**F) TEST PROCEDURES, EXPERIMENTS AND TEST RESULTS**

There are four main subsystems to be tested in solution proposed by Revolutionary Systems Inc. to “Gimme Fast” project. These are image acquisition subsystem, communication subsystem, transportation subsystem and image reconstruction subsystem. The tests are designed so that the requirements of the each subsystem can be checked whether the implemented parts are satisfying these requirements or not. The procedures that are followed in these tests along with the results are given the following subsections.

**…**

**F.2) Image Acquisition Subsystem**

**F.2.1) Test Procedure**

In the test procedure followed in the image acquisition subsystem, a photo is captured using the Raspberry Pi camera module. Upon capturing the photo, this photo is processed.

In the first part of the test, the dimensions and the quality parameter of the photo are changed and the sizes of the output photos are observed. In the second part of the test, one of the resized images is divided to 5 pieces vertically.

**F.2.2) Test Results**

In the first part of the tests, an example photo captured via Raspberry Pi camera module is seen in Figure 1. The dimensions of this photo are 2592\*1944 and its size is 2.7 MB. The dimensions of the photo are constant while its size is varying at different photos.



*Figure 1: The photo captured by the Raspberry Pi camera module.*

The dimensions of 256\*512 and 256\*256 are tested for quality values of 100%, 90%, 80% and 50%. The obtained photos are seen in Figure 2 and Figure 3. The sizes of the output photos are given in the Table 1.



*(2.a) (2.b) (2.c) (2.d)*

*Figure 2: The output photos where the dimensions are decreased to 256\*512 and quality is decreased (100% in 2.a, 90% in 2.b, 80% in 2.c and 50% in 2.d).*



*(3.a) (3.b) (3.c) (3.d)*

*Figure 3: The output photos where the dimensions are decreased to 256\*256 and quality is decreased (100% in 3.a, 90% in 3.b, 80% in 3.c and 50% in 3.d).*

*TABLE 1: Size of the Image vs Quality and Photo Dimensions*

|  |  |  |
| --- | --- | --- |
| **Quality** | **Photo Dimensions** | |
| **256\*256** | **256\*512** |
| 100 | 50.2 kB | 91.6 kB |
| 90 | 17.2 kB | 30.3 kB |
| 80 | 11.6 kB | 20.4 kB |
| 50 | 6.7 kB | 11.6 kB |

The requirement related with the image acquisition subsystem is “The system must be able to take a photo.” As seen at the tests, image acquisition subsystem is able to take photo. Furthermore, this photo is successfully resized.

At the second part of the test, photo seen in Figure 1 is vertically divided to 5 pieces. These new photos are given in Figure 4. These photos are the converted to bit arrays.



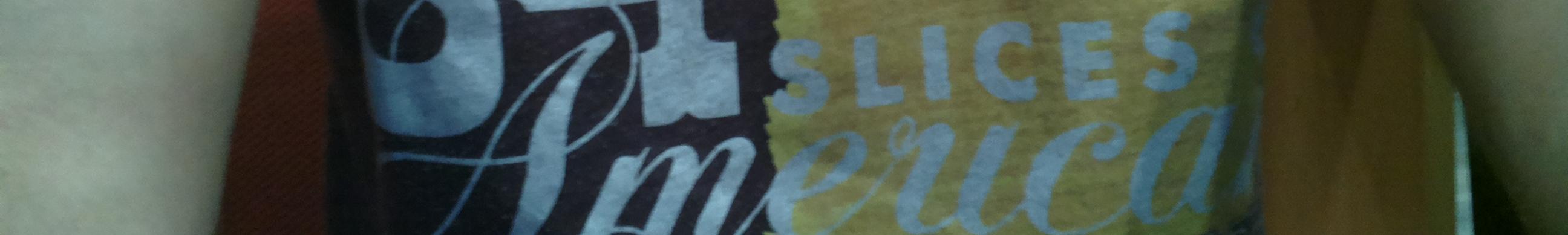
*(4.a)*



*(4.b)*



*(4.c)*



*(4.d)*



*(4.e)*

*Figure 4: The vertical slices of the photo given in Figure 1, which are obtained using Python Image Library.*

Therefore, the image acquisition subsystem is successfully divided to 5 pieces.

**F.3) Image Reconstruction Subsystem**

**F.3.1) Test Procedure**

In the test procedure followed in the image reconstruction subsystem, a photo is divided to 5 pieces vertically; these pieces are transformed to bit arrays. The bit arrays represent the data delivered by the communication system. These arrays are then used to construct photos and 5 photos are merged to construct the whole photo back and this photo is displayed at an LCD screen.

**F.3.2) Test Results**

The photos seen in the Figure 4 are converted to bit arrays. Then these bit arrays are concatenated and an image is successfully reconstructed using relevant Python tools. This photo is then displayed in an LCD screen. The requirement related with the image reconstruction subsystem is “As the full photo is delivered, the photo must be displayed at the receiver terminal.” Although the data used to reconstruct a photo was not delivered by communication subsystem, it is seen that image reconstruction subsystem is able to construct an image from the inputted bit arrays and display it.