting, the number of eggs counted, and the accuracy. Other methods have been developed using digital image processing (Dembo et al., 2014; Mollahosseini et al., 2012; Da Silva et al., 2011). Meanwhile, Hamesse et al., 2023 developed the Ovitrap Monitor, which can count mosquito eggs automatically using digital image processing and report the results on the web (Hamesse et al., 2023). However, a limitation of this prototype is the small coverage area of its automatic counting system, which cannot count eggs located outside the paddle stick. Recently, image processing techniques have been developed to distinguish *Aedes aegypti* and *Aedes albopictus* eggs for real-time applications (Gunara et al., 2023).

This paper presents two innovations of the ovitrap design. The first contribution is a new image processing algorithm to support the automatic egg counting algorithm. The second is the incorporation of IoT to facilitate forwarding the data to a database system for further analysis and interconnection with other application systems. The materials and methods section will discuss the ovitrap design and the image pre-processing procedure. Section three explains the automatic egg counting system and the integration of the ovitraps in IoT. Section four provides a discussion of the experimental result obtained by using the ovitrap equipped with IoT. The last section gives our conclusions.

2. MATERIALS AND METHODS

This work required obtaining a great number of repetitions of samples giving a response to the same known theoretical model. However, as the theoretical model was unknown, Monte-Carlo simulation was used with the input data generated by a computer according to a fixed theoretical model.

2.1. Ovitrap Design

In general, there are three different positions for mosquitoes to organize their eggs (Shragai et al., 2018; Day, 2016). The types of ovopositions are: individually pasted above the water line (Aedes sp mosquito genus), in rafts on the water (Culex sp mosquito genus), and individually floating on the water surface (Anopheles sp mosquito genus). The following figures, Figure 1(a) to 1(c) show the ovopositions of Aedes sp, Culex sp, and Anopheles sp mosquitoes, respectively, where each genus has a different ovoposition preference.

(a) (b) (c)

Figure 1. Ovopositions of mosquitoes: (a) Aedes, (b) Culex, (c) Anopheles. Source:
(a) https://ecommons.cornell.edu,
(b) http://bugoftheweek.com/blog/2014/5/26/there-will-be-blood-mosquitoesculicidae,
(c) https://media.sciencephoto.com/image/c0234413/800wm

The main objective of the present research was to monitor Aedes sp mosquitoes. In this case, the design was focused on modifying the ovitrap, where it functions to lure Aedes sp mosquitoes to lay their eggs inside the trap whilst providing a good angle for a camera to take pictures of the eggs for further image processing. The developed ovitrap design consists of a black container, a black Alvaboard to attach the egg trap cloth to (Figure 2(a)), and a white spandex cloth as a medium for the mosquitoes to lay their eggs on. The use of the color black is based on the host-seeking behavior of mosquitoes, who tend to be attracted by dark and low-reflective color surfaces, especially black (Hoel et al., 2011). A lid was built on top of the ovitrap to prevent unwanted objects, such as debris, from entering (Figure 2 (b)). Alongside it, a Pi camera connected to a Raspberry pi mini-computer was placed in the ovitrap for capturing images of the eggs.

To prevent the water inside the ovitrap from overflowing on rainy days, a drainage system was made, consisting of two holes on each side of the ovitrap (Figure 2(c)). The holes also have a secondary function as an alternative way for mosquitoes to enter the ovitrap. Overall, the hardware design was successful in attracting mosquito to lay their eggs inside the ovitrap. However, a downside is that it might be invaded by unwanted insects or debris from the outside. These kinds of unwanted things could severely affect the accuracy of the automatic counting process. Therefore, in this paper, the prototype was given a treatment before the automatic counting process started by cleaning out any unwanted objects without interfering with the eggs.

(a) (b) (c)

Figure 2. (a) Spandex cloth attached to the Alvaboard, (b) the ovitrap lid, and (c) drainage hole in the ovitrap.

2.2. Image Processing System and Image Pre-processing

In this paper, the number of eggs within the prototype was calculated using the OpenCV library. First, an image pre-processing algorithm was designed. The purpose of this operation is to change the input image format from RGB to binary (black and white) and suppress noise in the image so that the accuracy of the calculation is improved. After that the counting system was designed and the accuracy of the results was calculated. The image pre-processing operation consists of four steps as expressed in the algorithm below:

Algorithm 1: Image pre-processing

Input : Image from the camera

Output :

1) The input image format is changed from RGB to grayscale format for the thresholding operation.
2) The thresholding operation is accomplished on the image with an image in binary format as output.
3) The opening operation is performed on the image to reduce any existing noise in the image and separate any connected objects due to the proximity of eggs.
4) A noise-cleaning process is performed by setting a minimum pixel limit value; objects with value below the limit will be considered noise and removed.

After this series of processes is completed, the image is forwarded to the calculation process.

2.3. Binary-Inverse Thresholding

Thresholding is a segmentation method used frequently in image processing. This method can be used to separate an area in an image to be further investigated. In this paper, the thresholding method used is inverse-binary thresholding. This technique is the opposite of binary thresholding, where any pixels that have a value above the threshold rate will be converted to 0 (black) and otherwise to 1 (white). To be able to apply the function, the image first needs to be converted to grayscale. The binary-inverse thresholding operation can be written as follows (OpenCV, 2022):

$$dst(x,y) = \begin{cases} 0, src(x,y) > \text{Threshold} \\ \max val, \text{Otherwise} \end{cases} \quad (1)$$

where $src(x,y)$ represents the resource image and $dst(x,y)$ denotes the target image.

Another definition of thresholding, as explained in (Sahoo et al.,1988; Siswoyo et al., 2022), is the following: let N be the set of natural numbers, (x,y) the spatial coordinates of a digitized image, and $G = (0,1,\ldots I-1)$ a set of positive integers signifying gray levels. Afterward, an image function can be expressed as the mapping $f : N \times N \to G$. The brightness (i.e., gray level) of a pixel with coordinates (x,y) is represented as $f(x,y)$.

Let $t \in G$ be the threshold and $B = \{b_0, b_1\}$ be a pair of binary gray levels, where $b_0, b_1 \in G$. The result of thresholding an image function $f(\because)$ at gray level t is a binary image function $f_t : N \times N \to B$, such that

$$f_t(x,y) = \begin{cases} b_0, if\ f(x,y) < T \\ b_1, \text{ Otherwise} \end{cases} \quad (2)$$

To get a clean output image from the thresholding operation, the threshold value has to be optimized. The value of the threshold parameters was searched manually by taking an image of the ovitrap sample, iterating the threshold with different parameter values and counting the number of eggs both manually and using the connected components function. The accuracy was then found by using the absolute percentage error (APE) formula. The formula for the iterations can be written as follows:

$$Z(i) = 100 - 20i \quad (3)$$

where $Z(i) = \text{Treshold value}$ at the i-th iteration and $Z(i) > 0$.

The value of 100 was chosen as the initial iteration parameter under the consideration that there would be too much noise generated if a higher value was used. From the iterations, a range of parameter value results was found that produces an image with minimum noise and can display all the mosquito eggs contained in the image. The optimal initial parameter range was found to be between 60 and 40. Furthermore, from this initial range, a further iteration was carried out after reducing the subtraction constant from 20 to 5. The result was a new, more optimal range with a value between 40 and 55.

After obtaining the new upper and lower value ranges of the threshold, further optimization of these parameters was carried out by iterating the egg count automatically using the connected components

function at each value in the parameter range and then calculating the counting accuracy value by comparing it to the actual number of mosquito eggs in the sample used. The accuracy of the automatic counting results was calculated using the formula for absolute percentage error and relative accuracy as follows:

$$\mathrm{APE}\left(y_i, \hat{y}_i\right) = \frac{\left|y_i - \hat{y}_i\right|}{y_i}.100\% \tag{4}$$

Relative accuracy = 100% - APE, where $\hat{y}_i$ = the result of the automatic counting of eggs from the i-th sample, y_i = the result of counting eggs manually from i samples. After doing the iterations, the optimal threshold value was found to be 53. Figure 3 below shows the correlation between the threshold value and the accuracy score.

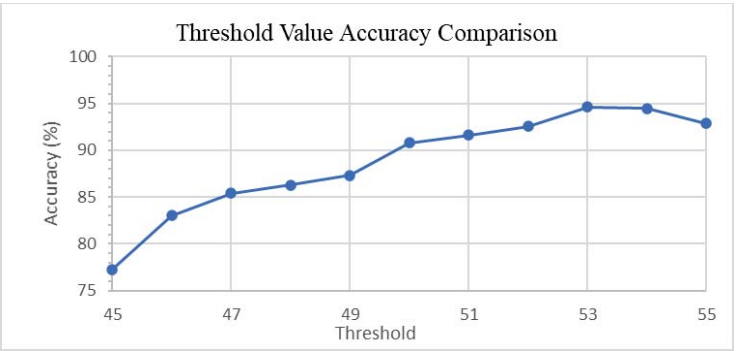


Figure 3. Threshold value chart.

2.4. Opening

The next step after applying the binary-inverse thresholding operation was executing the opening operation. Opening is a morphological operation feature of OpenCV that is a combination of erosion followed by dilation. This operation is usually performed on binary images, although it can also be used on grayscale images. Generally speaking, the opening operation smoothes the contour of an object, breaks narrow isthmuses, and removes thin protrusions.

Erosion is a process that convolutes an image using a kernel. A kernel is an area that can be utilized in many forms, such as a square, a circle, and so on. It also has an anchor point, which is usually located in the middle of the kernel area. When the operation starts, the kernel scans the original image. If the value of all pixels covered by the kernel area is 1 (white) or 0 (dark) then the image will be left in its original form. However, if it contains both 0 and 1 then its value will be converted to 0 at the anchor point of the kernel. This results in the light areas of the image becoming thinner while the dark areas of the image becoming thicker (Figure 4(c)). The dilation operation is the reverse of the erosion operation. It makes the light areas of the image become thicker and vice versa (Figure 4(d)).

Thus, the output object of the opening operation becoming thinner because of the erosion operation is reversed whilst the noise in the image is reduced.

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

Figure 4. Opening operation: (a) source image, (b) image after binary-inverse thresholding, (c) image after erosion, (d) image after dilation.

The opening of set A by arranging element B, designated by A ◦ B, is stated as: $A \circ B = \left(A\,!\;\; B \right) \oplus B$ which has the following geometrical interpretation: the opening of A by B is the union of all the translations of B so that B fits entirely in A.

2.5. Noise Elimination

After the threshold and opening operations have been executed, the next step in the image pre-processing phase is noise elimination. Noise elimination is an operation that deletes objects above a minimum pixel size. The purpose of this is to eliminate objects with a small size, which are most likely noise. First, the minimum pixel value must be set. This is done by selecting several image samples that have different numbers of mosquito eggs or come from different ovitraps and then performing the thresholding and opening operations on the image. After that, the sizes of all objects in the sample image are checked using the connected components function. After obtaining the size data of each object, the noise elements are searched by manually checking the sizes and positions of the objects based on the information provided from the connected components function. The noise elements can be identified based on their size, which most likely is tiny compared to that of the other elements in the image.

2.6. Egg Counting Methods

In this work, three counting methods were designed to be used later in the mosquito egg counting system. These counting methods are the estimation method, the connected components method, and a combination of the estimation and the connected components function. Out of these methods, the best method was selected to be used as the main counting system for the prototype. A treatment was given to the prototype, i.e., the ovitrap was cleaned from any unwanted object except the mosquito eggs.

3. RESULTS AND DISCUSSIONS

3.1. Estimation Method

The egg counting system is carried out by dividing the total area of the object contained in the pre-processed image by the average area of a mosquito image that has been calculated first (Mello et al., 2008; Bandong et al., 2019). In this case, all objects in the input image are assumed to be mosquito eggs. The total area is computed based on the number of white pixels in the image after doing the pre-processing operation to the input image. Below is the explanation of this algorithm.

Algorithm 2: Estimation Method

Input : Image taken from the camera

Output : Counting result

1. Take the ovitrap sample with the most eggs.
2. Count the eggs manually.
3. Perform binary-inverse thresholding on the sample image.
4. Calculate the object area of the sample image.
5. Divide the object area by the number of eggs (stored as $\bar{\alpha}$).
6. Perform image pre-processing.
7. Calculate the white pixel area of the input image (stored as A).
8. Divide (A) by ($\bar{\alpha}$).

In this method, first, an estimation of the average area of the mosquito eggs (ᾱ) is calculated. This is done by taking an image of the ovitrap sample that has the most mosquito eggs, pre-processing the image and then dividing the area of the object in the image by the actual number of mosquito eggs.

The formula of $\bar{\alpha}$ can be written as:

$$\bar{\alpha} = \frac{A}{N} \tag{5}$$

where: A = the total pixel area of mosquito eggs, N = the actual number of mosquito eggs, $\bar{\alpha}$ = the average pixel size of the mosquito eggs. The sample used to find the value of $\bar{\alpha}$ was sample number 10 in Table 1. From the calculations, the value of $\bar{\alpha}$ was found to be 66.6 pixels. After obtaining the value of $\bar{\alpha}$, the method was used to count the eggs in the input image after applying the pre-processing operations to it.

The number of eggs was calculated by means of the following formula:

$$\bar{N} = \frac{A}{\bar{\alpha}} \tag{6}$$

where $\bar{N}$ = the estimated number of mosquito eggs. The result of the estimation method is shown in Table 1. From this method, the relative accuracy was found to be 74.26%. However, this result was still far from our goal of around 90% accuracy.

There are several factors that caused this low accuracy score:

1. Different egg sizes. The difference between the value of ᾱ and the pixel size of each mosquito egg results in a significantly lower accuracy of the method.
2. Uneven light intensity in the ovitrap. The uneven intensity of light, especially between the center and the edge area of the egg trap, affects the image size of the eggs in the image, where eggs that get less light intensity will have a smaller size in the image. This makes the resulting pixel size of the egg image disproportionate.
3. Effect of embryogenesis. When the eggs are exposed to water, embryogenesis will occur. This results in the production of extracellular matrix cells, which results in enlargement of the volume and thickening of the eggshell (Indonesia Ministry of Health, 2018).
4. Side effects of the threshold and opening operations. When the input image is pre-processed with the thresholding and opening operations, the output image become smaller in size. This effect contributes to a lower accuracy of the counting method.

3.2. Connected Components Method

To increase the accuracy of the counting system, a novel method is proposed in this paper. This method uses a connected component function to count the number of objects in the picture. Connected components is a function of the OpenCV module that is widely used in the segmentation and identification of binary images. This function can also be used in detecting and counting objects in an image (Dharpure, 2013). It consists of an algorithm that can be used to check whether two or more objects in an image are connected based on the pixel connectivity between these objects. Pixel connectivity can be interpreted as a condition that makes two or more pixels connected, for example based on spatial distance and pixel brightness (Fisher et al., 2000).

The connected components function itself can be defined as in (Di Stefano, 1999). Let I be a binary image and F and B, the subsets of I that contain the foreground and background pixels respectively. The connected components in I, here denoted to as C, are a subset of F of maximal size, meaning that all pixels in C are connected. Two pixels, P and Q, are connected if there exists a path of pixels $\left(p_0; p_1 \ldots p_n\right)$ such that $p_0 = P$, $p_n = Q$, and $\forall 1 \le i \le n; p_{i-1}$ and p_i are neighbors. Consequently, the definition of connected components depends on that of a pixel's neighborhood: if this includes four neighbors, C is said to be four-connected. Assuming that a pixel has eight neighbors, C is said to be eight-connected. The mechanism of the connected components work is described in (He et al., 2017). The algorithm of the counting system is expressed as follows:

Algorithm 3: Connected Components Method

Input : Image taken by the camera
Output : Counting result

1. Pre-process the image.
2. Apply the connected components function.
3. Count the results.

After executing the pre-processing procedure on the input image, the mosquito eggs in the image were counted using this method. The counting result is shown in Table 2, which shows that the accuracy score for this counting method was far superior to that of the previous method. This was because the variety in egg size does not affect the counting process of this method, while pixel connectivity plays a big role in deciding which objects in the image are considered single mosquito eggs.

However, this method has a weakness. It cannot count the exact number of eggs that coincide or are very close to each other. As can be seen from samples 13 and 14 in the table, the accuracy decreased significantly when the image contained some coinciding and very closely positioned eggs. These objects will be counted as one egg by this method, as the connected components algorithm cannot differentiate them. To tackle this problem, another counting method that can count individual eggs along with coinciding eggs was designed.

3.3. Fusion Method

A new counting method, called the fusion method, is proposed here to count individual eggs along with coinciding eggs and eggs that are positioned close to each other. This method is called 'fusion' method because it combines the two methods described above, i.e., the estimation and the connected components method. In the fusion method, coinciding eggs are counted by using the estimation method, while the rest are counted using the connected components method. The algorithm of the method is shown below.

Algorithm 4: Fusion Method

Input : Image taken by the camera
Output : Counting result

1. Pre-process the image.
2. Apply the estimation method to overlapping eggs (stored as Λ).
3. Apply the connected components method to non-overlapping eggs (stored as Φ).
4. Sum the results of the estimation and the connected components method ($\Lambda + \Phi$).

A workflow diagram of the fusion method is shown in Figure 5.

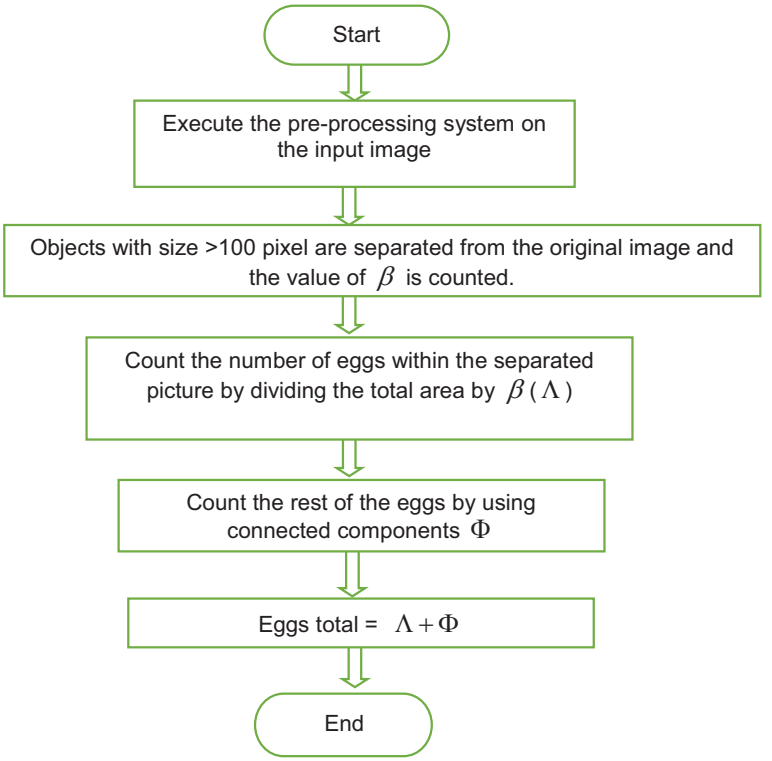


Figure 5. Fusion method workflow diagram.

In the fusion method, the first step is to estimate the value of β. This parameter is the estimation of the pixel size of an egg when it is adjacent to other eggs. In this research, the value β was found by choosing the largest coinciding egg pixel size and then dividing it by the actual number of coinciding eggs. The formula of β can be written as:

$$\beta = \frac{B}{M} \tag{7}$$

where B denotes the total pixel area of the selected coinciding mosquito eggs, M is the actual number of coinciding mosquito eggs, and β denotes the estimated pixel size of the coinciding mosquito eggs.

After β has been found, the size of every object in the image is scanned. Objects with pixel size >100 pixel are assumed to be coinciding eggs and then separated from the original picture. The use of 100 pixels as the minimum value is based on the sample image that has previously been scanned for objects. After this, the separated objects are counted by dividing their pixel size by β. The result is then saved as variable Λ. The rest of the objects are then counted by using the connected components method and the result is saved as variable Φ. Finally, the total number of

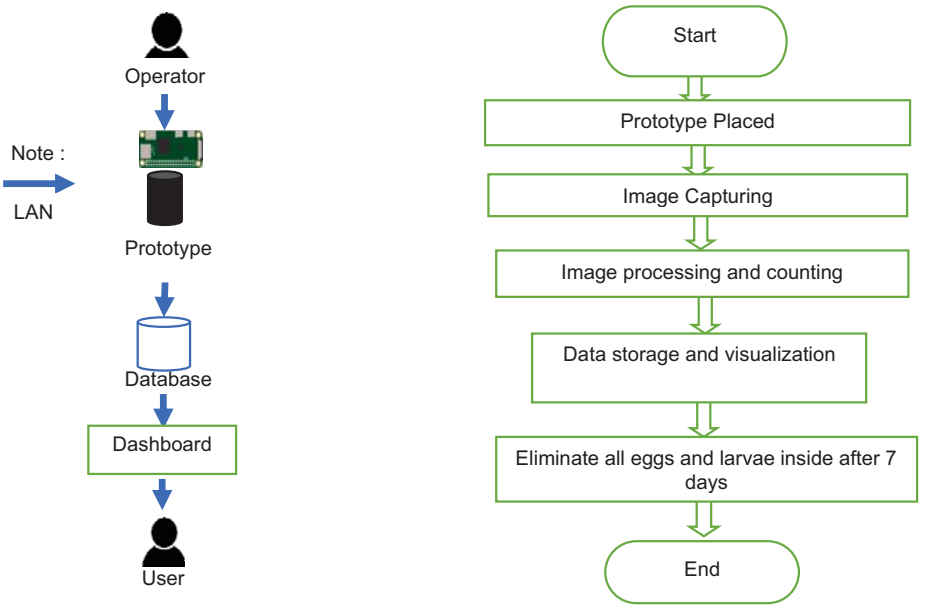
eggs can be calculated by adding the values of Λ and Φ. The final result of this counting method can be written as follows:

$$\Sigma = \Lambda + \Phi \tag{8}$$

where Σ = the total number of eggs, Λ = the estimated number of coinciding eggs, and Φ = the number of normal eggs. In this paper, the value of β was found to be 93 pixels. The calculation results and the accuracy of this method are shown in Table 3. The result of the estimation, the connected components method, and the fusion method yielded a MAPE value of 25.74%, 4.08%, and 2.21%, respectively. From there, it can be inferred that the fusion method's accuracy was the highest among these three methods.

3.4. Design of the System Architecture & IoT

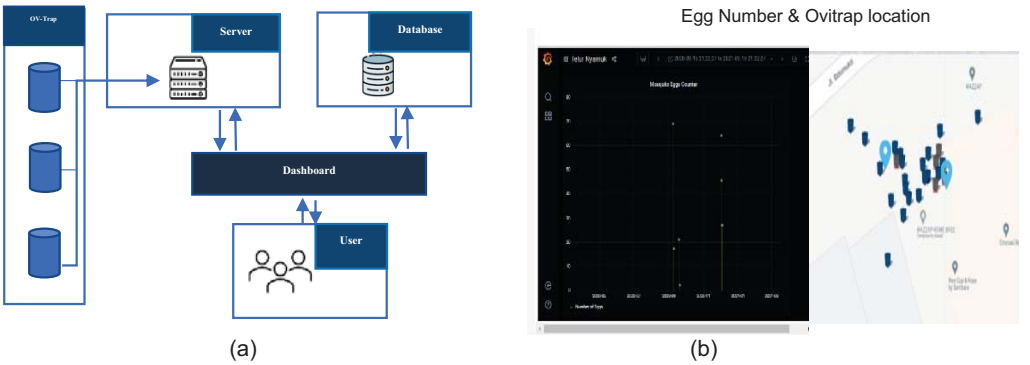
The Internet of Things (IoT) has grown in popularity and has been combined with algorithms based on artificial intelligence (AI) to obtain systems capable of handling complex problems (Téllez et al., 2018; Alshehri and Muhammad, 2020; Sihombing et al., 2021; Suratkar et al., 2021). Figure 6 illustrates the components of the IoT system developed in this research. The prototype used a micro-computer as the computing system and a camera attached to the upper lid of the ovitrap to take mosquito egg images.



(a) (b)

Figure 6. (a) System architecture, (b) prototype workflow diagram.

The IoT system facilitates the ovitraps and their microcomputer to be interconnected over the internet. Figure 7(a) shows the architecture of the IoT system of the prototype. This feature allows users to operate the ovitraps and monitor the egg abundance in the surveillance area in real-time via internet, where the counting data are stored in a cloud database and visualized as shown in Figure 7 (b).



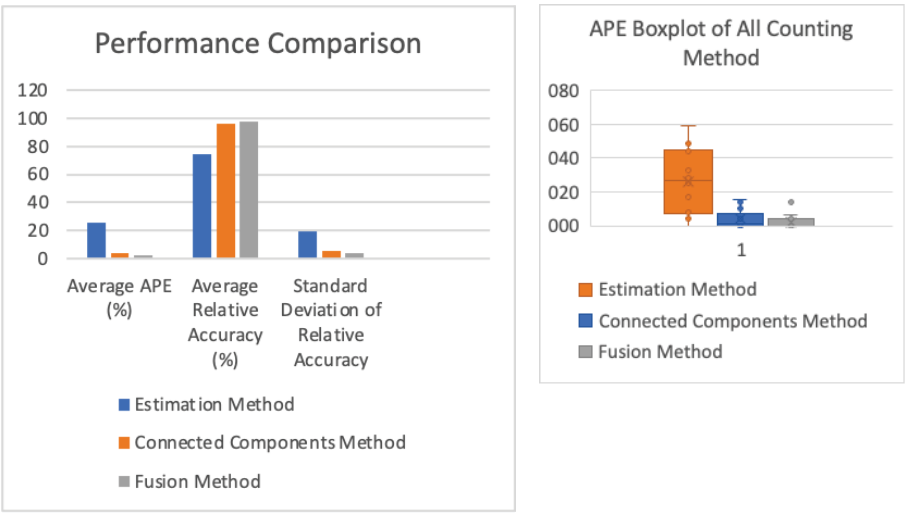
(a) (b)

Figure 7. (a) IoT system architecture, (b) system dashboard.

4. DISCUSSION

The designed prototype along with its systems was tested by placing it in a house yard and checking it regularly on a weekly basis. The checking process consisted of egg counting – both automatically and manually – and the disposal of the eggs and any present larvae. From the technical side, by using the designed prototype along with the automatic counting system, the burden of counting mosquito eggs manually is eliminated, as there is no need to use trained personnel or laboratory facilities such as a microscope anymore. It is also worth noting that the time-consuming procedure of using conventional ovitraps is eased.

Figures 8(a) and (b) show a performance comparison of each method and an APE value boxplot of all counting method results. It shows that the estimation, connected components, and fusion methods yielded a MAPE value of 25.74%, 4.08%, and 2.21%, respectively. The fusion method was the most accurate method due to its ability to accurately count adjacent and/or coinciding eggs as well as individual eggs. It counts individual eggs by using the connected components function and coinciding eggs by using the estimation method separately. The only downside of this method is that it cannot count the exact number of coinciding eggs as it is very dependent on the shape of the coinciding eggs. If the value of β is set too high or too low compared to the size of the coinciding eggs, then the accuracy of this method is low. However, it must be noted that the prototype was treated before the counting process by cleaning the spandex cloth from any unwanted objects such as debris and insects.



(a) (b)

Figure 8. (a) Performance comparison, (b) APE boxplot of all counting methods.

4. CONCLUSIONS

This paper proposed a prototype mosquito egg trap with an automatic counting system. The prototype consisted of an ovitrap, a mini-computer, and a camera connected to each other in an IoT system. This prototype worked by attracting mosquitoes to lay eggs in it, taking pictures of the eggs inside, and then automatically counting the number of eggs using the designed egg counting system. Three automatic egg counting methods were designed, namely the estimation method, the connected components method, and a fusion method. The latter is a combination of the connected components method for separate eggs and the estimation method for eggs that coincide with each other. The proposed method had an accuracy rate of 97.79%. To operate the prototype, the user must first connect to the ovitrap via Wi-Fi and then access a dashboard where commands are available to take pictures of the eggs in the ovitrap, and count and visualize them.

An Internet of Things (IoT) based ovitrap has great potential in monitoring and controlling dengue fever nationally and globally with various benefits. Ovitraps combined with IoT can monitor mosquito eggs in real-time to count them and measure environmental conditions to monitor the development of *Aedes aegypti* mosquitoes. This tool can predict the possible spread of dengue fever by analyzing egg growth and environmental data to identify high-risk areas and provide recommendations or solutions quickly. An early warning system can provide information to the public regarding the risk of spreading dengue fever in risk areas. The system can optimize the placement of ovitraps based on the data that has been collected. The public can connect to the system so that there is community involvement through the use of mobile applications or online platforms to report and obtain information quickly. This tool can also be used to optimize resource use by increasing the efficiency of larvicide use and vector control measures using the ovitraps. Collaboration between countries can be

implemented to support global data exchange to provide more effective early warnings of possible dengue fever outbreaks. It is hoped that the application of this technology can increase the effectiveness of efforts to prevent dengue fever nationally and globally.

ACKNOWLEDGMENTS

This work was supported by Innovative-Productive Research Invitation-RISPRO, KEP. 52/LPDP/2019, Ministry of Finance, 2020-2023, Indonesia. The authors are thankful to the Institute for Innovation and Entrepreneurship Development (LPIK), Institut Teknologi Bandung (ITB), Bandung, Indonesia for providing administrative and technical support. The authors are also thankful to CINOVASI Ltd., Bandung, Indonesia for providing technical support to the conducted research. The authors are thankful to Ramadhani Eka Putra, Ph.D., School of Life Sciences and Technology, ITB, Bandung, Indonesia for providing critical comments on the research.

REFERENCES

Alshehri, F., Muhammad, G., 2020, A comprehensive survey of the Internet of Things (IoT) and AI-based smart healthcare. *IEEE Access* **9**, 3660-3678.

Bandong, S., Joelianto, E., 2019, Counting of Aedes Aegypti Eggs using Image Processing with Grid Search Parameter Optimization. *ICSECC 2019 – Int. Conf. Sustain. Eng. Creat. Comput. New Idea, New Innov. Proc.*, 293–298.

Candra, A., 2010, Dengue Hemorrhagic Fever Epidemiology, Pathogenesis, and Its Transmission Risk Factors. *Aspirator J. Vector Borne Dis. Stud.* **2**(2), 110–119.

Caputo, B., Manica, M., 2020, Mosquito surveillance and disease outbreak risk models to inform mosquito-control operations in Europe. *Current Opinion in Insect Science*/ **39**, 101–108.

CDC, 2016, Surveillance and Control of Aedes aegypti and Aedes albopictus in the United States. Centers Dis. Control Prev., 1–16.

Da Silva, M.G.N.M., Rodrigues, M.A.B., De Araujo, R.E., 2011, Aedes aegypti egg counting system. *in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 6810–6812.

Day, J.F., 2016, Mosquito oviposition behavior and vector control. *Insects*, **7**(4), p.65.

Dembo, E., Ogboi, J., Abay, S., Lupidi, G., Dahiya, N., Habluetzel, A. and Lucantoni, L., 2014, A user friendly method to assess Anopheles stephensi (Diptera: Culicidae) vector fitness: fecundity. *Journal of Medical Entomology* **51**(4), 831-836.

Dharpure, J.K., Potdar, M.B., Pandya, M., 2013, Counting Objects using Homogeneous Connected Components. *Int. J. Comput. Appl.* **63**(21), 31–37.

Di Stefano, L., Bulgarelli, A., 1999, A simple and efficient connected components labeling algorithm. *Proc. Int. Conf. Image Anal. Process. ICIAP 1999*, 322–327.

Fisher, R., Perkins, S., Walker, A., Wolfart, E., 2000, Pixel Connectivity.
[Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/connect.htm. [Accessed: 01-Sep-2020].

Focks, D.A., 2004, *A Review of Entomological Sampling Methods and Indicators for Dengue Vectors*. World Health Organization on behalf of the Special Programme for Research and Training in Tropical

Diseases, 2003.

Gaburro, J., Duchemin, J.B., Paradkar, P.N., Nahavandi, S., Bhatti, A., 2016, Assessment of ICount software, a precise and fast egg counting tool for the mosquito vector Aedes aegypti. *Parasites and Vectors* **9**(1): 1–9.

Gunara, N.P., Joelianto, E., Ahmad, I., 2023, Identification of Aedes aegypti and Aedes albopictus eggs based on image processing and elliptic Fourier analysis. *Scientific Reports* **13**(1), 17395.

Hamesse, C., Andreo, V., Gonzalez, C.R., Beumier, C., Rubio, J., Porcasi, X., Lopez, L., Guzman, C., Haelterman, R., Shimoni, M., Scavuzzo, C.M., 2023, Ovitrap Monitor-Online application for counting mosquito eggs and visualisation toolbox in support of health services. *Ecological Informatics* **75**, 102105.

Hay, S.I., Sinka, M.E., Okara, R.M., Kabaria, C.W., Mbithi, P.M., Tago, C.C., Benz, D., Gething, P.W., Howes, R.E., Patil, A.P. and Temperley, W.H., 2010, Developing global maps of the dominant Anopheles vectors of human malaria. *PLoS Medicine* **7**(2), e1000209.

He, L., Ren, X., Gao, Q., Zhao, X., Yao, B. and Chao, Y., 2017, The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition* **70**, 25-43.

Hoel, D.F., Obenauer, P.J., Clark, M., Smith, R., Hughes, T.H., Larson, R.T., Diclaro, J.W., Allan, S.A., 2011, Efficacy of Ovitrap Colors and Patterns for Attracting Aedes albopictus at Suburban Field Sites in North-Central Florida1. *Journal of the American Mosquito Control Association* **27**(3), 245-251.

Hotez, P.J., Molyneux, D.H., Fenwick, A., Kumaresan, J., Sachs, S.E., Sachs, J.D., Savioli, L., 2007 Control of neglected tropical diseases. *New England Journal of Medicine* **357**(10), 1018-1027.

Indonesia Ministry of Health, 2018, Data dan Informasi profil Kesehatan Indonesia 2018.

Jorge, M.R., de Souza, A.P., dos Passos, R.A., Martelli, S.M., Rech, C.R., Barufatti, A., do Amaral Crispim, B., dos Santos Nascimento, H., de Arruda, E.J., 2019, *The Yellow Fever Mosquito Aedes aegypti (Linnaeus): The Breeding Sites. In Life Cycle and Development of Diptera*, 1-21. IntechOpen.

Lee, M.W., Fok, M.Y., 2008, Dengue vector surveillance in Hong Kong-2007. *Dengue Bull*. **32**, 38-43.

Mello, C.A., Dos Santos, W.P., Rodrigues, M.A., Candeias, A.L.B., Gusmao, C.M., 2008, August, Image segmentation of ovitraps for automatic counting of Aedes aegypti eggs. *In 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 3103-3106, IEEE.

Ministry of Health of the Republic of Indonesia. 2016. Indonesia Dengue Hemorrhage Fever Situation. Pusat Data dan Informasi Kementrian Kesehatan RI. 1–12.

Mollahosseini, A., Rossignol, M., Pennetier, C., Cohuet, A., Anjos, A.D., Chandre, F., Shahbazkia, H.R., 2012, A user-friendly software to easily count Anopheles egg batches. *Parasites and vectors* **5**, 1-7.

"OpenCV. *4.8.0-dev. Miscellaneous Image Transformations*." 2022. [Online]. Available: https://docs.opencv.org/4.x/d7/d1b/group__imgproc__misc.html#ggaa9e58d2860d4afa658ef70a9b1115576a19120b1a11d8067576cc24f4d2f03754.

Ooi, E.E., Goh, K.T., Gubler, D.J., 2006, Dengue prevention and 35 years of vector control in Singapore. *Emerging Infectious Diseases* **12**(6), 887.

Polson, K.A., Curtis, C., Seng, C.M., Olson, J.G., Chantha, N., Rawlins, S.C., 2002, The use of ovitraps baited with hay infusion as a surveillance tool for Aedes aegypti mosquitoes in Cambodia. *Dengue Bull.* **26**, 178–184.

Redaksi, 2010, Demam Berdarah Dengue. *Buletin Jendela Epidemiologi* **2**, August, 1-48.

Ritchie, S.A., Long, S., Smith, G., Pyke, A., Knox, T.B., 2004, Entomological investigations in a focus of dengue transmission in Cairns, Queensland, Australia, by using the sticky ovitraps. *Journal of Medical Entomology* **41**(1), 1-4.

Sahoo, P.K., Soltani, S.A.K.C., Wong, A.K., 1988, A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing* **41**(2), 233-260.

Sasmita, H.I., Neoh, K.B., Yusmalinar, S., Anggraeni, T., Chang, N.T., Bong, L.J., Putra, R.E., Sebayang, A., Silalahi, C.N., Ahmad, I., Tu, W.C., 2021, Ovitrap surveillance of dengue vector mosquitoes in Bandung city, West Java province, Indonesia. *PLoS Neglected Tropical Diseases* **15**(10), e0009896.

Shragai, T., Mader, E., Harrington, L., 2018, *Egg Identification Guide for Aedes albopictus in the Northeast, USA*. Cornell University.

Sihombing, P., Herriyance, H., Syaputra, M.R., 2021, Smart System to Prevent Forest Fire Based on Internet of Things. *Internetworking Indonesia Journal* **13**(1), 45-50.

Siswoyo, A. A., Joelianto, E., Sutarto, H. Y., 2022, Traffic Congestion Estimation using Video without Vehicle Tracking. *Internetworking Indonesia Journal* **14**(1), 33-37.

Suratkar, S., Dhapre, M., Vaje, A., 2021, Long Short Term Memory and Gated Recurrent Unit Predictive Models for Industrial Control Systems. *International Journal of Artificial Intelligence* **19**(1), 138-156.

Téllez, N., Jimeno, M., Salazar, A., Nino-Ruiz, E., 2018, A tabu search method for load balancing in fog computing. *International Journal of Artificial Intelligence* **16**(2), 1-30.

Townson, H., Nathan, M.B., Zaim, M., Guillet, P., Manga, L., Bos, R., Kindhauser, M., 2005, Exploiting the potential of vector control for disease prevention. *Bulletin of the World Health Organization* **83**(12), 942-947.

Vasconcelos, D., Nunes, N., Ribeiro, M., Prandi, C., Rogers, A., 2019, LOCOMOBIS: A low-cost acoustic-based sensing system to monitor and classify mosquitoes. *In 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 1-6, IEEE.

VDCI Mosquito Management, Protecting Public Health: Mosquito Surveillance and Disease Testing. [Online]. Available: https://www.vdci.net/surveillance-disease-management-experts/.

Velo, E., Kadriaj, P., Mersini, K., Shukullari, A., Manxhari, B., Simaku, A., Hoxha, A., Caputo, B., Bolzoni, L., Rosà, R., Bino, S., 2016, Enhancement of Aedes albopictus collections by ovitrap and sticky adult trap. *Parasites & Vectors* **9**(1), pp.1-5.

World Health Organization, 2011, Comprehensive guideline for prevention and control of dengue and dengue haemorrhagic fever.

World Health Organization, 2014, Dengue and severe dengue. WHO Fact Sheet.

World Health Organization, 2019, Promoting Dengue Vector Surveillance and Control. World Health Organization: Geneva, Switzerland.

World Health Organization, 2023, Dengue and severe dengue, 17 March 2023. Accessed 04/01/2024. https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue

APPENDIX

Table 1: Estimation method results for $\bar{\alpha}$ = 66.6

Nr.	Actual Number	Result	APE	Relative Accuracy (%)
1	6	5	16.67	83.33
2	16	11	31.25	68.75
3	17	9	47.06	52.94
4	19	20	5.26	94.74
5	22	9	59.09	40.91
6	27	15	44.44	55.56
7	29	15	48.28	51.72
8	50	46	8.00	92.00
9	70	47	32.86	67.14
10	71	68	4.23	95.77
11	83	62	25.30	74.70
12	84	60	28.57	71.43
13	107	117	9.35	90.65
14	110	110	0.00	100
Average (%)			25.74	74.26
Standard Deviation of Relative Accuracy			19.06	

Table 2: Connected components results

Nr.	Actual Number	Result	APE	Relative Accuracy (%)
1	6	6	0.00	100.00
2	16	15	6.25	93.75
3	17	17	0.00	100.00
4	19	19	0.00	100.00
5	22	21	4.55	95.45
6	27	27	0.00	100.00
7	29	25	13.79	86.21
8	50	52	4.00	96.00
9	70	70	0.00	100.00
10	71	69	2.82	97.18
11	83	83	0.00	100.00
12	84	84	0.00	100.00
13	107	96	10.28	89.72
14	110	93	15.45	84.55
Average (%)			4.08	95.92
Standard Deviation of Relative Accuracy			5.44	

Table 3: Fusion method results with β = 93.

Nr.	Actual Number	Result	APE	Relative Accuracy (%)
1	6	6	0.00	100.00
2	16	15	6.25	93.75
3	17	17	0.00	100.00
4	19	19	0.00	100.00
5	22	21	4.55	95.45
6	27	27	0.00	100.00
7	29	25	13.79	86.21
8	50	52	4.00	96.00
9	70	70	0.00	100.00
10	71	72	1.41	98.59
11	83	83	0.00	100.00
12	84	84	0.00	100.00
13	107	107	0.00	100.00
14	110	109	0.91	99.09
Average (%)			2.21	97.79
Standard Deviation of Relative Accuracy			3.93	