

DLIP : LAB1

Author : HanMinung

Student ID : 21800773

School of Mechanical and Control Engineering

LAB : Grayscale and segmentation

Date : 2023.03.24

DLIP : LAB1

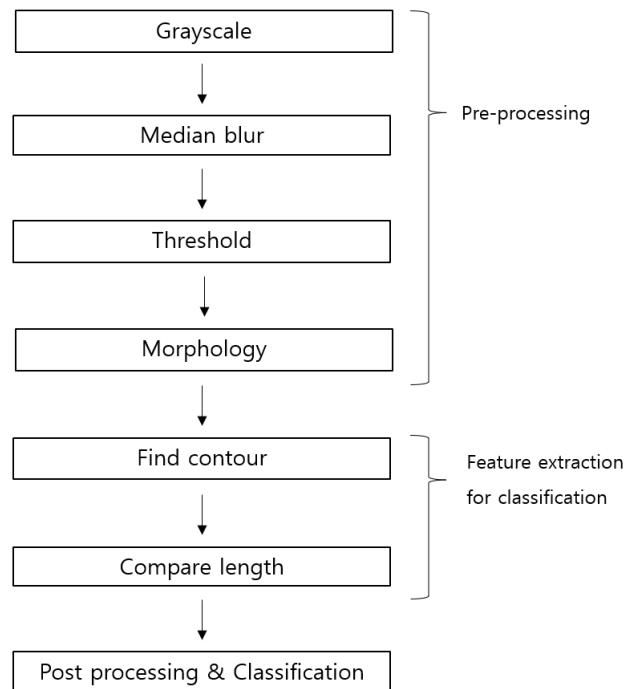
- 1. Purpose
- 2. Flow chart
- 3. Processing
 - 3.1. Preprocessing
 - 3.2. Processing
- 4. Classification
- 5. Result
- 6. Discussion and analysis

1. Purpose

- The purpose of this experiment was to develop a system that can automatically distinguish bolts and nuts of different sizes and count each type accurately using classical image processing techniques. This system can be useful in various manufacturing and assembly processes where different types of bolts and nuts are used, and their sizes need to be identified and counted quickly and accurately.
- Object to classify
 - M5 BOLT
 - M6 BOLT
 - M5 Hexa Nut
 - M6 Hexa Nut
 - M5 Rect Nut
- Source image



2. Flow chart

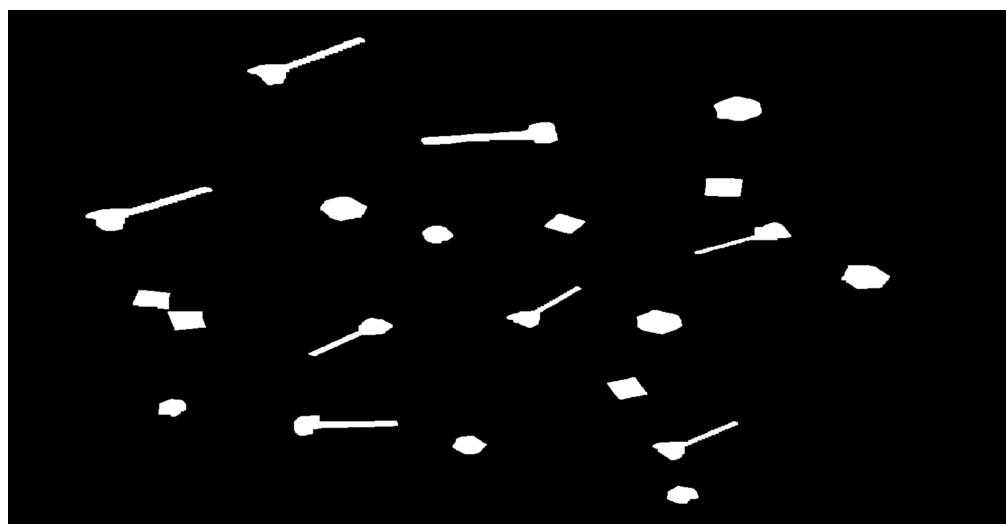
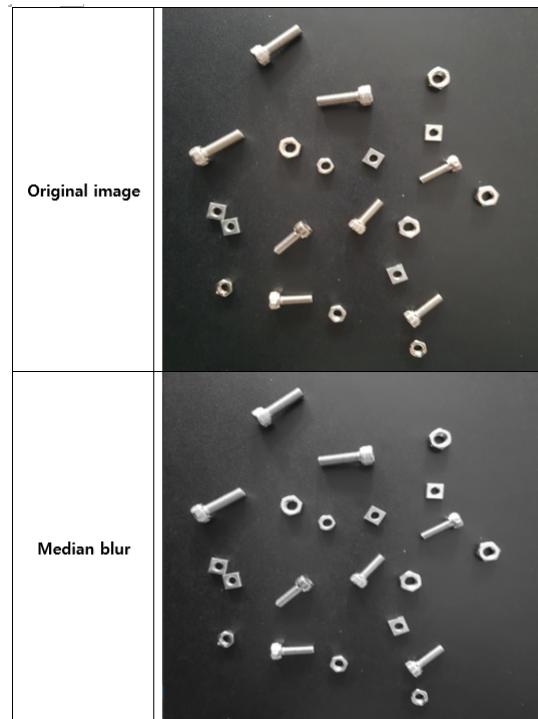


3. Processing

3.1. Preprocessing

- Before extracting features of each objects, it is proper to process given data in advance to reduce noise and unnecessary points which can return inappropriate results after we extract those features.

- First, there are several blur processing techniques to eliminate noise, but I used a Median blur with kernel size 5 in this lab. In addition, an appropriate threshold value and morphology type were determined using a track bar. Increasing the threshold value can remove all unnecessary points, but there are cases where problems occur during post-processing due to unintended disconnection of the dark part of the nut. Therefore, even if there are some unnecessary points, dilation method was applied to minimize the effect, and the remaining points were considered by giving conditions in the post-processing.
- The results of blurring image and thresholding & morphology are as follows in sequence :



In summary, the values related to preprocessing are as follows :

- Median blur with kernel size 5
- Threshold : type 0, value : 118
- Morphology : dilation with 13 times - erosion with 23 times - dilation with 2 times
 - To fill out the hole of Nut, many times of dilation process was performed.

- And then, to separate two nuts which are located in the left below of the image, 23 times of erosion was performed.

3.2. Processing

```
#define M6BOLT          0
#define M5BOLT          1
#define M6HEXNUT        2
#define M5HEXNUT        3
#define M5RECTNUT       4

// dst_morph : final result of preprocessing
findContours(dstMorph, contours, hierarchy, RETR_TREE,
CHAIN_APPROX_SIMPLE);

for (int i = 0; i < contours.size(); i++) void main() {

    src = imread("Lab_GrayScale_TestImage.jpg", IMREAD_COLOR);

    // Check image loading
    checkLoad(src);

    // image preprocessing
    preProcessing(src);

    // finding contour
    findContours(dstMorph, contours, hierarchy, RETR_TREE,
    CHAIN_APPROX_SIMPLE);

    for (int i = 0; i < contours.size(); i++) {
        double len = arcLength(contours[i], true);
        Moments mu = moments(contours[i], false);
        Point centroid = Point(mu.m10 / mu.m00 + 30, mu.m01 / mu.m00 - 50);

        // Determine type
        classType = detType(len);

        // Drawing contours
        visualizing(i, classType);
    }
}
```

```
// preprocessing
void preProcessing(Mat src) {

    // gray scale
    cvtColor(src, srcGray, CV_BGR2GRAY);
```

```

// Median blur with kernel size 5
medianBlur(srcGray, dstBlur, kernelsize);

// Threshold process
threshold(dstBlur, dstThresh, threshval, maxBinaryval, threshType);

// Morphology process
dilate(dstThresh, dstMorph, element);

dilate(dstMorph, dstMorph, element, Point(-1,-1), 13);
erode(dstMorph, dstMorph, element, Point(-1,-1), 23);
dilate(dstMorph, dstMorph, element, Point(-1, -1), 2);

}

```

```

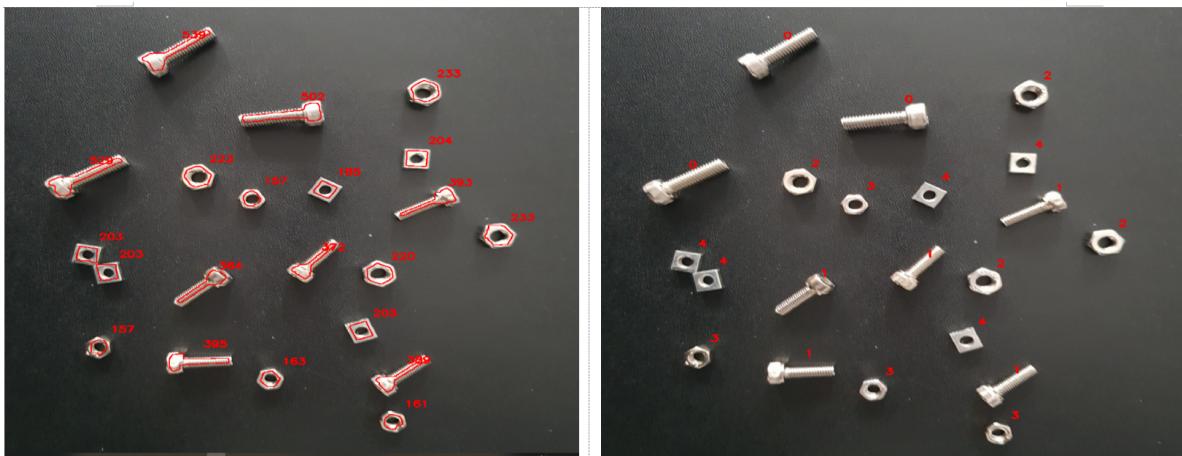
// Function to determine the type of object
int detType(int len) {

    if (len > 450) classType = M6BOLT;
    if (len < 450 && len > 330) classType = M5BOLT;
    if (len > 210 && len < 250) classType = M6HEXNUT;
    if (len > 180 && len < 210) classType = M5RECTNUT;
    if (len > 100 && len < 180) classType = M5HEXNUT;

    return classType;
}

```

As stated in the code above, I used arcLength to get first features of each objects. In the step of post-processing, classification requires many information for high accuracy operation. I used length of each contours among those informations to classify those objects. I classified objects with information of length and the following result is as follows :



4. Classification

- M6 Bolt

```

if (classType == M6BOLT) {

    cv::Rect BoundingBox = cv::boundingRect(contours[i]);
    cv::rectangle(src, BoundingBox, color1, 2);
    Moments mu = moments(contours[i], false);
    Point centroid = Point(mu.m10 / mu.m00 - 50, mu.m01 / mu.m00 - 80);
    putText(src, "M6 BOLT", centroid, FONT_HERSHEY_SIMPLEX, 1, color1,
2);

    Bolt_M6++;
}

```

- M5 Bolt

```

if (classType == M5BOLT) {

    cv::Rect BoundingBox = cv::boundingRect(contours[i]);
    cv::rectangle(src, BoundingBox, color2, 2);
    Moments mu = moments(contours[i], false);
    Point centroid = Point(mu.m10 / mu.m00 - 50, mu.m01 / mu.m00 - 80);
    putText(src, "M5 BOLT", centroid, FONT_HERSHEY_SIMPLEX, 1, color2,
2);

    Bolt_M5++;
}

```

- M6 Hexa Nut

```

if (classType == M6HEXNUT) {

    cv::Rect BoundingBox = cv::boundingRect(contours[i]);
    cv::rectangle(src, BoundingBox, color3, 2);
    Moments mu = moments(contours[i], false);
    Point centroid = Point(mu.m10 / mu.m00 - 50, mu.m01 / mu.m00 - 80);
    putText(src, "M6 Hexa Nut", centroid, FONT_HERSHEY_SIMPLEX, 1,
color3, 2);

    M6_HEXA_NUT++;
}

```

- M5 Hexa Nut

```

if (classType == M5HEXNUT) {

    cv::Rect BoundingBox = cv::boundingRect(contours[i]);
    cv::rectangle(src, BoundingBox, color5, 2);
    Moments mu = moments(contours[i], false);
    Point centroid = Point(mu.m10 / mu.m00 - 50, mu.m01 / mu.m00 - 80);
    putText(src, "M5 Hexa Nut", centroid, FONT_HERSHEY_SIMPLEX, 1,
color5, 2);

    M5_HEXA_NUT++;
}

```

- M5 Rect Nut

```

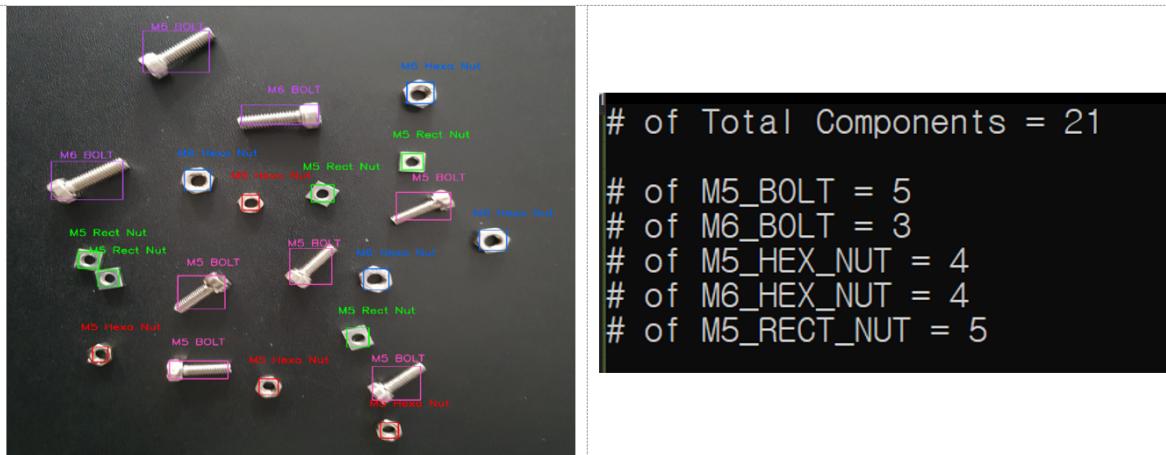
if (classType == M5RECTNUT) {

    cv::Rect BoundingBox = cv::boundingRect(contours[i]);
    cv::rectangle(src, BoundingBox, color4, 2);
    Moments mu = moments(contours[i], false);
    Point centroid = Point(mu.m10 / mu.m00 - 50, mu.m01 / mu.m00 - 80);
    putText(src, "M5 Rect Nut", centroid, FONT_HERSHEY_SIMPLEX, 1,
color4, 2);

    M5_SQUARE_NUT++;
}

```

5. Result



All objects were properly classified with the method introduced above. Since I drew rectangle with the calculated contours, the shape cannot be matched perfectly as we can see in the left figure above.

6. Discussion and analysis

By applying different methods to the objects in an image, it becomes clear how important preprocessing techniques like blur, thresholding, and morphology can be. When preprocessing is done well, it makes it easier to extract and classify important features in the image. In this lab, we encountered a challenge when two nuts that were connected to each other were recognized as one contour, which made counting them difficult. However, we were able to solve this issue by using appropriate morphology techniques to separate the nuts. Furthermore, as the presence of various environmental factors or variables can produce different outcomes than those achieved through traditional image processing, it is advisable to classify any supplementary data obtained through techniques like deep learning.