GUBJSE: ISSN: 2409-0476

# An Automated Testing Framework for Gesture Recognition System using Dynamic Image Pattern Generation with Augmentation

Md. Ashaduzzaman, Sheikh Monirul Hasan, Md. Saiful Islam and Muhammad Aminur Rahaman

Abstract—In the field of information technology, the gesture recognition system plays a very essential role. As it has achieved vast importance, it is mandatory to test the recognition system to ensure the quality of the system by identifying the bugs in the software. In our research, we suggested a dynamic testing method for gesture recognition software. using dynamic image pattern generation with augmentation. The automated software testing framework is a set of processes to create new test cases for properly testing a image processing software. The research intention for generate automated testing cases by following a standard process which helps to increase the performance and efficiency of the gesture recognition system. We have built the framework to give proper testing and give result (accuracy and defect) for which gesture recognition system already in the market. our research, the team strongly following and adding two software testing standard. First one is ISO/IEC/IEEE/291129-3 to define the process for testing software. And the second one is ISO/IEC/IEEE/291129-5 to implement the techniques for software testing. We proposed this framework with major five parameters by noise, rotation, background, contrast, and scale. Which are the most use with every gesture recognition system. Our developed framework's phase is used to generate new testing cases based on the existing gesture recognition system's data. There are we work with five systems, commonly with the gesture recognition for experiments. We provide the testing report with total accuracy and defect by comparing existing well-known system's data. At the final result, our system suggested an analysis report based on the testing result. And tell what are the improvement needs for the existing system to consider noised images or different scaled images to build a robust system.

Index Terms—gesture recognition, test automation frameworks, dynamic image pattern generation, Augmentation, software testing standard.

**DIO:** https://doi.org/10.3329/gubjse.v7i0.54023

This paper was received on 7 June 2020, revised on 11 February 2021 and accepted on 19 April 2021.

Md. Ashaduzzaman is with Dept. of Computer Science of Engineering of GUB, mail: ashaduzzaman@cse.green.edu.bd

Sheikh Monirul Hasan is with Dept. of Computer Science of Engineering of GUB, mail: sheikhmonirulhasan@gmail.com

Md. Saiful Islam is with Dept. of Computer Science of Engineering of GUB, mail: aiful.cse.gub@gmail.com

Muhammad Aminur Rahaman is with Dept. of Computer Science of Engineering of GUB, mail: aminur@cse.green.edu.bd

## I. INTRODUCTION

THE gesture recognition system plays a key role in the area of artificial intelligence because it's capability to interact with the electronic device efficiently. This system is able to connect human gestures through the techniques of mathematical algorithms. It is capable of capturing humans and other gestures without any physical touching. There has been a huge number of research and contributions conducted to compile sign language. Still there are some difficulties for the scientific contributors especially in the area of gesture recognition system testing. The vision-based gesture recogni- tion system works among computers and humans. So there is a need for high performance and accuracy [1]. A stan- dard testing framework needed to determined accuracy and identify the defects of the gesture recognition system by a proper testing . The problems are , how can we design an automation framework to test a gesture recognition system? A test automation framework is a standard or a set of processes to test a system, intended to identify bugs and defects to ensure the software quality. We contributed a gesture recognition testing framework that works automatically to test any gesture recognition system.

In the software industry, overall half of the software development cost is spent on testing [2]. Due to the standard tool and process, the testing cost and time will increase. We have developed a framework using dynamic image pattern generation with augmentation to establish a reliable system with correct testing result for which software use to recognition gesture activities. In this time every image processing system playing key role. Already the market has several system that work using gesture recognition technology, and also there are many software testing tools. But there is a scarcity for software testing tools dedicated only to gesture recognition system. A well-organized testing framework is important to test a system properly. Therefore, we proposed a standardized testing tool to test any gesture recognition system in a proper way.

To satisfy the system requirements, every organization has to follow a correct and exact software testing standard. In the area of software testing, standard means a benchmark to test software to identify the bug, defect, and failure as well as fulfill the system requirement to release quality software. Some popular ISO standard which is used for software testing. In the beginning, we use a standard as a software testing process. The standard name ISO/IEC/IEEE/291129-3 and ISO/IEC/IEEE/291129-5 is using to maintain the software testing techniques. This standard is the benchmark of properly maintain testing (process, techniques, design, maintenance and implementation). The traditional gesture recognition system captures the image sample manually. We indicate the point because it's a very time-consuming process. While gathered images for training the system. In the present its easy to gathering the training pictures increase, also it is cost effective. Therefore, we planing for generate training pictures automatically to test the gesture recognition system to minimize time, and the analysis expenditure. As well as to pup the efficiency of gesture recognition software.

These are some major facts that the majority of the gesture recognition system takes sample training, and testing images manually, not following a fixed standard to test their system. As the total process is manual; this process takes much time. As well as there haven't any fixed procedures to create the sample (training and testing )images [3]–[11]. Therefore every system needs to the automated process where the gesture recognition system uses a manual approach to testing their system. After analyzing a few gesture recognition systems we found some problems and identified some solutions [3]–[7] scrutinizing the experimental result. We presented a dynamic process for test a system, especially to image processing soft- ware testing.The research team developed a model based on the common parameters for converting the sample image with different testing cases with the parameters. Here every system is different and their working procedure is not common. We experimented five specific well known gesture recognition system to know how the systems work, how gesture images are taken, what are the conditions or bugs that are not considered in the system. The first system is proposed Mr.Akoum [3] to identification gesture activities for hand sign system, where extracts attributes to images to create a signature by identifying the key points. Jarman et.al. [4] provides a fingerprint finder algorithm that automatically recognizes Bengali sign language by using a multilayer feed-forward neural network. This system's average accuracy is 88.69%. In there experiment, they considered resized image and advised to scale the sample images.

M.A.Rahman et.al. [5] proposed a system for identification sign language for Bangle applying real-time methodologies based on the computer vision. They used KNN classifiers for sign language recognition. They applied a manual approach to generate numbers of images using a test sample by following several common parameters, for example, scaling and size as well as using contrast and rotation to capture the pictures.Here the existing system took sample pictures to trainup the system by the number 600(10x10x6) photos and 3000(10x10x30) differently for thaking vowels and consonant of Bangla, to traing and testing. In the reffered system developed by Nikam et.al. [6] for the people dumb deaf by completely using hand sign system for recognition the gesture to communicate with other people The system recognized images 30 per second as a frame rate to recognize a portion. The system identifies a single segment using the techniques of image processing. The system applied some general parameters such as reflection, size, and resolution to operate the testing images. Uddin et.al. [7] was developed the system to facilitate dump and deaf people to communicate with others by recognizes the Bengali alphabet hand sign using a support vector machine. The system takes sampling objects as simple images to recognized the alphabetic to extract the features. They create testing and training images by implementing a few parameters like contrast, size, resolution, and rotation. Their systems final accuracy is 95.9% by applying the provided method.

On the basis of our study of these gesture recognition systems, We developed a software testing method by dynamically generated sample test images which are mandatory for testing any gesture recognition system's accuracy [3]–[7]. From our research, we detected a few required parameters that helped to produce sample images. We used five common parameters in our model as like features what is the background of the sample image, how much noise in the sample image, what is the size or scale of the gesture in the image, how much contrasted image is considered, what will happen if different rotated images are provided. These parameters are used in our framework to generate different testing cases to test a gesture recognition system to ensure efficiency. It will help to judge a system by providing proper results within the minimum time and computational cost. Already there are a number of software testing standards in the software industries. But there is no standardized testing method or framework to test gesture recognition systems. We provided a gesture recognition system testing framework by following the ISO/IEC /IEEE 291129 - 2013 testing process, and ISO/IEC /IEEE 291129 -2015 testing techniques. Already there are many standards available but these rules aren't possible to follow in a tool. So far we just took two standards for a testing plan, test execution, test monitoring, functional non-functional testing, and bug management. We studied Kasurinen et.al. [12] where they developed a process and strategies for software testing. There they have discussed the best software testing process, testing tool, methods, and test cases, to release quality software.

S.M. Hasan et.al [13]. provided the similar content of gesture recognition system testing we studied the paper and proposed this paper of an automated testing framework for gesture recognition system using dynamic image pattern generation with augmentation based on the existing one. In the following paper, our developed framework is divided into three sections. In section II we discussed the methodology of our proposed framework, working procedure of converting the input for generating different kinds of test cases by the five phases where the system tests the images with the existing system. Section III is especially for system testing. In the section we test the existing system using different test cases and provided an performance analysis. Section IV is used to summarized the output that depicts how many errors and defects in the system and how much accuracy is achieved finally.

## II. Proposed Methodology

We proposed an automated testing framework using dynamic image pattern generation with augmentation. The design of the proposed framework is depicted with the Fig.1. the proposed method, we considered image rotation, scaling, applied different background, noise, contrast in the sample image. Applying these five crucial parameter, we created many numerous sample images from a single test image. These dynamically generated image pattern are used to test any gesture recognition system. Finally, we compare our experimental result with the existing systems result. If we find similar result, we can say there is no flaw in the system. On the other hand, if there is a huge variation in the result, there might be some crucial features which are not considered in the system.

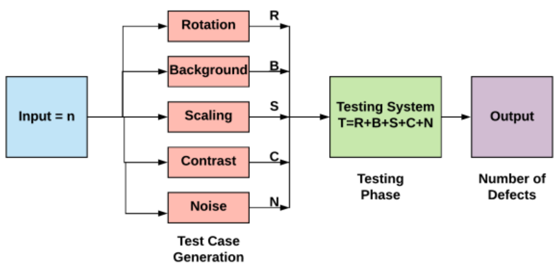



Fig. 1: We designed the proposed dynamic testing model to using dynamic image pattern generation

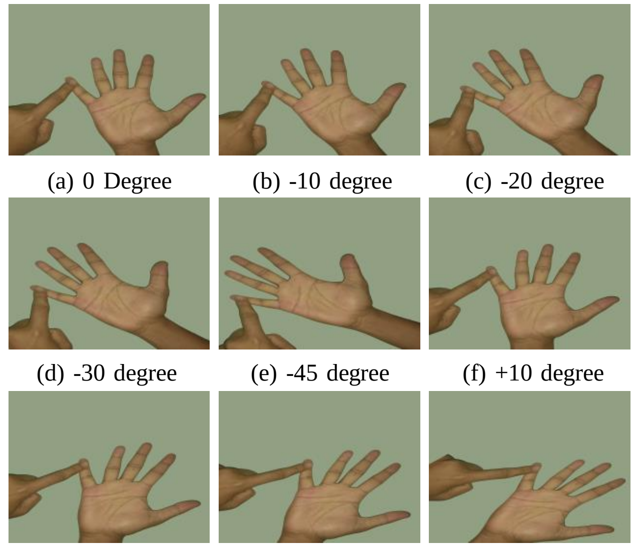
In the Fig.1, In our methodology we divided the proposed framework features with four parts, every part called as a phase itself. Every single phase of the framework used for taking those images used for testing. We use the images to test the system which is already in the market. The working procedure of our system from the first phase. Our system accepts the input images as a sample. We have pointed out that with a different degree then try to match with the real image. And also the same process applying for background, scaling, contrast and noise. Which are directly used to taking the sample image from the existing system. And convert the images with the phases of different parameters. The framework working like take the input then transfer that with five parameters R, B, S, C and N then testing the images and finally calculate the accuracy by (T=R+B+S+C+N) then give the result based on accuracy and defect of the existing system.

### *A. Data Use For The Proposed Model as Input*

To test any gesture recognition system, the system generally takes captures of the any human gesture. This gestures are generally a single image, or a video which we may consider a series of images in a time frame. So, we can consider n number of images are the inputs for the system.

### *B. Dynamic Test Case Generation*

In the next phase of our framework after taking inputs, we applied some significant factors. Rotating images, various scaled images, applying different effects like noise, background, contrast - these are factors which are needed to consider for taking sample input images for any gestures recognition system. Our framework applied these factors to



(a) 0 Degree (b) -10 degree (c) -20 degree
(d) -30 degree (e) -45 degree (f) +10 degree
(g) +20 degree (h) +30 degree (i) +45 degree

Fig. 2: Dynamic Test Cases generation applying rotating factors to a single image.

a single image and produce multiple augmented images dynamically. A brief description of the factors and how these factors are provided dynamically are given below:

1) Rotating Images:: In real life, when gesture are provided in any recognition system, they not same as inputs which are taken for experiments. We need to consider different angled or rotated images because object may be different angled position compare to the camera position. In our system, we applied some rotating factors so that we can have several sample images having different rotating angles from a single input image.

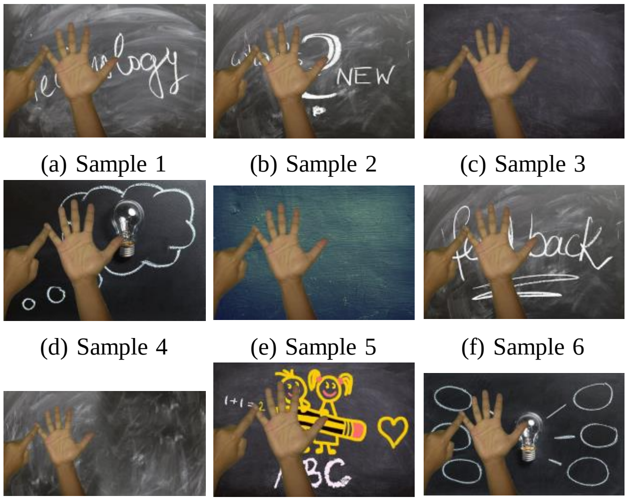
**Algorithm 1:** Create Rotation Samples

**Data:** Image I, Limit MAX, Rotating Factor $\Delta\Theta$
**Result:** 2*(MAX/$\Delta\Theta$) set of Rotated Images R

```
Procedure:
R ← φ
i = ΔΘ
while i ≤ MAX do
    R ← R ∪ Rotate I to ±i Degree
    i = i + ΔΘ
end
Return: Return R
```

Algorithm1 showing this developed data structural formula for implement the image rotation real life model. The Fig.2 is showing the there we have use different pictures with changes the rotations by degree. When creating the sample image the main picture has rotating by 10 degree both left and right up to 45, we can use the process as per our needs of training images.

2) Providing Different Backgrounds:: It is crucial to consider checking whether the changing background of a sample affects the accuracy of the system or not. Any system's robustness must be tested with sample images having dif-



(a) Sample 1 (b) Sample 2 (c) Sample 3

(d) Sample 4 (e) Sample 5 (f) Sample 6

(g) Sample 7 (h) Sample 8 (i) Sample 9

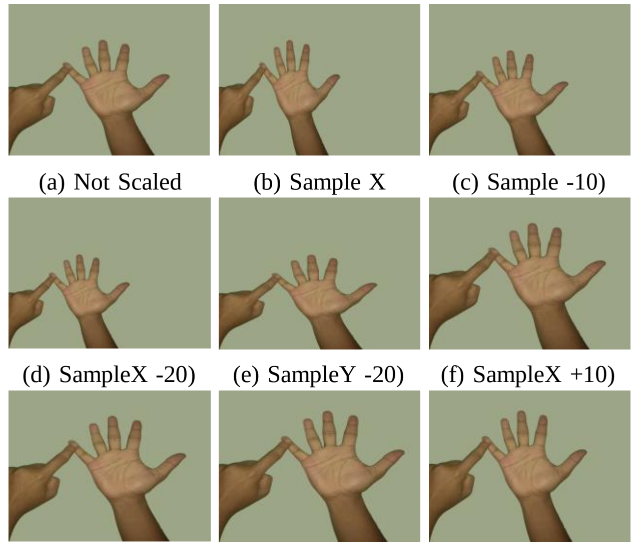
Fig. 3: Dynamic test cases generation applying different backgrounds.

ferent backgrounds. In our research, we detect the system background of the sample images, then created by attaching different backgrounds on single images dynamically. The proposed system can create numerous training pictures with various colours to compaired with background as shown in Fig.3. Algorithm2 presents the proposed algorithm for adding different backgrounds.

```
Algorithm 2: Creating Different Background Samples
  Data: Image I, a Group of pictures for various
        Background N
  Result: a set of output Images B
Procedure:
B ← ϕ
e = Extract object from Image I
for each image i ∈ N do
  | B ← B ∪ Insert extracted image e on i
end
Return: Return B
```

*3) Resizing Objects::* It is important to test the gesture recognition system with different scaled images. System may not perform well if image are captured from either too closely or too far. Keeping this factor in mind, we applied dynamic method which will capture the object, then scale the object with different heights and widths. In Fig.4, we provided some sample images where enlargement was done by changing the height and width of the object.Algorithm3 presents the proposed algorithm for creating different scaled images.

*4) Providing Different Contrast:* Our next crucial parameter is applying different contrast on single image by applying varied color and brightness. A resilient gesture recognition system should be invariant to varied contrast level. We provided different contrast level dynamically on single image to produce different test cases to test any system. Some sample images are



(a) Not Scaled (b) Sample X (c) Sample -10)

(d) SampleX -20) (e) SampleY -20) (f) SampleX +10)

(g) SPXY +10) (h) SPXY +20) (i) SPY +20)

Fig. 4: Dynamic test cases generation applying different scaling factors.

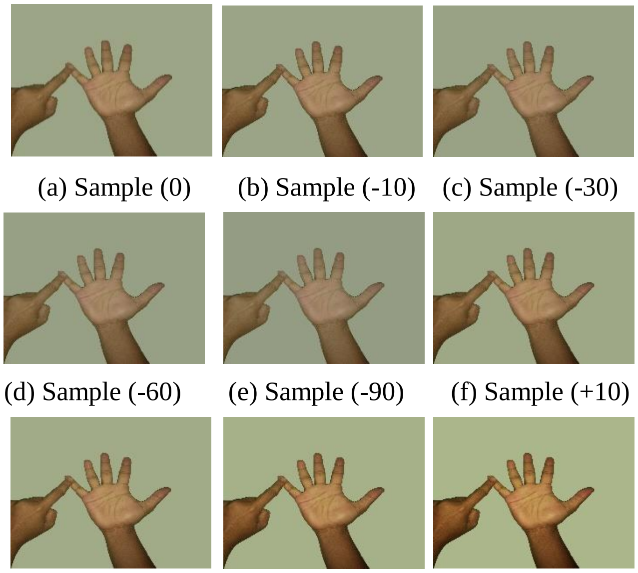
```
Algorithm 3: Create Scaling Samples
  Data: Image I, Limit MAX, Scaling Factor Δs
  Result: 2*3*(MAX/Δs) set of Rotated Images S
Procedure:
S ← ϕ
i = Δs
while i ≤ MAX do
  | e = Extract object from Image I
  | b = Extract background from Image I
  | x = scale the object e by ±i factor in X axis
  | y = scale the object e by ±i factor in Y axis
  | xy = scale the object e by ±i factor in both X and
  |   Y axis
  | S ← S ∪ Place the object x to image b
  | i = i + Δs
end
Return: Return S
```

provided in Fig.5. Algorithm4 presents the proposed algorithm for creating different contrasted images.

*5) Providing Different Noise:* In our research, we identified that in real life the image may have different noise in the process of taking pictures. The framework obvious unchanging to various steps with the noised images. In the developed methodology, we dynamically applied different level of well known noise, such as Gaussian noise to each image. In Fig.6, we showed that several test cases are generated from a single image by providing different level of Gaussian noise. Algorithm5 presents the proposed algorithm for creating different noised images.

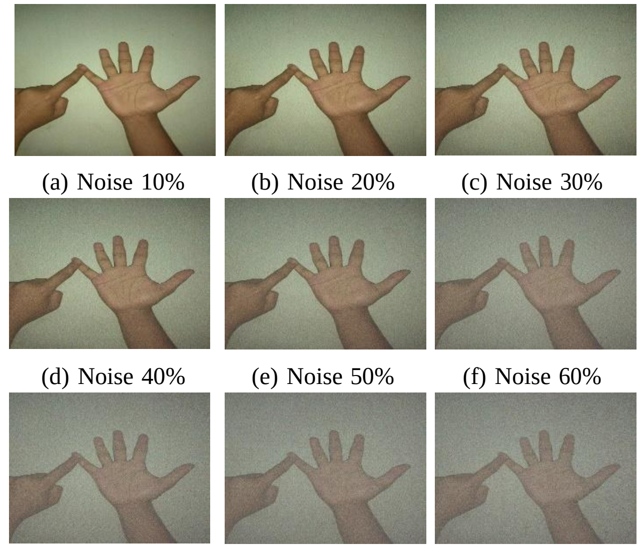
### *C. Methodology of the Framework*

We proposed an algorithm 6 to implement our framework to test any gesture recognition system. First of all, we need



(a) Sample (0) (b) Sample (-10) (c) Sample (-30)
(d) Sample (-60) (e) Sample (-90) (f) Sample (+10)
(g) Sample (+30) (h) Sample (+60) (i) Sample +9(0

Fig. 5: Dynamic test cases generation applying different contrasting factors.



(a) Noise 10% (b) Noise 20% (c) Noise 30%
(d) Noise 40% (e) Noise 50% (f) Noise 60%
(g) N.sample 70% (h) N.sample 80% (i) N.sample 90%

Fig. 6: Dynamic test cases generation applying different noise filters.

**Algorithm 4:** Creating Different Contrast Samples

**Data:** Image I, Limit MAX, Contrast Factor Δc
**Result:** 2*(MAX/Δc) set of Different Contrast Images C
**Procedure:**
$C \leftarrow \phi$
$i = \Delta$c
**while** $i \leq MAX$ **do**
  C ← C ∪ Increase contrast of image I by i factor
  C ← C ∪ Decrease contrast of image I by i factor
  $i = i + \Delta\Theta$
**end**
**Return:** Return C

**Algorithm 5:** Creating Different Noised Samples

**Data:** Image I, Limit MAX, Noise Factor Δn
**Result:** MAX/Δn set of Different Noised Images N
**Procedure:**
$N \leftarrow \phi$
$i = \Delta n$
**while** $i \leq MAX$ **do**
  N ←N ∪ Increase the noise of Image I by i factor
  $i = i + \Delta n$
**end**
**Return:** Return N

an existing model M and set of sample images X. For each image in the set X, we call our previous five algorithms 1, 2, 3, 4, and 5 to create differently rotated, backgrounds, scaled, contrasted, and noised images. Then these images are provided as test cases in the model M. Accuracy for each type of the images are calculated. Finally, total accuracy is calculated by averaging the accuracy result of the five categories.

## III. Proposed System Testing Data Analysis

### A. Experimental Setup

For develop our suggested model, integrated Canon Eos 4000D for capturing input images. We used laptop ASUS TUF FX504 and 8 gigabyte Random access memory (RAM) with the Intel’s m7 core processor for implementing the system. We also use here C of Microsoft®Visual Studio®2010 and Open CV wrapper There has been also used windows operating system, and Microsoft® office 10 for the other work.

According to our developed system, we creating testing cases based on the existing gesture recognition system analysis, the data shown Table I. There we indicate five parameters to generate all test cases. The Table II, Showing here the complete result total accuracy and defects, compare with the experimental system.

As the final result we are showing the complete testing result regarding all particular system. We use two result parameters accuracy = how many images found correctly and defect = how many are not found.

### *B. Experimental Result*

In our whole working process first we take the input from existing gesture recognition system. Then the second phase passing the sample images through our system’s five parameters and convert many variations images. We called it testing cases, a complete report of the process showing in the Table I. After that the third phase testing the all existing

**Algorithm 6:** Proposed Algorithm to Test any Recognition System.

**Data:** Images X, Testing Model M
**Result:** Total Accuracy Ac for the Images in X tested on Model M

**Procedure:**
*R, B, S, C, N* ← **M**
**for** *each x image in X* **do**
  *R* ← *R* ∪ Create$_R$*otated*$_S$*amplesforx*
  *B* ← *B* ∪ Create$_D$*ifferent* − *Background*$_S$*amplesforx*
  *S* ← *S* ∪ Create$_S$*caling*$_S$*amplesforx*
  *C* ← *C* ∪ Create$_C$*ontrast*$_S$*amplesforx*
  *N* ← *N* ∪ Create$_N$*oised*$_S$*amplesforx*
**end**
**for** *each sample in R* **do**
  Find accuracy $R_{ac}$ using Model M
**end**
**for** *each sample in B* **do**
  Find accuracy $B_{ac}$ using Model M
**end**
**for** *each sample in S* **do**
  Find accuracy $S_{ac}$ using Model M
**end**
**for** *each sample in C* **do**
  Find accuracy $C_{ac}$ using Model M
**end**
**for** *each sample in N* **do**
  Find accuracy $N_{ac}$ using Model M
**end**
$Ac = (R_{ac} + B_{ac} + S_{ac} + C_{ac} + N_{ac})/5$
**Return:** Return Ac

system with those pictures. Just see the how many images can correctly identify the existing image processing system. Finally we calculate the result by accuracy and defect. The Table II showing the complete report of the testing results.

### C. Test Case-1 Rotation

In this section, rotation is a testing case of our proposed framework which is implemented using Algorithm 1 . When an image input in the proposed framework then rotated the image with some degree. We generated training and testing image through providing different rotated factors to the image. Suppose we want to generate ninety testing sample, then we should rotate the input image as 0 to 45 degree, we use there a difference 5 of on image to another image conversion, that is work in both side positive side forty-five and the negative forty-five with plus-minus. Then, we can get the required sample images.

In the graph in the Fig.7 a comparative analysis of each system's accuracy for rotated images has shown. In the graph x-axis showing the number of sample and the y-axis showing the existing system accuracy according to the rotation testing cases. We implemented rotation testing cases with the systems

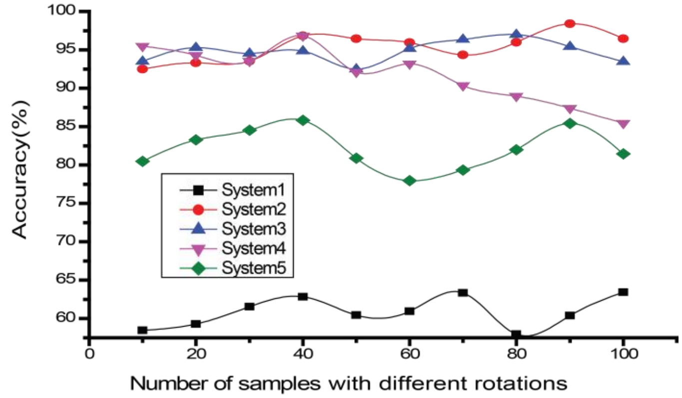



Fig. 7: Comparative analysis of each system's accuracy for rotated images

[5], [14]–[17] for testing their system. From the graph we identified that system [17] works poorly for rotated images.

### D. Test Case-2 Contrast

In this work, contrast is a testing case of our proposed framework which is implemented using Algorithm 4 . When an image input in the proposed framework then converted the image with different contrast. We generated training and testing image through providing different contrast to the image. The contrast works as changing the RGB colour combination.

Suppose we want to generate ninety testing sample, then we should change the colour the input image increases the colour 0 to 45, difference 5 the positive side forty-five and the negative side forty-five with plus-minus then we can get the required sample. this sample for the input, there has one more sample in an existing system . If we got 100 converted images from the input, then the concept is clear getting lots of images from a sample image.

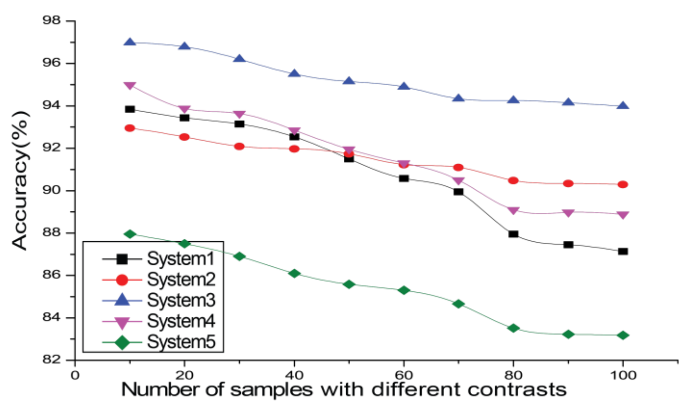



Fig. 8: Comparative analysis of each system's accuracy for contrasted images

In the graph in the Fig.8 a comparative analysis of each system's accuracy for contrasted images has shown. In the graph x-axis showing the number of sample and the y-axis showing the existing system accuracy according to the

TABLE I: EXPERIMENTAL TESTING CASES AMONG OUR DEVELOPED FRAMEWORK AND OTHERS EXISTING SOFTWARE

| Method | T. N. O. (p) | Test samples for Proposed Framework | | | | | | | E.M | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Samples per Entity | | | | | | T. T. I. (p*q) | S. P. E. (j) | N. C. (k) | T. T. I. (j*k) |
| | | Rot | Con | Sca | Bac | Noi | Tot (q) | | | | |
| Rahaman et al. [5] | 36 | 100 | 100 | 100 | 100 | 100 | 500 | 18000 | 100 | 36 | 3600 |
| Rahaman et al. [14] | 10 | 100 | 100 | 100 | 100 | 100 | 500 | 5000 | 100 | 10 | 1000 |
| Rahaman et al. [15] | 46 | 100 | 100 | 100 | 100 | 100 | 500 | 23000 | 100 | 46 | 4600 |
| Rahaman et al. [16] | 52 | 100 | 100 | 100 | 100 | 100 | 500 | 26000 | 100 | 52 | 5200 |
| Ahmed et al. [17] | 10 | 100 | 100 | 100 | 100 | 100 | 500 | 5000 | 100 | 10 | 1000 |

NOTE:- T.N.O= Total Number of Images, T.T.I=Total Test Images, E.M= Existing Method, N.C= Number of Classes, Images,Tot= Total sample, Rot=Rotation, Con= Contrast, Sca= Scaling, Noi= Noise

TABLE II: EXPERIMENTAL RESULT OF THIS FRAMEWORKS AUTO GENERATED TEST CASES BY OUR PROPOSED TESTING FRAMEWORK AND EXISTING METHOD

| Method | Measure Accuracy With Developed Framework (%) | | | | | | | E.M |
|---|---|---|---|---|---|---|---|---|
| | Rot | Con | Sca | Bac | Noi | mean | | M.A (%) |
| Rahaman et al. [5] | 60.45 | 90.58 | 65.05 | 80.56 | 90.3 | 77.388 | | 96.46 |
| Rahaman et al. [14] | 96.45 | 90.66 | 94..85 | 30.24 | 50.8 | 67.0375 | | 95.85 |
| Rahaman et al. [15] | 92.5 | 94.9 | 95.23 | 91.73 | 90.89 | 93.05 | | 95.67 |
| Rahaman et al. [16] | 92.1 | 93.34 | 96.13 | 92.56 | 92.6 | 93.346 | | 95.83 |
| Ahmed et al. [17] | 80.9 | 85.3 | 90.45 | 75.6 | 65.8 | 79.61 | | 92.00 |

NOTE: Rot= ROTATION, Con= CONTRAST, Sca= SCALING, Bac= BACKGROUND, Noi= NOISE, E.M= EXISTING METHOD, M.A= MEAN ACCURACY

contrasted testing cases. We implemented contrasted testing cases with the systems [5], [14]–[17] for testing their system. From the graph we identified that system [17] works poorly for contrasted images.

*E. Test Case-3 Scaling*

In this work, scaling is a testing case in Fig.1 as shown of our proposed framework. There is use the scaling as a testing case to test the existing system. When an image input in the proposed framework then converted the image with the scaling according to proposed **Algorithm 3.** The scaling work as changing the input with changing the x and y-axis pixels when x-axis change that's time y-axis will constant. when y-axis changing that s time x-axis will constant.

In the graph in the Fig.9 a comparative analysis of each system's accuracy for different scaled images has shown. In the graph x-axis showing the number of sample and the y-axis showing the existing system accuracy according to the scaled testing cases. We implemented scaled testing cases with the systems [5], [14]–[17] for testing their system. From the graph we identified that system [5] works poorly for scaled images.

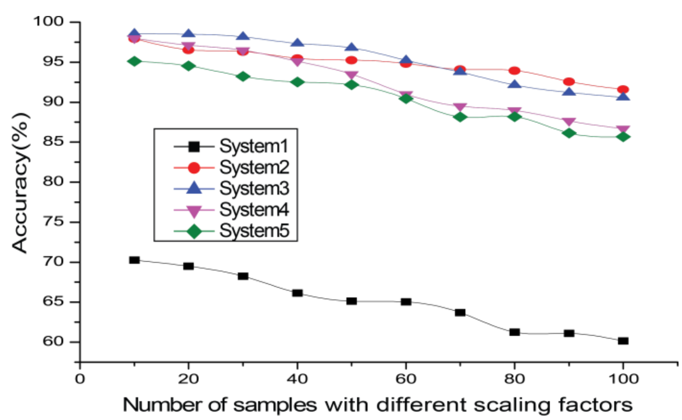



Fig. 9: Comparative analysis of each system's accuracy for scaling images

*F. Test Case-4 Background*

In Fig.3 has shown the background images use as a testing case to test the existing system [5], [17]. When an image input in the proposed framework then converted the image with different background. To generate training and testing image through different background to the image, the background is removed from the input image background and adding a new background with a new image. There continues a loop that removes background and adding a background with the new background less image.

In the graph in the Fig.10 a comparative analysis of each system's accuracy for different background images has shown. In the graph x-axis showing the number of sample and the y-axis showing the existing system accuracy according to the testing cases. We implemented different background testing cases with the systems [5], [14]–[17] for testing their system. From the graph we identified that system [14] works poorly for different background images.

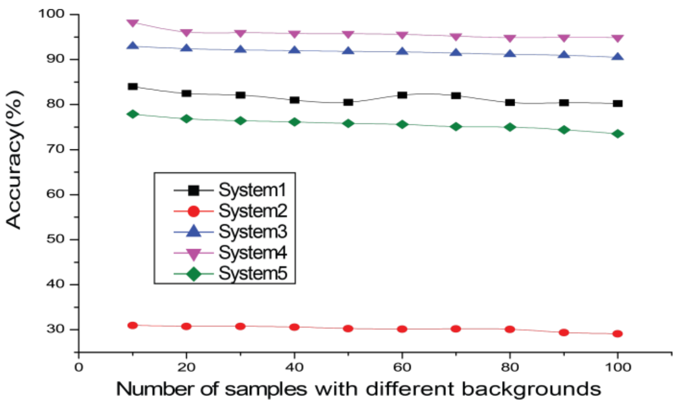



Fig. 10: Comparative analysis of each system's accuracy for different background images

### G. Test Case-5 Noise

In this work, the Noise is used as a testing case framework Fig.1 as shown of our proposed. When an image input in the proposed framework then converted the image with Noise. In this testing case, the main goal is adding new noise being the input. To generate training and testing image through the Noise to the image, a new noise with the main object with the input image. There continues a loop that a process adding noise with the new input image, according to the requirements of the software. It simply adding different kinds of noise with the input images. This is a process to creating different types of sample image according to the input images.

In the graph in the Fig.11 a comparative analysis of each system's accuracy for different noisy images has shown. In the graph x-axis showing the number of sample and the y-axis showing the existing system accuracy according to the testing cases. We implemented noisy testing cases with the systems [5], [14]–[17] for testing their system. From the graph we identified that system [14] works poorly for different noisy images.

### H. Result of Proposed System

The Table II has shown the result of the proposed system and existing system. The existing system A.R. [5] used 3600 Training photos to testing their software. based on the existing system's experiments, their system achieved an accuracy of

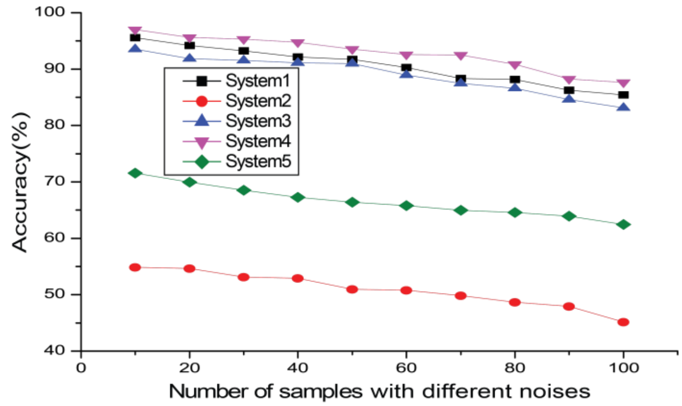



Fig. 11: Comparative analysis of each system's accuracy for noised images

95.32%, Where the system we developed identified the existing system's total accuracy is 77.388%. Their system perform poorly for the differently rotated and scaled images.

Some of the system's accuracy of us matched with existing system accuracy. But there are some results that are not similar to the proposed system's result. This is because, some systems do not consider all the crucial parameter for testing their gesture recognition system.

## IV. CONCLUSION

We developed a complete gesture recognition system testing framework in our research. In the framework, Our proposed five testing cases phase to dynamically generate sample test cases. We applied our framework on five existing gesture recognition systems. Some of the systems accuracy do not matched with our experimental results. We identified that their systems do not consider some of the crucial factors which are important in gesture recognition system. Thus our framework is not only useful to identify accuracy and defects of any image processing system but also huge amount training pictures automatically generated, the process decrees time and expenditure.

## ACKNOWLEDGMENT

This work was supported in part by the Center for Research, Innovation, and Transformation (CRIT) of Green University of Bangladesh (GUB) under grant No. 3/2019 (CRIT, GUB).

## REFERENCES

[1] T.-H. Tsai, C.-C. Huang, and K.-L. Zhang, "Design of hand gesture recognition system for human-computer interaction," Multimedia Tools and Applications, vol. 79, no. 9, pp. 5989–6007, 2020.
[2] B. Boehm and V. R. Basili, "Software defect reduction top 10 list," SOFTWARE MANAGEMENT.
[3] A. Akoum and N. Al Mawla, "Hand gesture recognition approach for asl language using hand extraction algorithm," Journal of Software Engineering and Applications, vol. 8, no. 08, p. 419, 2015.
[4] A. Jarman, S. Arshad, N. Alam, and M. J. Islam, "An automated bengali sign language recognition system based on fingertip finder algorithm," International Journal of Electronics and Informatics, vol. 4, 07 2015.
[5] M. A. Rahaman, M. Jasim, M. H. Ali, and M. Hasanuzzaman, "Real-time computer vision-based bengali sign language recognition," in 2014 17th (ICCIT). IEEE, 2014, pp. 192–197.

[6] A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," in 2016 Online International Conference on Green Engineering and Technologies (IC-GET). IEEE, 2016, pp. 1–5.
[7] M. A. Uddin and S. A. Chowdhury, "Hand sign language recognition for bangla alphabet using support vector machine," in 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET). IEEE, 2016, pp. 1–4.
[8] S. Murugesan, "Attitude towards testing: a key contributor to software quality," in Proceedings of 1994 1st International Conference on Software Testing, Reliability and Quality Assurance (STRQA'94), Dec 1994, pp. 111–115.
[9] M. S. Islalm, M. M. Rahman, M. H. Rahman, M. Arifuzzaman, R. Sassi, and M. Aktaruzzaman, "[ieee 2019 international conference on innovation and intelligence for informatics, computing, and technologies (3ict) - sakhier, bahrain (2019.9.22-2019.9.23)]," 2019. [Online]. Available: http://gen.lib.rus.ec/scimag/index.php?s=10.1109/3ICT.2019.8910301
[10] T. Tabassum, I. Mahmud, M. P. Uddin, E. Ali, M. I. Afjal, and A. M. Nitu, "Enhancement of single-handed bengali sign language recognition based on hog features," Journal of Theoretical and Applied Information Technology, vol. 98, no. 05, pp. 743–756, 2020.
[11] Z. Lu, S. Qin, X. Li, L. Li, and D. Zhang, "One-shot learning hand gesture recognition based on modified 3d convolutional neural networks," Machine Vision and Applications, vol. 30, 8 2019. [Online]. Available: http://gen.lib.rus.ec/scimag/index.php?s=10.1007/ s00138-019-01043-7
[12] J. Kasurinen, "Elaborating software test processes and strategies," in 2010 Third International Conference on Software Testing, Verification and Validation, April 2010, pp. 355–358.
[13] S. M. Hasan, M. S. Islam, M. Ashaduzzaman, and M. A. Rahaman, "Automated software testing cases generation framework to ensure the efficiency of the gesture recognition systems," in 2019 22nd International Conference on Computer and Information Technology (ICCIT. IEEE, 2019, pp. 1–6.
[14] M. A. Rahaman, M. Jasim, T. Zhang, M. H. Ali, and M. Hasanuzzaman, "Real-time bengali and chinese numeral signs recognition using contour matching," in 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2015, pp. 1215–1220.
[15] M. Rahaman, M. Jasim, M. Ali, T. Zhang, and M. Hasanuzzaman, "A real-time hand-signs segmentation and classification system using fuzzy rule based rgb model and grid-pattern analysis," Frontiers of Computer Science, vol. 12, 11 2018.
[16] M. A. Rahaman, M. Jasim, M. H. Ali, and M. Hasanuzzaman, "Bangla language modeling algorithm for automatic recognition of hand-sign-spelled bangla sign language," Fron. Comput. Sci., vol. 14, no. 3, pp. 143 302–, 2020.
[17] S. Ahmed, M. Islam, J. Hassan, M. U. Ahmed, B. J. Ferdosi, S. Saha, M. Shopon et al., "Hand sign to bangla speech: A deep learning in vision based system for recognizing hand sign digits and generating bangla speech," arXiv preprint arXiv:1901.05613, 2019.

Md. Ashaduzzaman received his B.Sc. Engineering degree in Computer Science and Engineering (CSE) from University of Dhaka, in 2015. He is currently working as a Sr. Lecturer of CSE Department in Green University of Bangladesh. He is also performing duty as a program coordinator. He is a member of the Artificial Intelligence and Machine Learning Research Group of Green University of Bangladesh. His research interests include Data Mining, Ma- chine Learning, Artificial Intelligence, Blockchain and Natural Language Processing.

Sheikh Monirul Hasan born in Faridpur, Dhaka, Bangladesh in 1995. He completed a B.Sc. in engineering degree from the Department of Computer Science and Engineering, the Green University of Bangladesh in 2019, successfully. Previously, He completed a Diploma in Engineering Degree from the Department of CSE, Faridpur Polytechnic Institute in 2015. He is a Digital Marketing Engineer popularly known as a professional Digital Marketing Consultant and SEO Expert in Bangladesh. He is certified by Bangladesh Hi-Tech Park Authority, as a Software Quality Assurance Engineer. Also, he has achieved Google Digital Marketing Certification. He started his career as a Software Quality Assurance and Test Engineer at a software company in Bangladesh. Currently, he professionally serving SEO, Digital Marketing, and software testing services through his website www.sheikhmonirulhasan.com, His research interests are software engineering, Software Quality Assurance, and Testing, Machine Learning, Image processing, Robotics, Big Data Analysis, Computer Vision, Artificial Intelligence, Digital Marketing and Search Engine Optimization (SEO)

Md. Saiful Islam completed his B.Sc. in engineering degree from the Department of Computer science and Engineering, the Green University of Bangladesh in 2019. Currently, he working as a software support engineer in the freelancing marketplace freelancer.com. And working with various software companies as a technical support executive. His interest's research areas are software engineering, wireless networking, and image processing.

Muhammad Aminur Rahaman (M'39) received his PhD degree from the Department of Computer Science and Engineering, University of Dhaka, Bangladesh in 2018. He completed his B.Sc. and M.Sc. degree from the Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh in 2003 and 2004, respectively. He is the founder Director of the renowned software development company named Worldgaon (Pvt.) Ltd. established in 2008. Currently he is serving as an Assistant Professor in the Dept. of Computer Science and Engineering, Green University of Bangladesh from September 2018. He is a senior member of IEEE (SMIEEE), IEEE Computer Society (IEEE CS), and IEEE Robotics and Automation Society (IEEE RAS). He is the Branch Counselor of IEEE Student Branch GUB and IEEE Computer Society GUB Student Branch Chapter. He is the Publication Chair and TPC Co-Chair in International Conference on Sustainable Technologies for Industry 4.0 (2019, 2020). His current research interests include Computer vision, Image Processing, Robotics and Software Engineering.