OPTICAL MARK RECOGNITION SHEET SCANNER

(Raghesh Krishnan K, Assistant Professor in Department of **Computer Science and Engineering,   
Amrita Vishwa Vidyapeetham**)

**Authors Name:** Amidela Anil Kumar   
CB.EN.U4CSE20206  
Department of **Computer Science and Engineering Amrita Vishwa Vidyapeetham**Coimbatore 641112 India  
Email -ID: cb.en.u4cse20206@cb.students.amrita.edu

**Authors Name:** Manaswini Kar  
CB.EN.U4CSE20236  
Department of **Computer Science and Engineering Amrita Vishwa Vidyapeetham**Coimbatore 641112 India  
Email -ID: cb.en.u4cse20236@cb.students.amrita.edu **Authors Name:** Dhanush Penugonda  
CB.EN.U4CSE20216  
Department of **Computer Science and Engineering Amrita Vishwa Vidyapeetham**Coimbatore 641112 India  
Email -ID: cb.en.u4cse20216@cb.students.amrita.edu **Authors Name:** Gutha Venkat Sai  
CB.EN.U4CSE20221  
Department of **Computer Science and Engineering Amrita Vishwa Vidyapeetham**Coimbatore 641112 India   
Email -ID: cb.en.u4cse20221@cb.students.amrita.edu

*Abstract*— Optical mark recognition also called optical mark reading and OMR is the process of reading information that people mark on [surveys](https://en.wikipedia.org/wiki/Survey_(human_research)), [tests](https://en.wikipedia.org/wiki/Test_(assessment)) and other paper documents. OMR is used to read [questionnaires](https://en.wikipedia.org/wiki/Questionnaire), multiple choice examination [papers](https://en.wikipedia.org/wiki/Paper) in the form of shaded [areas](https://en.wikipedia.org/wiki/Area). It is essential that these documents and papers are read and evaluated without any error to produce unbiased results. These marked papers and documents are often scanned incorrectly resulting in production of inaccurate results. Using digital image processing techniques and methods these problems can be solved to produce accurate results in an efficient manner.

Keywords— OpenCV, numpy, Python, Spatial Filtering, Structuring element (SE), MSE, Evaluation Metrics, Segmentation, Morphological Processing, Masking, OMR sheet scanning, segmentation.

# **Introduction**

Optical mark recognition also called optical mark reading and OMR is the process of reading information that people mark on [surveys](https://en.wikipedia.org/wiki/Survey_(human_research)), [tests](https://en.wikipedia.org/wiki/Test_(assessment)) and other paper documents. OMR is used to read [questionnaires](https://en.wikipedia.org/wiki/Questionnaire), multiple choice examination [papers](https://en.wikipedia.org/wiki/Paper) in the form of shaded [areas](https://en.wikipedia.org/wiki/Area). It is essential that these documents and papers are read and evaluated without any error to produce unbiased results. These documents are generally scanned using a light scanner. In these scanners the marked sheet is passed under a light and the checking of the sheets are done based upon the regions that absorb maximum amount of light, i.e., the marked regions. This information is then compared with the answer key or any other valid document used for verification and the final result is produced. These marked papers and documents are often scanned incorrectly. This can happen due to reflection of light on the marked region due to type of ink used or when the sheet is not marked appropriately. These errors in scanning can lead to incorrect or inaccurate results.

Using digital image processing techniques and methods these problems can be solved to produce accurate results in an efficient manner. This can produce better results when compared to the traditional method of light scanning. OpenCv along with other libraries like pandas, numpy, etc, in python are used to scan the sheets. The images of these sheets are used to process the data in the image, the image is processed using various methods like segmentation and filters to produce the final result.

# **2. index**

1. Image Acquisition
2. Spatial Domain Preprocessing
3. Frequency Domain Preprocessing
4. Noise Models and Noise removal
5. Segmentation
6. Morphological Processing
7. Region Splitting

**3. IMAGE ACQUISITION**

Image is captured using a scanner or a digital camera and used as input to check the OMR sheet to produce results.

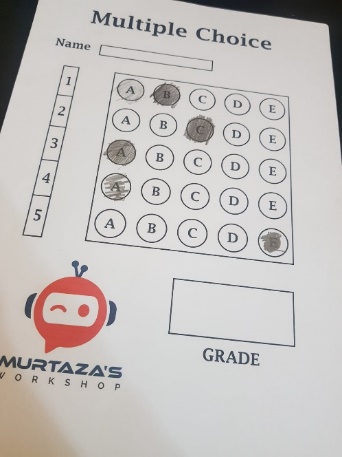


Figure 1: Input image captured

**4.1 SPATIAL DOMAIN PROCESSING**

Spatial Domain functions implemented:

* Resizing the image
* Conversion of color image to grayscale image
* Histogram Equalization

The image is resized for ease of use and also to cut out the unnecessary area of image.

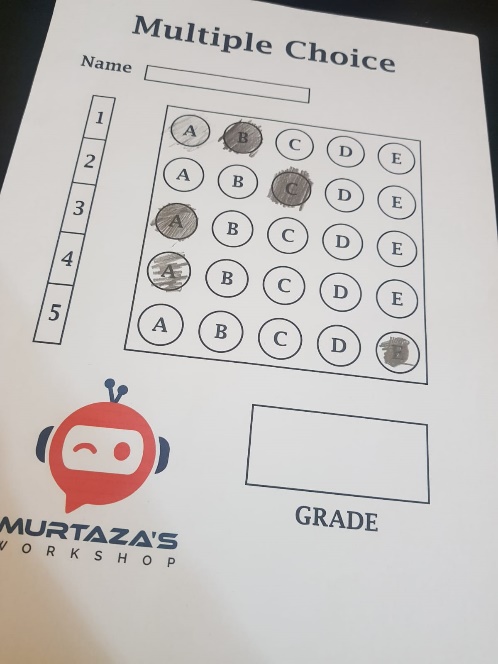


Figure 2: Original image

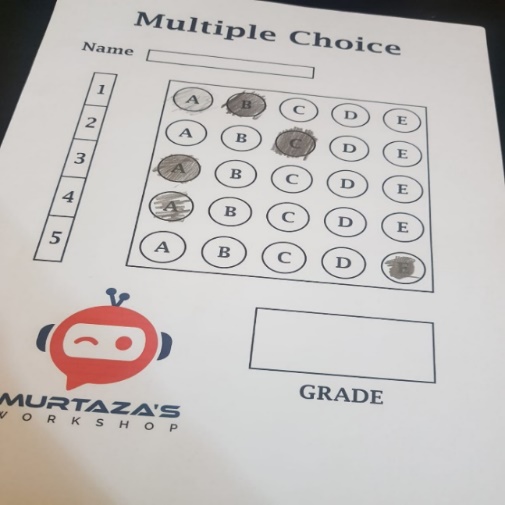


Figure 3: Resized image

Image is then converted into a grayscale image as a grayscale image can be easily processed using digital image processing methods and colored image is not required to produce results in the given problem.

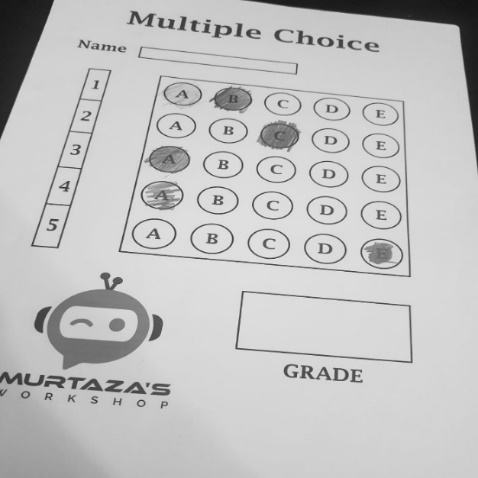


Figure 4: Image converted to grayscale image

Histogram equalization:

Histogram equalization is used to improve contrast in image. This technique stretches out or spreads out the intensity range of the image to accomplish improvement of contrast in image.

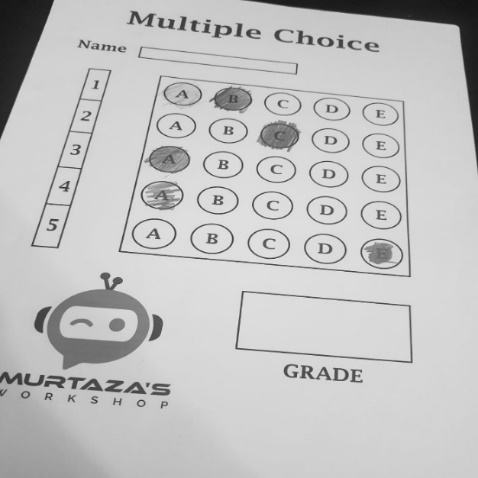


Figure 5: Image-1

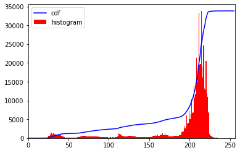


Figure 6: Histogram of image-1

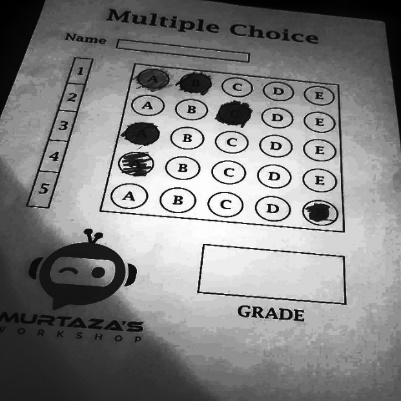


Figure 7: Image after histogram equalization

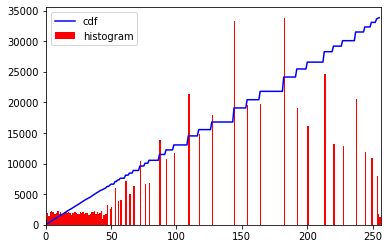


Figure 8: Histogram of image after histogram equalization

4.2 FREQUENCY DOMAIN PROCESSING

Frequency Domain is used when certain parts or characteristics of the image which cannot be changed using Spatial Domain.

4.2.1 FOURIER TRANSFORM

.

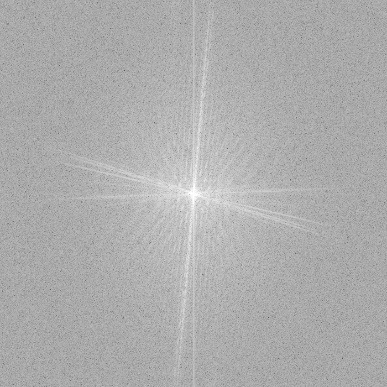


Figure 9: Magnitude spectrum of image

In the Fourier image the central regions represent the low frequency components and corner regions represent the high frequency components and the absolute center of the image represents the DC value with zero frequency which is the total intensity of the image. We can get back the spatial image by performing what is called the Inverse Fourier transform.

4.2.2 FILTERING IN THE FREQEUNCY DOMAIN

In frequency domain, low pass and high pass filtering are done. To view the original spatial image, we perform Inverse Fourier transform on the filtered Fourier image.

**A. Low Pass Filters**

**1. Ideal Low Pass Filter (ILPF)**

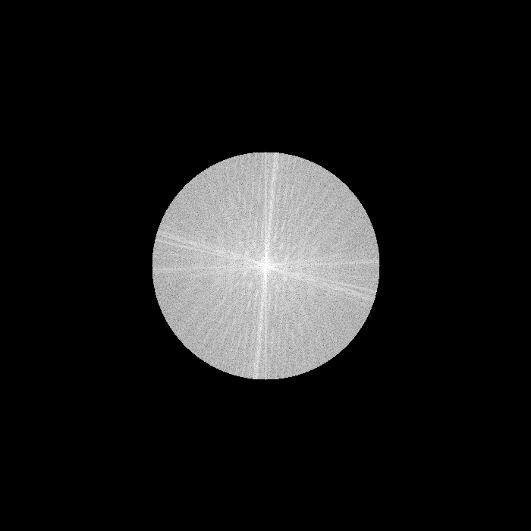
****

Figure 10: Magnitude spectrum of ideal lowpass filter

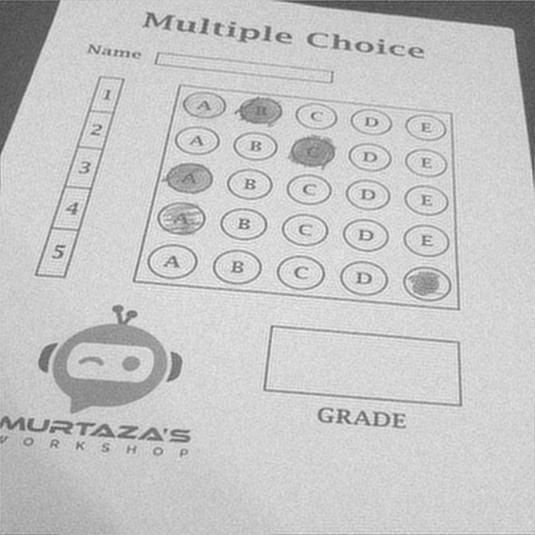
****

Figure 11: Image processed with ideal lowpass filter

**2. Gaussian Low Pass Filter (GLPF)**

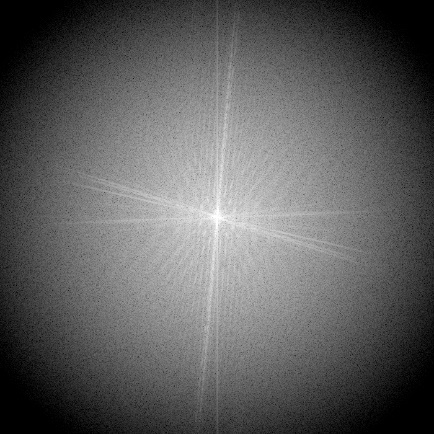
****

Figure 12: Magnitude spectrum of gaussian filter

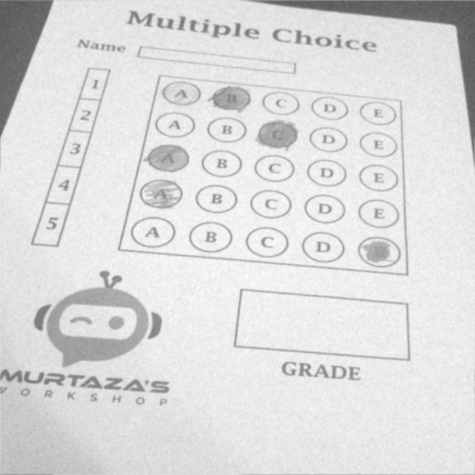
****

Figure 13: Image processed with gaussian lowpass filter

**3. Butterworth Low Pass Filter (BLPF)**

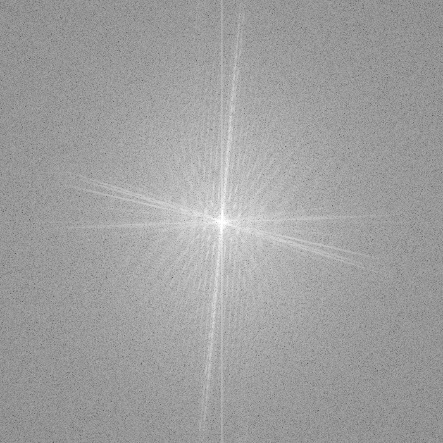
****

Figure 14: Magnitude spectrum of butterworth filter

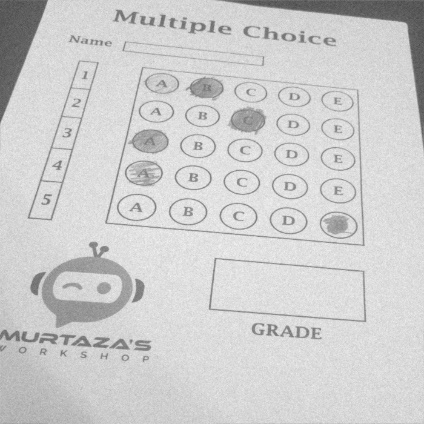
****

Figure 15: Image processed with butterworth lowpass filter

4.2.3 ANALYSIS OF FREQUENCY DOMAIN FILTERS

Ideal lowpass filter in used to smoothen the image in frequency domain. It removes high frequency noise from the image and preserves low frequency components. In the output image, smoothening of the image can be observed along with ringing effect. In ideal lowpass filtering, the attenuation of frequencies is sharp and not gradual, i.e., it completely eliminates all frequencies above the cutoff frequency while passing those below remain unchanged.

Both Ideal low pass and high pass filters exhibit the ringing effect. This is due to the sharp change from the blocked region to the allowed region. This ringing effect will hinder the performance of any further processing done on the image.

When butterworth filter is used, in the output image, smoothening of the image can be observed, but unlike ideal lowpass filter no ringing effect is observed and has quick roll-off around the cutoff frequency which improves in increasing order.

Gaussian filter is used for reduction of noise and blurring regions of an image. Ringing effect cannot be observed when gaussian filter is used.

4.3 NOISE MODELS

Image noise is random variation of brightness or color information in [images](https://en.wikipedia.org/wiki/Image), and is usually an aspect of [electronic noise](https://en.wikipedia.org/wiki/Electronic_noise). It can be produced by the [image sensor](https://en.wikipedia.org/wiki/Image_sensor) and circuitry of a [scanner](https://en.wikipedia.org/wiki/Image_scanner) or [digital camera](https://en.wikipedia.org/wiki/Digital_camera). Noise is always presents in digital images during image acquisition, coding, transmission, and processing steps.

The image can be affected by several kinds of noises, such as, impulse noise, gaussian noise, Poisson noise, uniform noise. OMR sheets might have gaussian noise or uniform noise.

4.3.1 EVALUATION METRICS FOR NOISE MODELS

Noise present in images vary from one image to another, it becomes hard to generalize a filter that is suited for all images. Furthermore, every filter has its own parameters which change the outcome of the amount of noise removed.

Mean squared error is one of the methods used to evaluate how well has the image been modified or processed, and the same has been used for our problem.

4.3.1.1 MSE

Mean Squared Error or MSE is used to measure the difference between the source image and the segmented image, the smaller the value of MSE, the closer the filtered image is towards the input image and lower is the performance. The MSE represents the cumulative squared error between the compressed or changed and the original image.

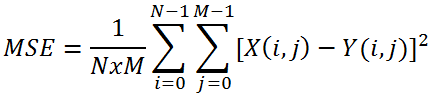


Figure 16: Mathematical Representation of MSE

4.3.2 FILTERS

1. Mean Filter:

The mean filter is used to blur an image in order to remove noise. It involves determining the mean of the pixel values within a [n x n] kernel. The pixel intensity of the center element is then replaced by the mean. This eliminates some of the noise in the image and smooths the edges of the image.

2. Median Filter:

The median filter calculates the median of the pixel intensities that surround the center pixel in a [n x n] kernel. The median then replaces the pixel intensity of the center pixel. The median filter does a better job of removing salt and pepper noise than the mean and gaussian filters.

3. Laplacian Filter:

A Laplacian filter is an edge detector used to compute the second derivatives of an image, measuring the rate at which the first derivatives change. This determines if a change in adjacent pixel values is from an edge or continuous progression.

4. Box Filter:

Box filter are frequently used to approximate gaussian blur. A box filter, also known as box blur is generally implemented as an image effect that affects the whole screen. The blurred color of the current pixel is the average of the current pixel’s color and its 8 neighboring pixels.

5. Bilateral Filter:

A bilateral filter is used for smoothening images and reducing noise while preserving edges. It is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels.

6. Wiener Filter:

The wiener filter is the MSE-optimal stationary linear filter for images degraded by additive noise and blurring. It is used to denoise and deblur noisy images corrupted by gaussian noise and motion blurring.

7. Minimum Filter:

The minimum filter is defined as the minimum of all pixels within a local region of an image. The minimum filter is typically applied to an image to remove positive outlier noise.

8. Maximum Filter:

Maximum filter is a non-linear filter commonly used to locally smooth data and diminish salt and pepper noise in an image. It replaces each pixel value with the maximum value in its neighborhood pixel window. It is the opposite of minimum filter.

9. Gaussian Filter:

A gaussian filter is a low pass filter used for reducing noise and blurring regions of an image. The filter is implemented as an odd sized symmetric kernel which is passed through each pixel of the region of interest.

4.3.2 NOISES AND METHODS TO DENOISE

1. Gaussian Noise:

It is also called electronic noise because it occurs in amplifiers and detectors. Gaussian noise generally disturbs the gray values in digital images. They are designed and characterized by its PDF or it normalizes histogram with respect to the gray values.

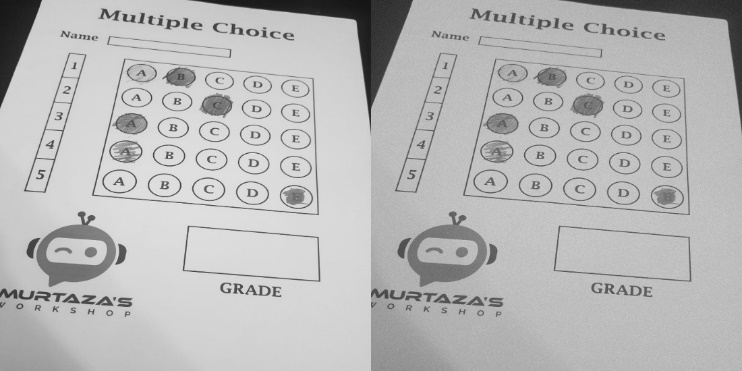


Figure 17: Original Image

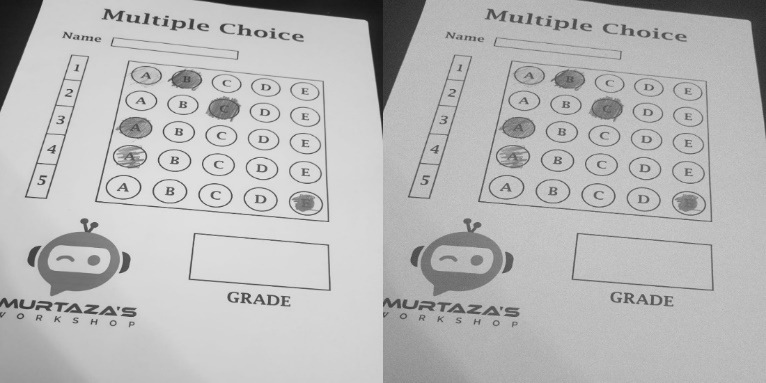


Figure 18: Image with Gaussian Noise

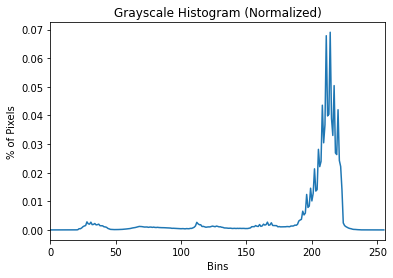


Figure 19: Histogram of original image



Figure 20: Histogram of image with gaussian noise

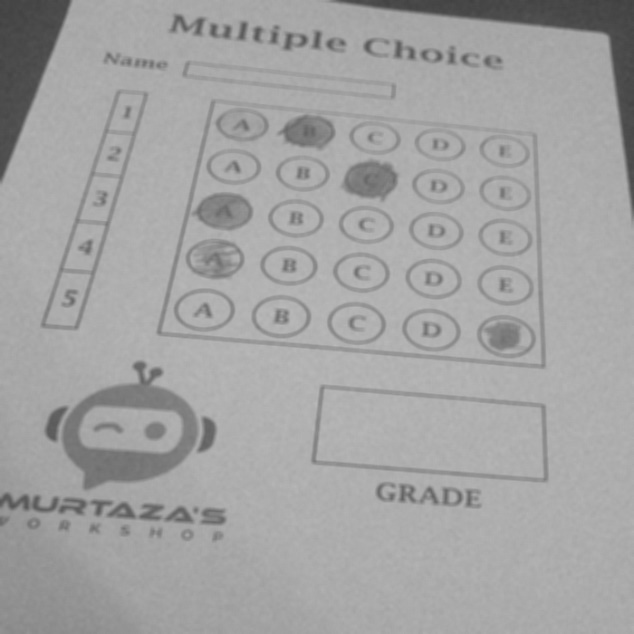


Figure 21: Image after applying mean filter of kernel size 5

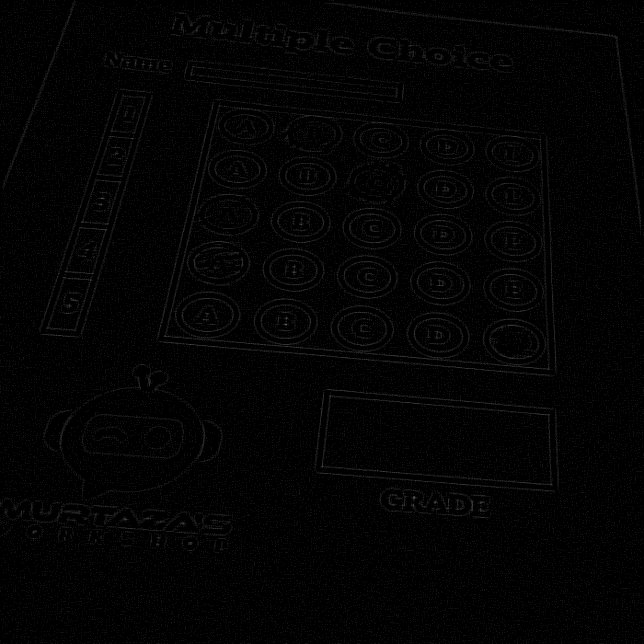


Figure 22: Image after applying Laplacian filter

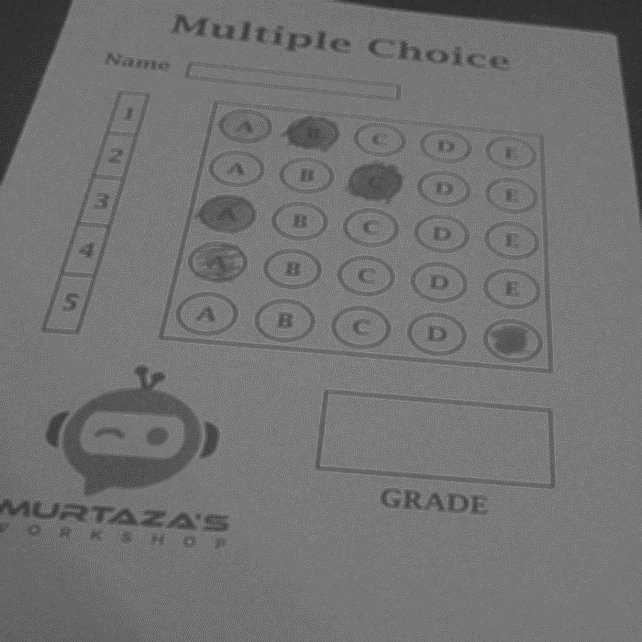


Figure 22: Image after applying both Laplacian and mean filters together

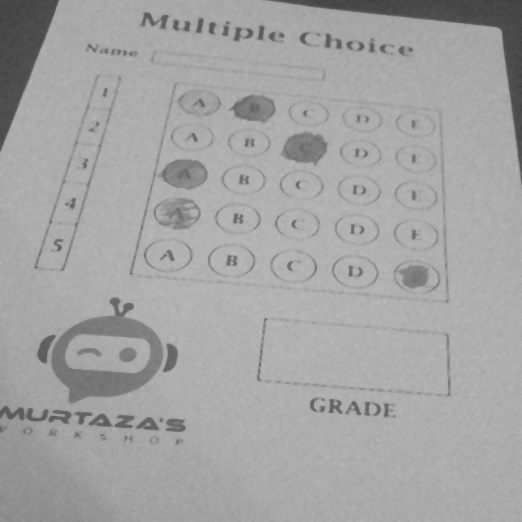


Figure 23: Image after applying median filter of kernel size 5

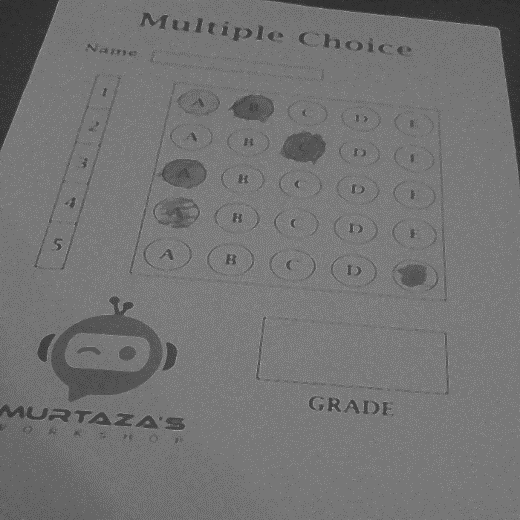


Figure 24: Image after applying both Laplacian and median filters together

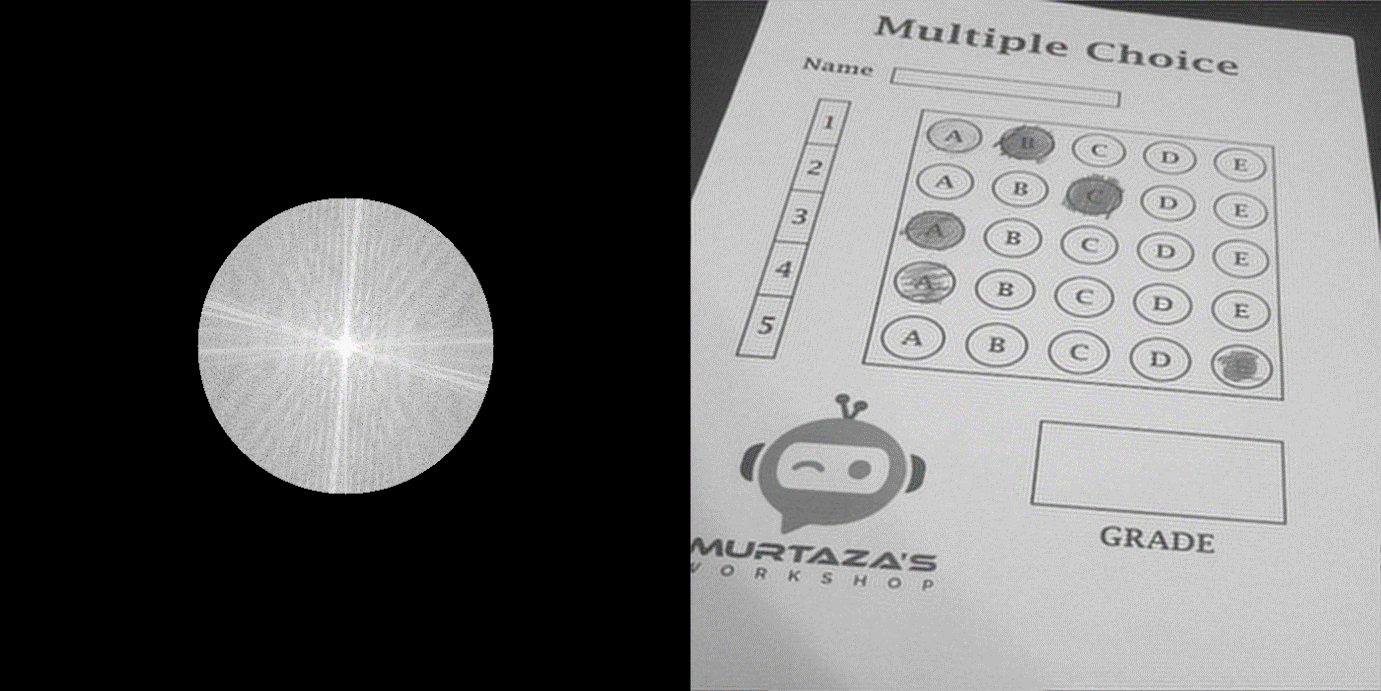


Figure 25: Image after applying Ideal lowpass filter

In the output image, smoothening of the image can be observed along with ringing effect. In ideal lowpass filtering, the attenuation of frequencies is sharp and not gradual, i.e., it completely eliminates all frequencies above the cutoff frequency while passing those below remain unchanged.



Figure 26: Image after applying Butterworth lowpass filter

In the output image, smoothening of the image can be observed, but unlike ideal lowpass filter no ringing effect is observed and has quick roll-off around the cutoff frequency which improves in increasing order.

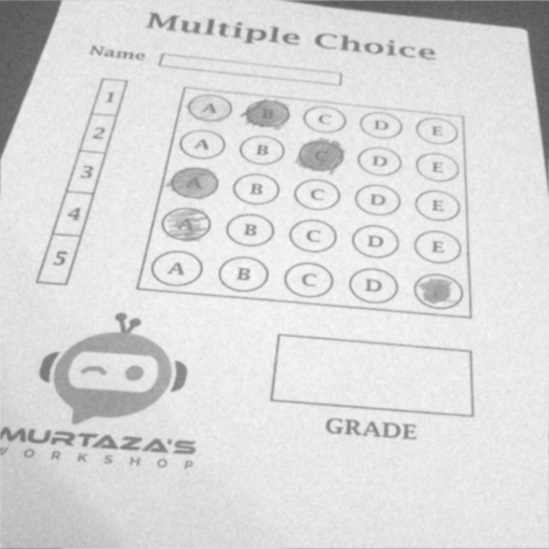


Figure 27: Image after applying Gaussian lowpass filter

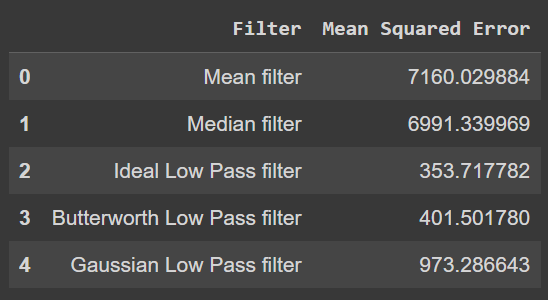


Figure 28: Mean squared error of different filter used to denoise an image having gaussian noise.

1. Uniform Noise:

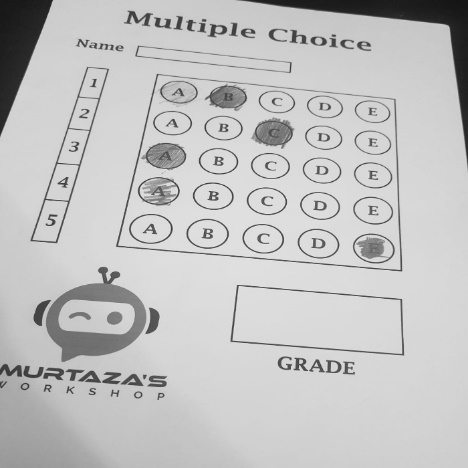


Figure 29: Original image

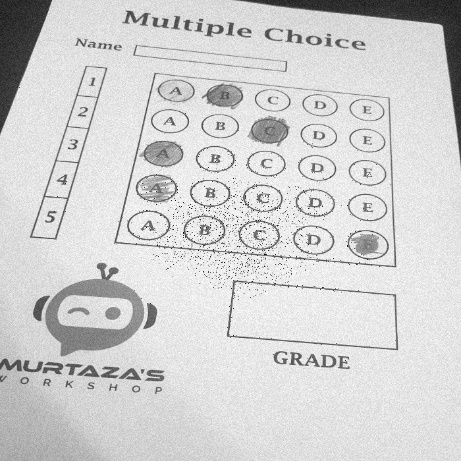


Figure 30: Image with uniform noise

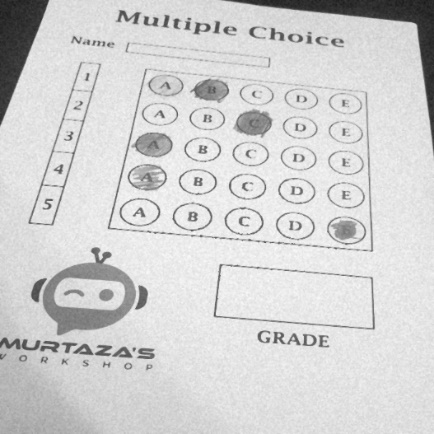


Figure 31: Image after applying median filter of kernel size 3

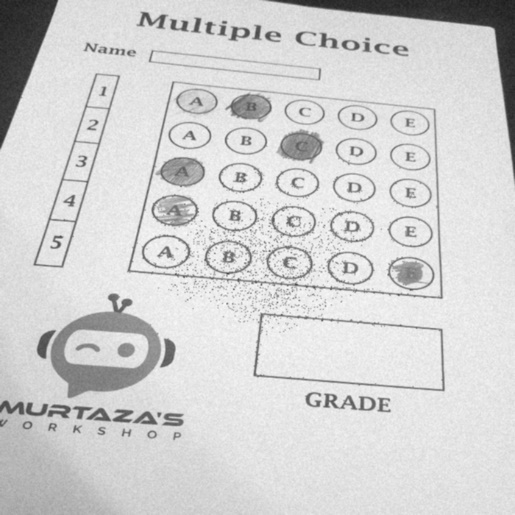


Figure 32: Image after applying box filter of kernel size 3

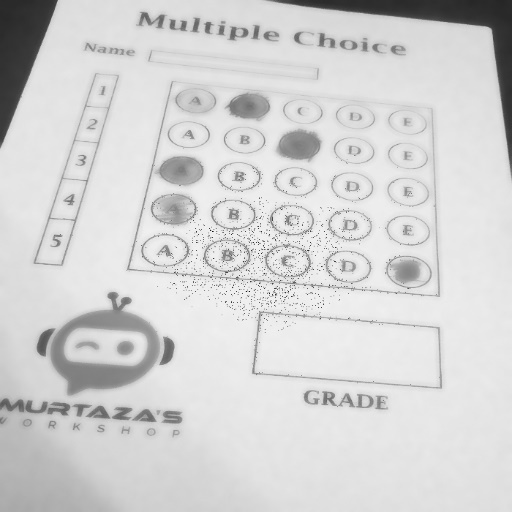


Figure 33: Image after applying bilateral filter



Figure 34: Image after applying wiener filter

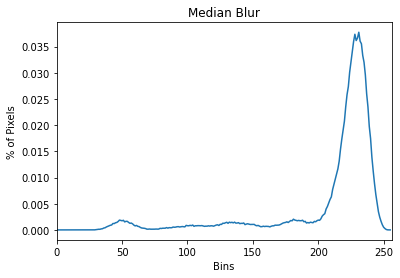


Figure 35: Histogram of image after applying median filter

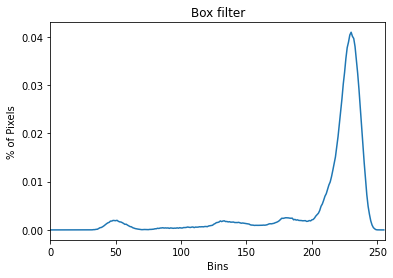


Figure 36: Histogram of image after applying box filter

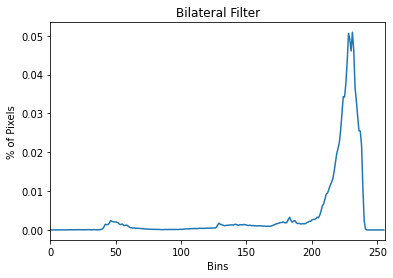


Figure 37: Histogram of image after applying bilateral filter

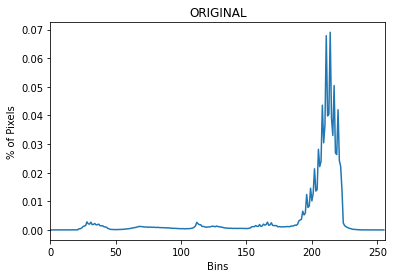


Figure 38: Histogram of original image

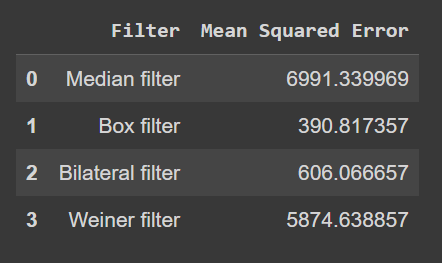


Figure 39: Mean squared error of different filter used to denoise an image having uniform noise.

It can be observed that bilateral filter denoises better by preserving the edges compared to box filter and median filter.

1. Impulse Noise:

Impulse noise is also called salt and pepper noise.

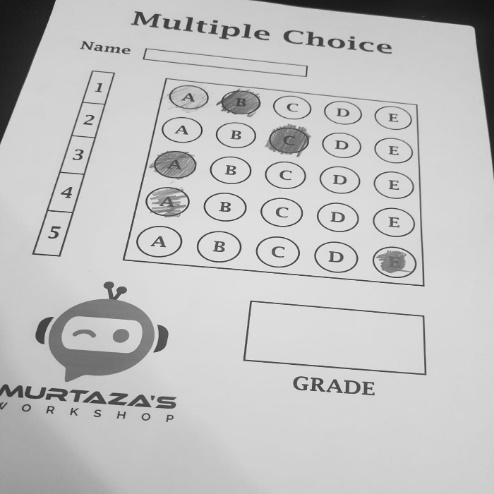


Figure 40: Original image



Figure 41: Image with impulse noise

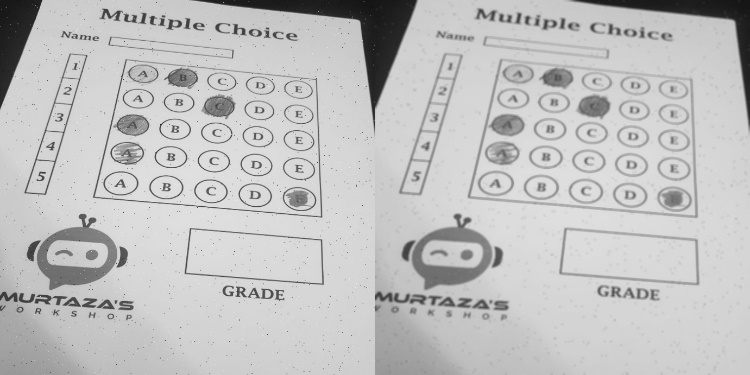


Figure 42: Image after applying mean filter with kernel size 5

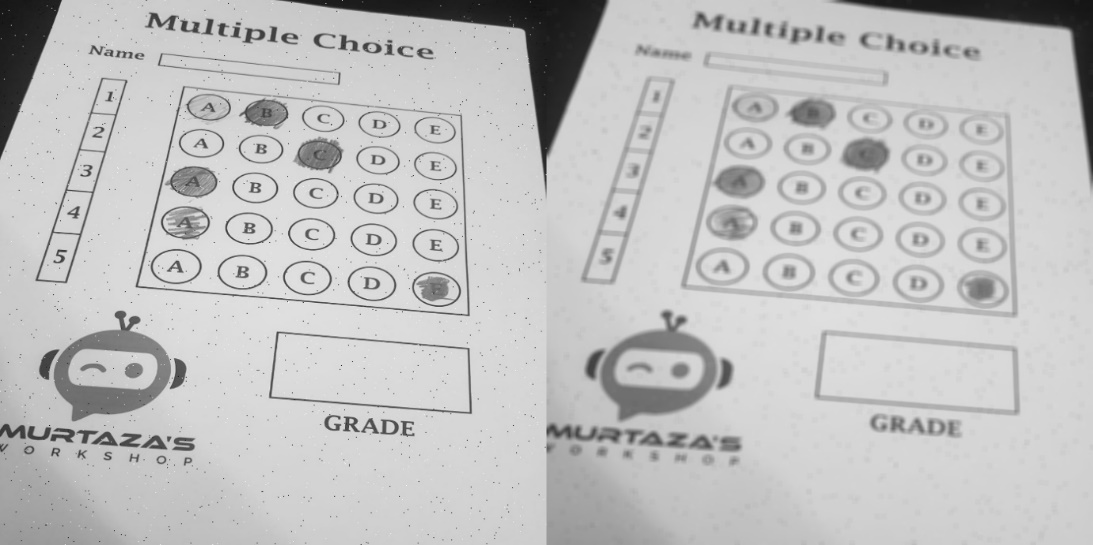


Figure 43: Image after applying mean filter with kernel size 7

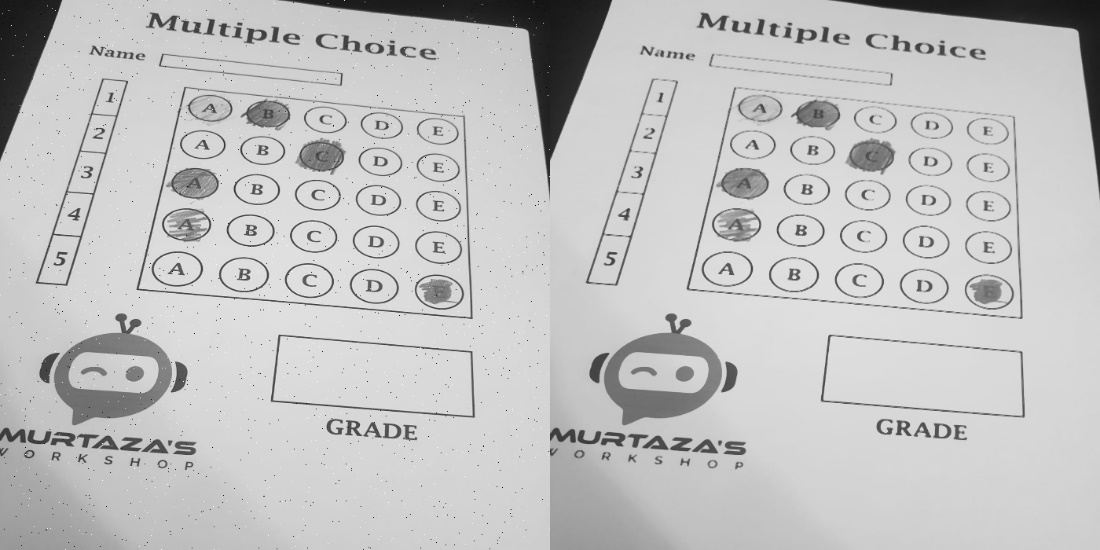


Figure 44: Image after applying median filter with kernel size 3

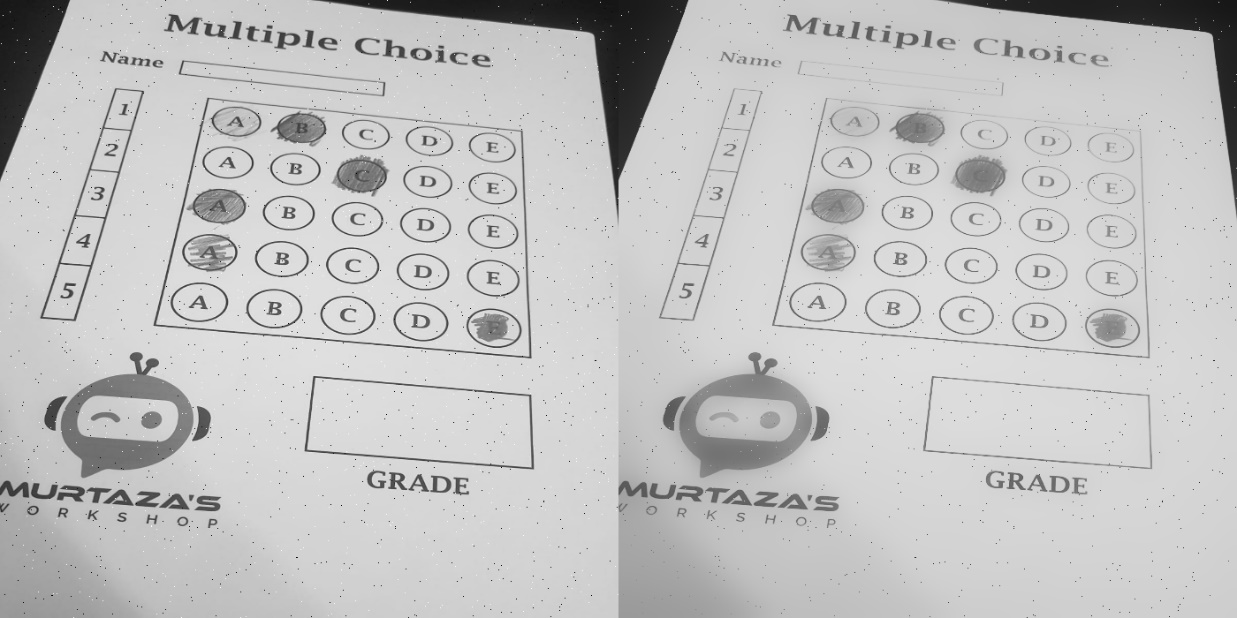


Figure 45: Image after applying bilateral filter

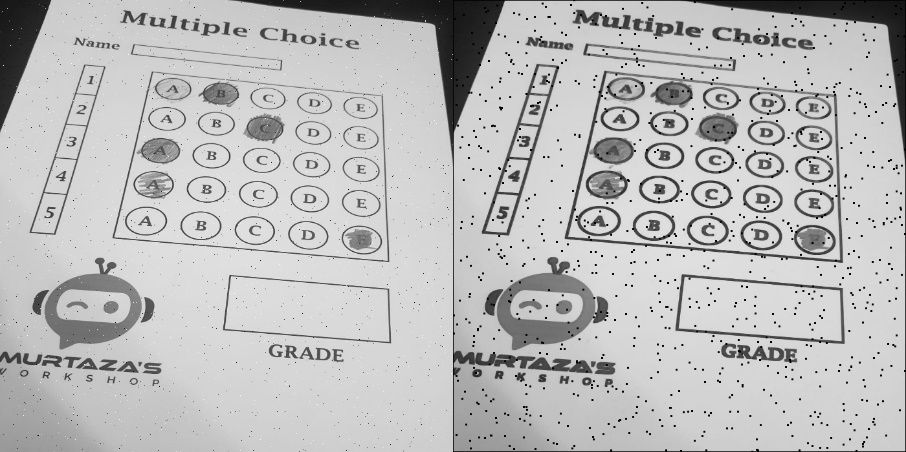


Figure 46: Image after applying minimum filter

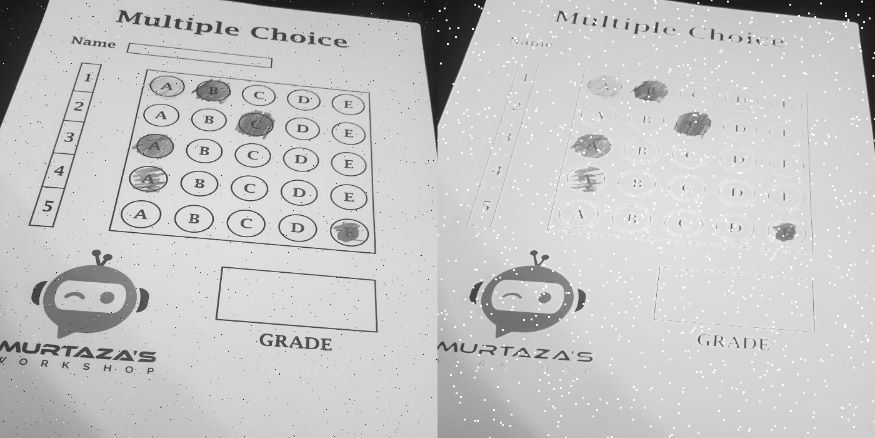


Figure 47: Image after applying maximum filter

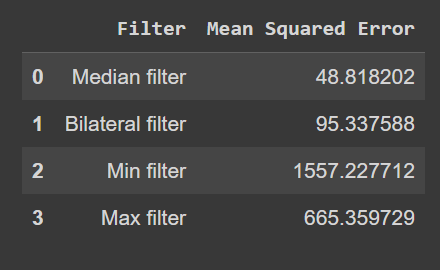


Figure 48: Mean squared error of different filter used to denoise an image having impulse noise

it can be observed that median can almost completely denoise the image which has impulse noise.

1. Poisson Noise:

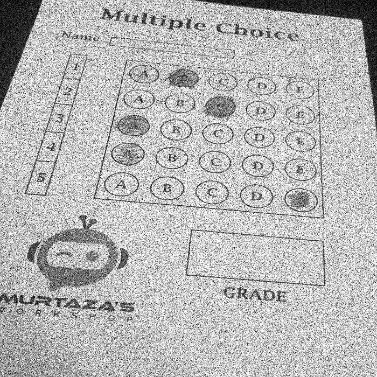


Figure 49: Image with poisson noise

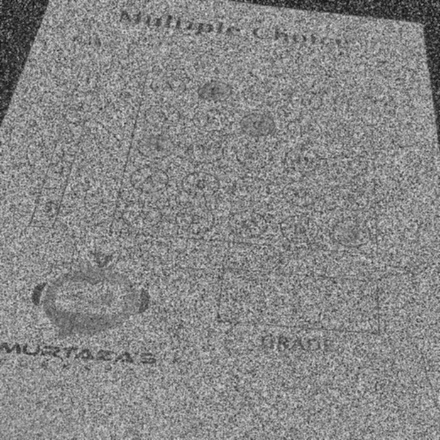


Figure 50: Image after applying gaussian filter of kernel size 3

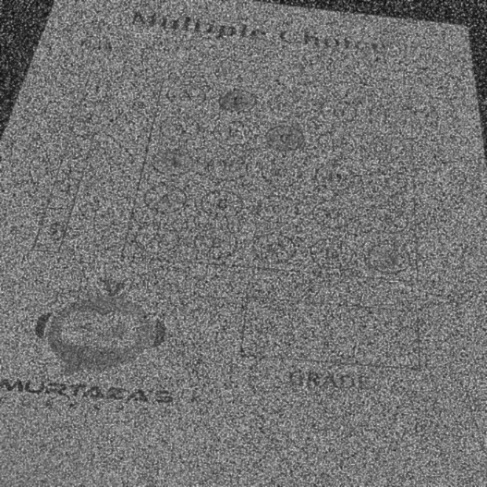


Figure 51: Image after applying Laplacian filter

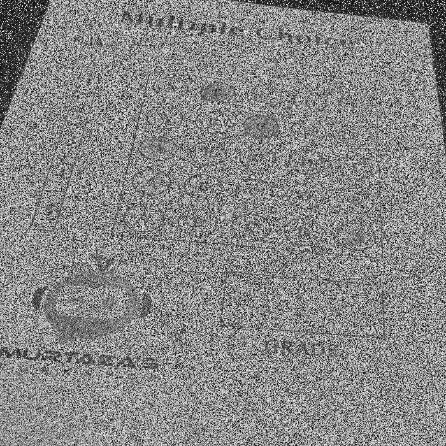


Figure 52: Image after applying bilateral filter

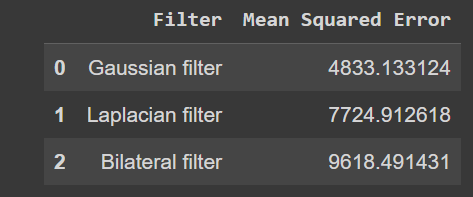


Figure 53: Mean squared error of different filter used to denoise an image having poisson noise

**5. SEGMENTATION**

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image or based on region of interest according to requirement. Image segmentation could involve separating foreground from background, or clustering regions of pixels based on similarities in color or shape.

In this problem to find our region of interest and segment it we will fill first have to find the contours in our image.

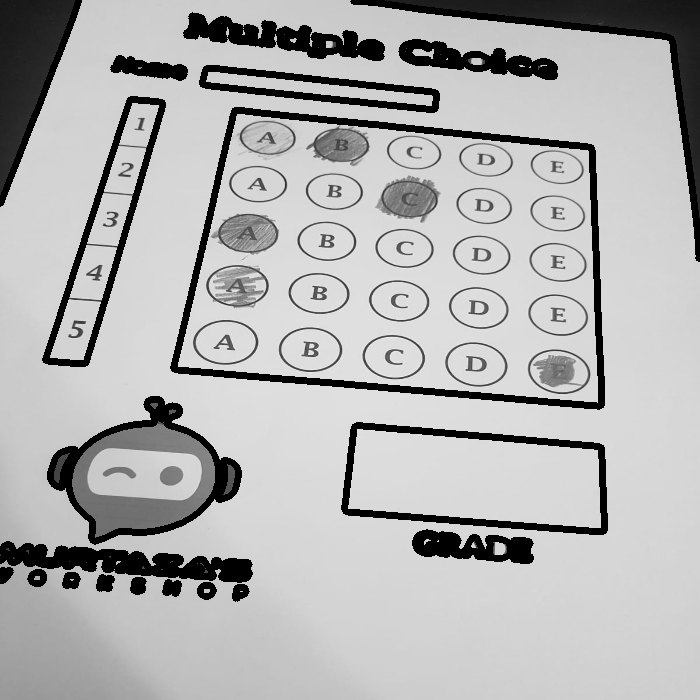


Figure 54: Image with contours detected

Further we find the rectangular box in which all the answers are marked, i.e., the rectangular box with maximum area.

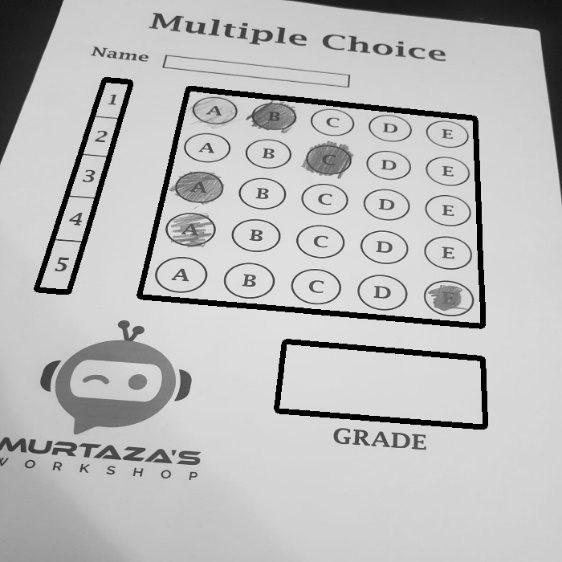


Figure 55: Image with rectangular contours detected

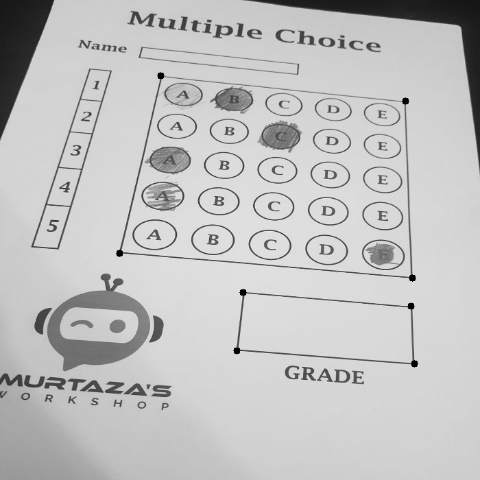


Figure 56: Image with the coordinates of required regions marked

It can be observed that the image is tilted. To further work on the image and obtain accurate result in an efficient manner, we change the perspective of the image using perspective transformation functions.

In **perspective transform**, we can change the perspective of a given image or video for getting better insight into the required information. Perspective transform is a feature that is very useful if we want to align the image properly. It transforms the image in a straight manner after perspective transformation is applied to it, i.e., the image appears as a top view of the image.

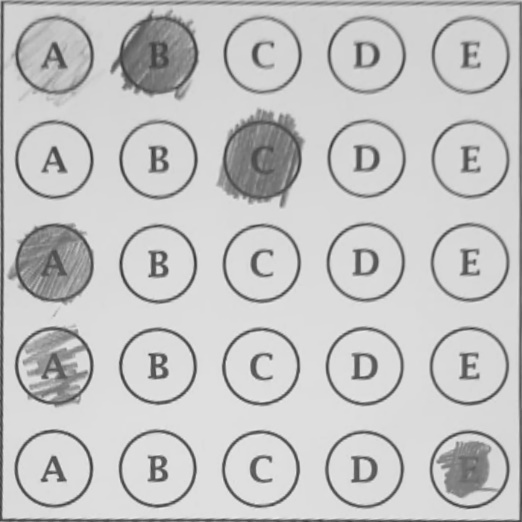


Figure 57: image after perspective transform

* 1. . EDGE DETECTION ALGORITHMS

The various algorithms for edge detection are sobel, Robert, Prewitt, Laplacian of Gaussian – Marr Hildreth, Laplacian, Kirsch Compass Mask. For our problem, we have used canny edge detection and also Laplacian of Gaussian.

CANNY EDGE DETECTION:

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

The Canny edge detection algorithm is composed of 5 steps:

1. Noise reduction;
2. Gradient calculation;
3. Non-maximum suppression;
4. Double threshold;
5. Edge Tracking by Hysteresis.

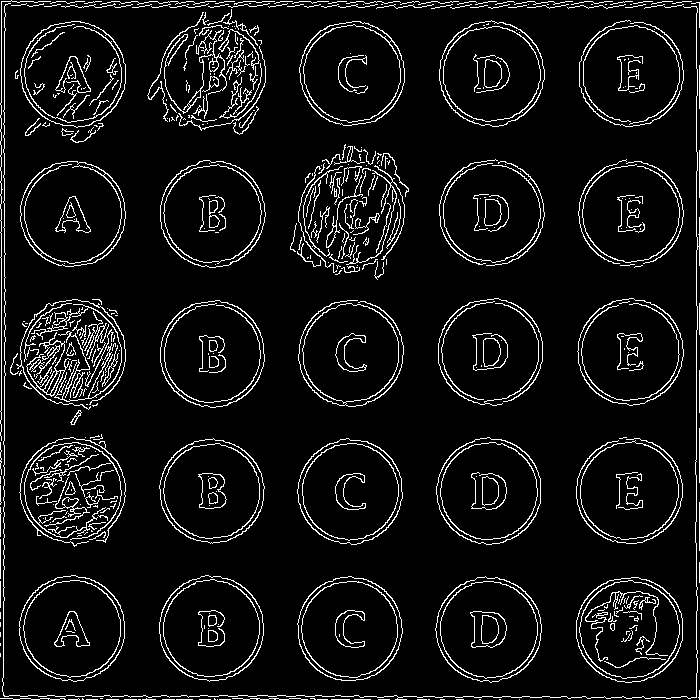


Figure 58: Canny edge detection done on image

LAPLACIAN OF GAUSSIAN:

The Laplacian filter is used to detect the edges in the images. But it has a disadvantage over the noisy images. It amplifies the noise in the image. Hence, first we use a gaussian filter on the noisy image to smoothen it and then subsequently use the Laplacian filter for edge detection.

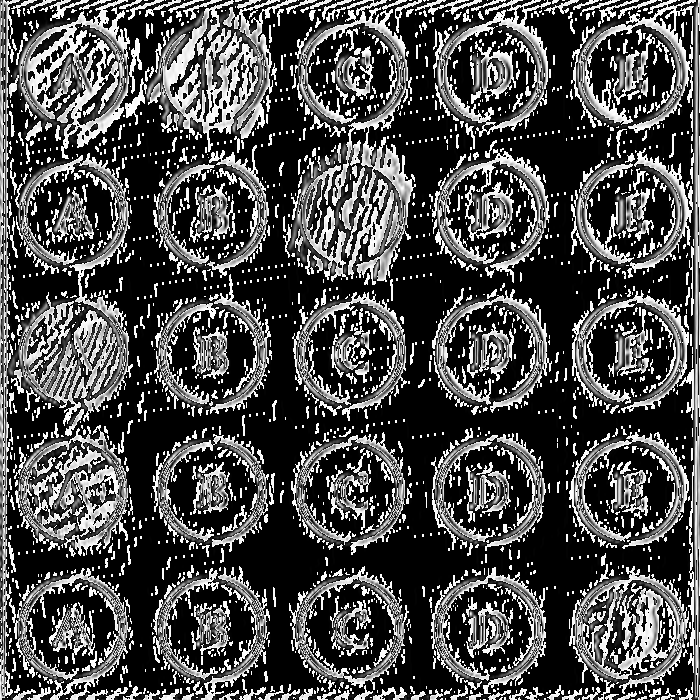


Figure 59: x-derivative of the image

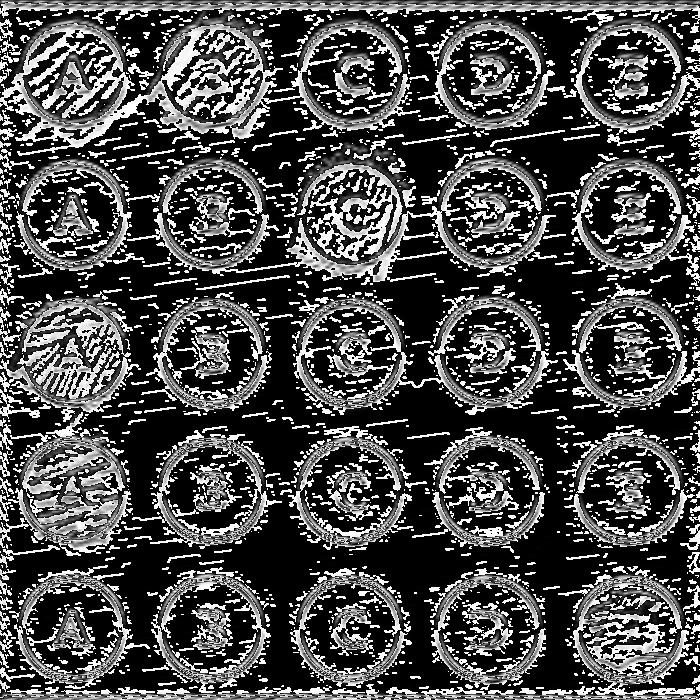


Figure 60: y-derivative of the image

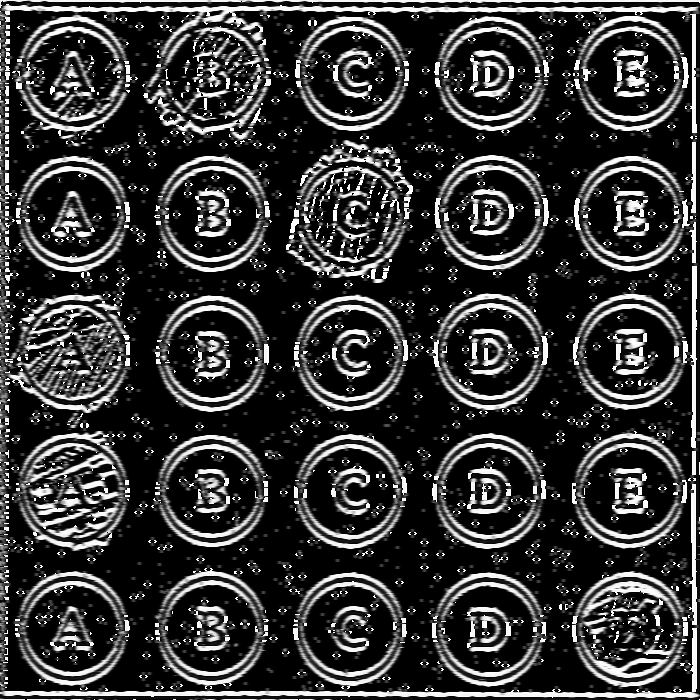


Figure 61: LoG with sigma value 1.4 of image with impulse noise

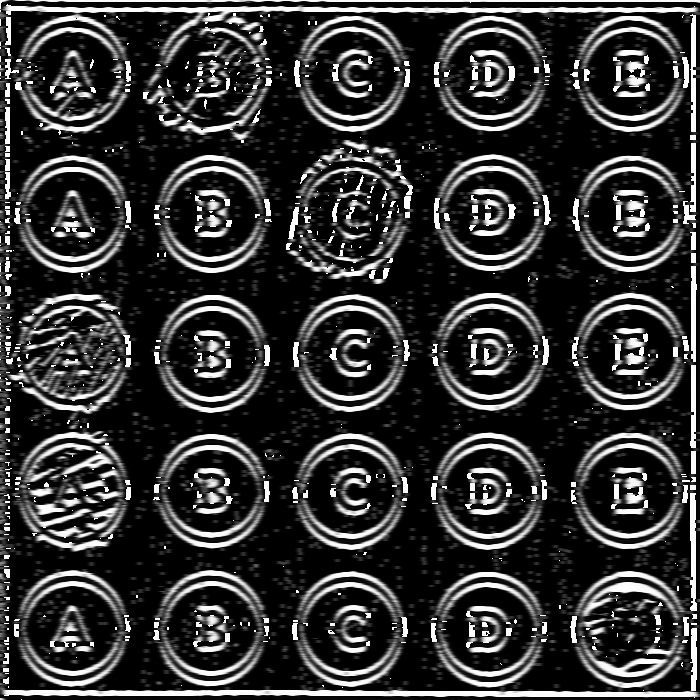


Figure 62: LoG with sigma value 2 of image with impulse noise

**4.5.1 OTSU THRESHOLDING**

Image thresholding is used to binarize the image based on pixel intensities. The input to such thresholding algorithm is usually a grayscale image and a threshold. The output is a binary image.

If the intensity of a pixel in the input image is greater than a threshold, the corresponding output pixel is marked as white (foreground), and if the input pixel intensity is less than or equal to the threshold, the output pixel location is marked black (background).

A problem with simple thresholding is that you have to manually specify the threshold value. We can manually check how good a threshold is by trying different values but it is tedious and it may break down in the real world.

Image thresholding is the simplest kind of image segmentation because it partitions the image into two groups of pixels — white for foreground, and black for background.

Steps involved in Otsu Thresholding:

1. Process the input image
2. Obtain image histogram (distribution of pixels)
3. Compute the threshold value T

Replace image pixels into white in those regions, where saturation is greater than T and into the black in the opposite cases.

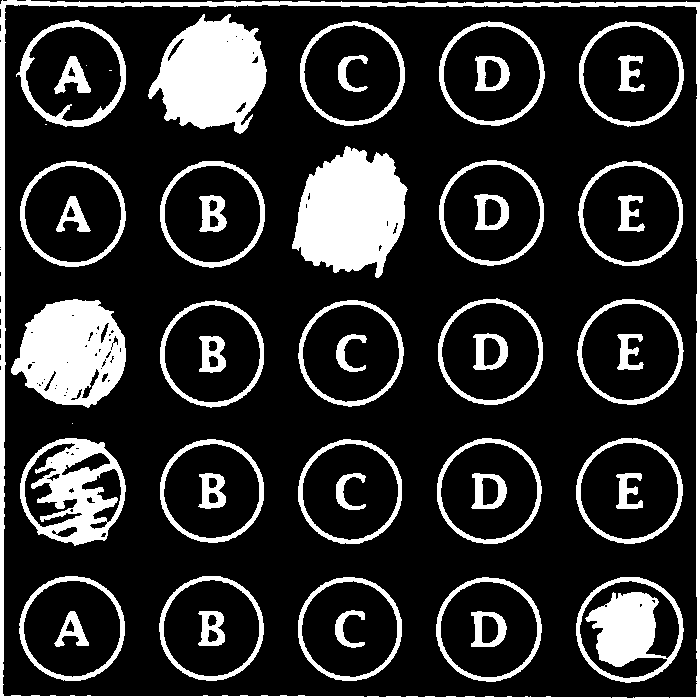


Figure 63: Image after applying otsu thresholding

**6. MORPHOLOGICAL PROCESSING**

Morphological processing is a collection of nonlinear operations based on the shape of features in an image. Morphological techniques examine images using small shapes or templates called as a Structuring Element. A structuring element is placed at every possible position in the image and compared to corresponding neighboring pixels.

These operations are applied to remove imperfections caused during segmentation. Thus they typically operate on Binary Images.

6.2 STRUCTURING ELEMENT (SE)

It is a small set/sub image used to probe the image under study for properties of interest. The shape and size of the SE must be in accordance with the geometric properties for the objects. We check for 3 conditions when we apply the SE.

* **Fit** - When all elements match.
* **Hit** - At least one element matches.
* **Miss** - None of the elements match.

Our application of Morphological processing is done on the Mask obtained post segmentation of the part where the answers for all questions are marked, this region will be further segmented into a greater number of parts later.

Various Morphological Processing operations performed:

* Dilation
* Erosion
* Opening
* Closing
* Gradient
* Top Hat
* Black Hat

1. DILATION

The value of the output pixel is the maximum value of all pixels in the neighborhood. In a binary image, a pixel is set to 1 if any of the neighboring pixels have the value 1.

Morphological dilation makes objects more visible and fills in small holes in objects. Lines appear thicker, and filled shapes appear larger.

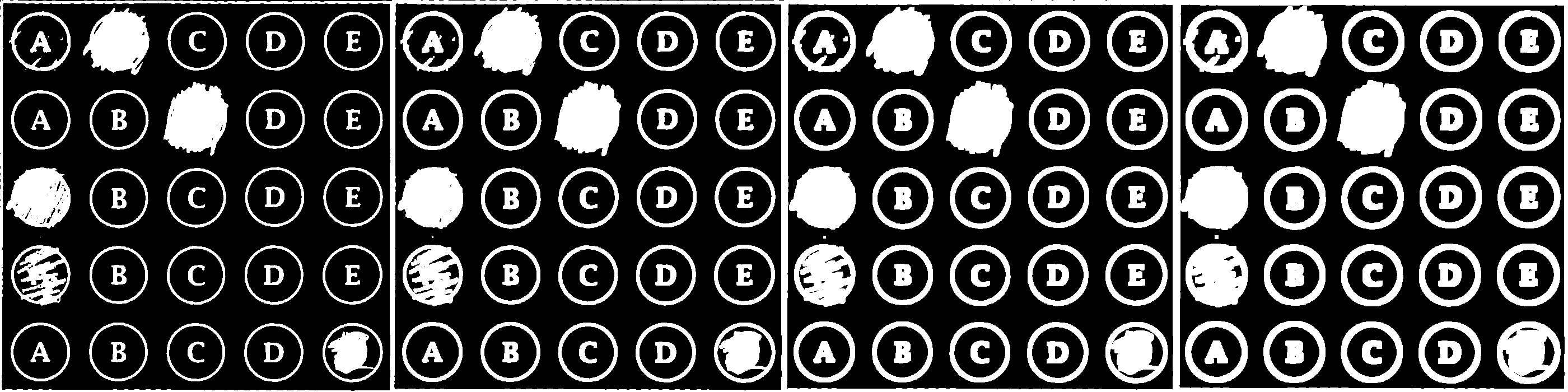


Figure 64: Image dilated with kernel size 3x3

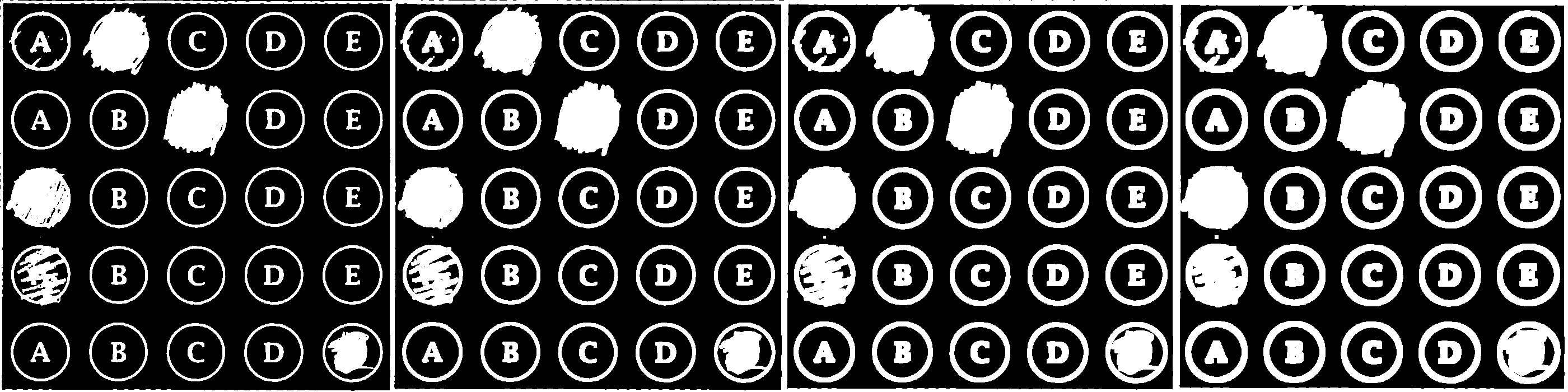


Figure 65: Image dilated with kernel size 5x5

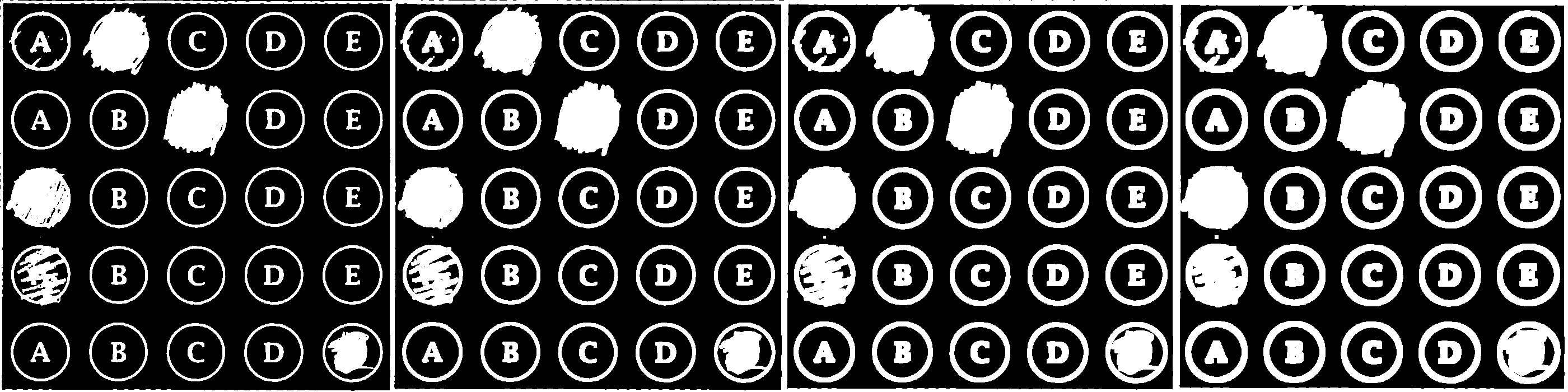


Figure 66: Image dilated with kernel size 7x7

1. EROSION

The value of the output pixel is the minimum value of all pixels in the neighborhood. In a binary image, a pixel is set to 0 if any of the neighboring pixels have the value 0.

Morphological erosion removes floating pixels and thin lines so that only substantive objects remain. Remaining lines appear thinner and shapes appear smaller.

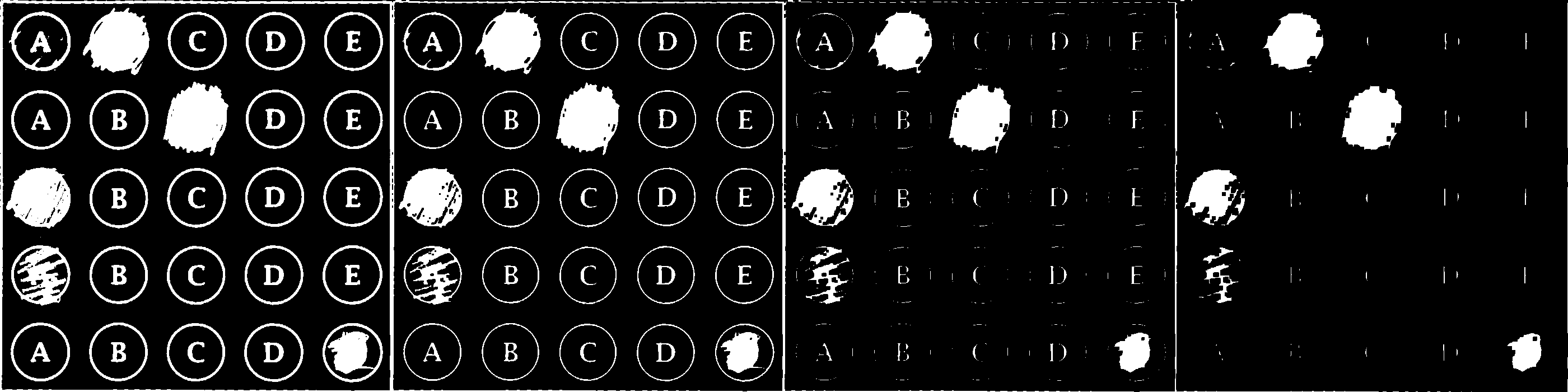


Figure 67: Image eroded with kernel size 3x3

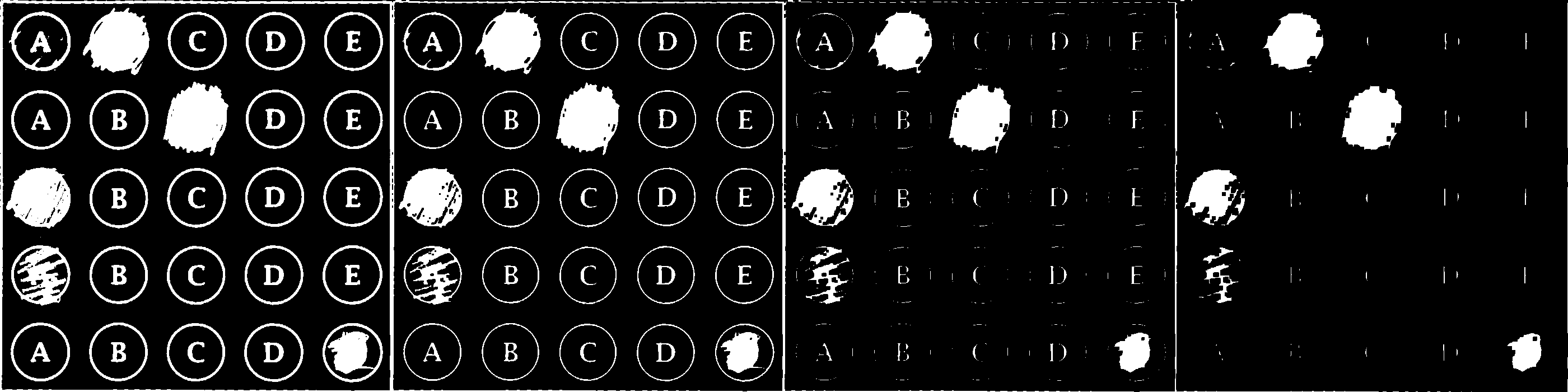


Figure 68: Image eroded with kernel size 5x5

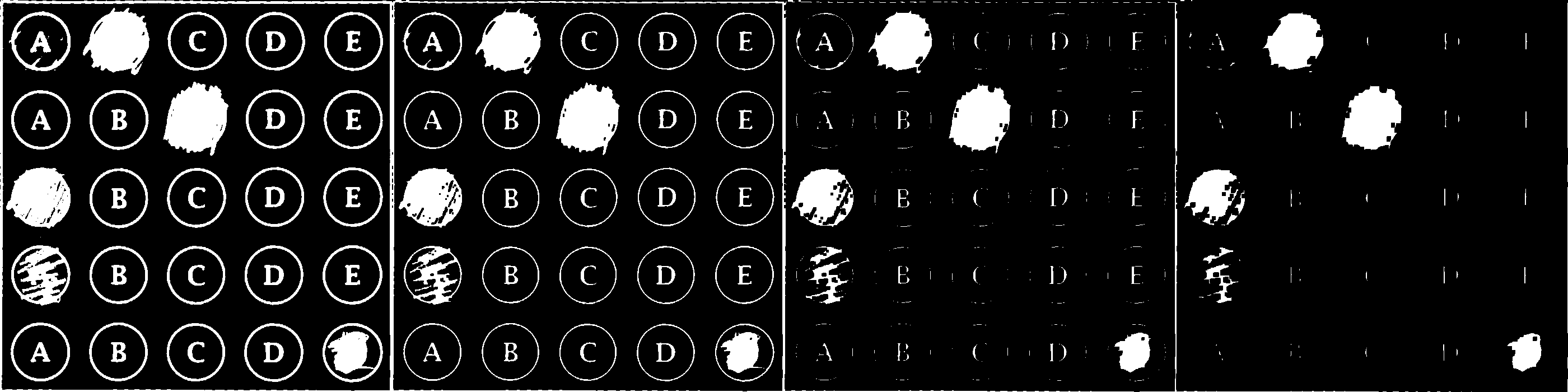


Figure 69: Image eroded with kernel size 7x7

1. OPENING

Perform morphological opening. The opening operation erodes an image and then dilates the eroded image, using the same structuring element for both operations.

Morphological opening is useful for removing small objects and thin lines from an image while preserving the shape and size of larger objects in the image.

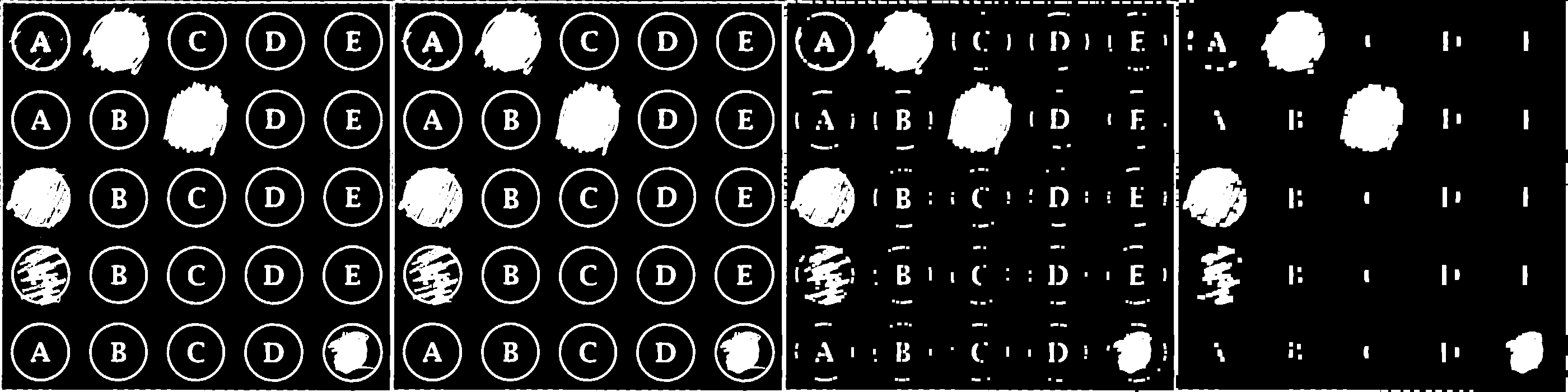
****

Figure 70: Image opened with kernel size 3x3

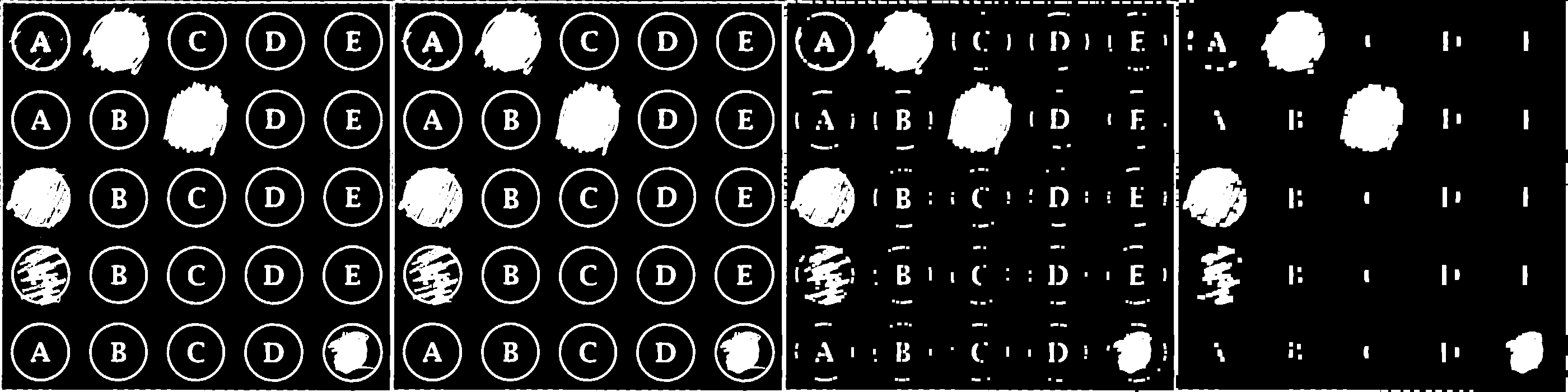


Figure 71: Image opened with kernel size 5x5

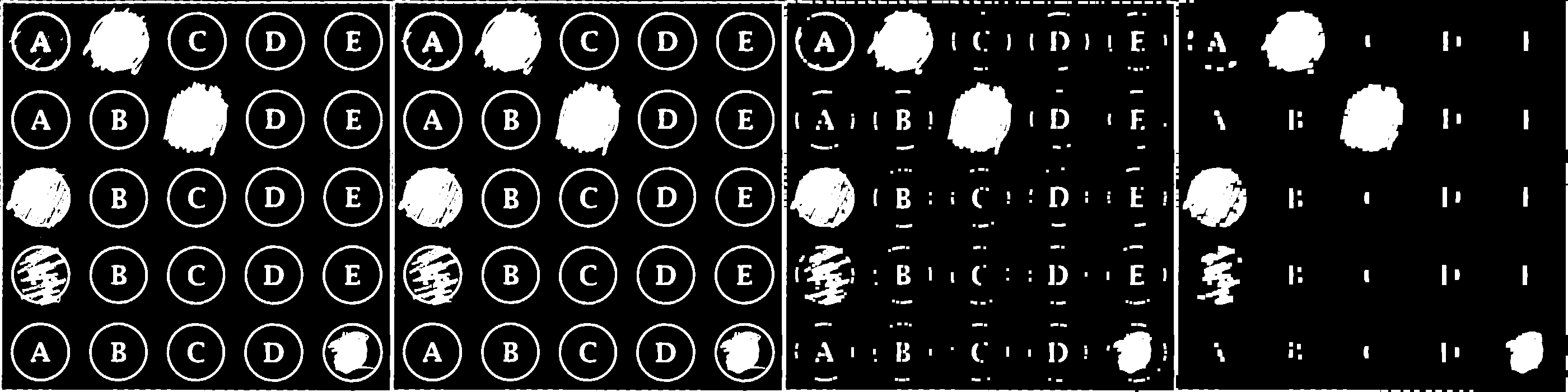


Figure 72: Image opened with kernel size 7x7

1. CLOSING

Perform morphological closing. The closing operation dilates an image and then erodes the dilated image, using the same structuring element for both operations.

Morphological closing is useful for filling small holes in an image while preserving the shape and size of large holes and objects in the image.

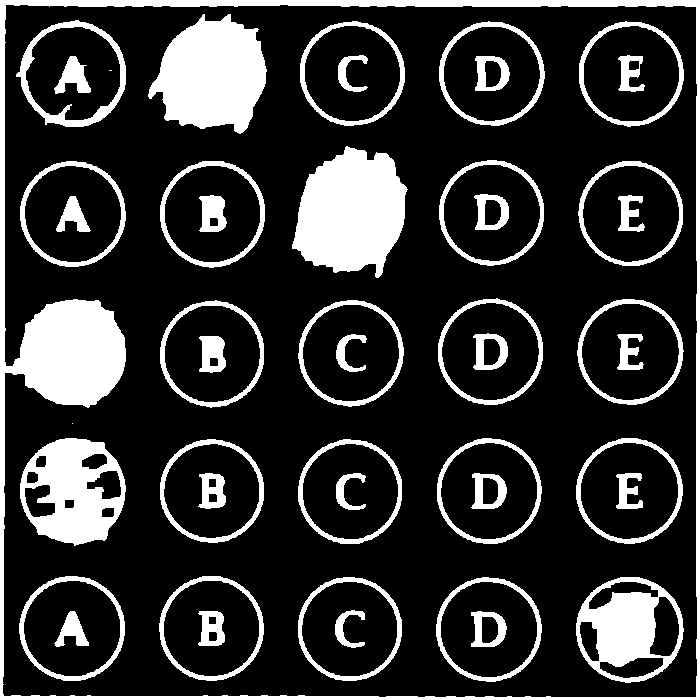


Figure 73: Image closed with kernel size 3x3

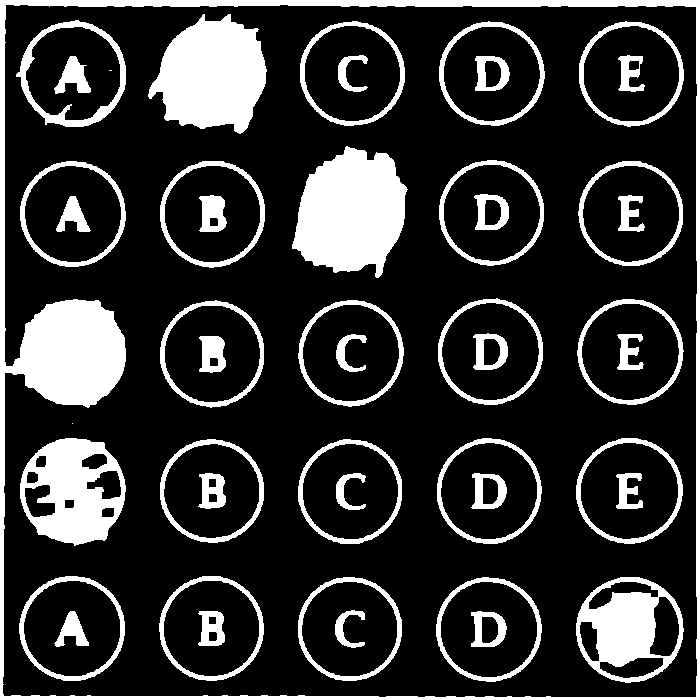


Figure 74: Image closed with kernel size 5x5

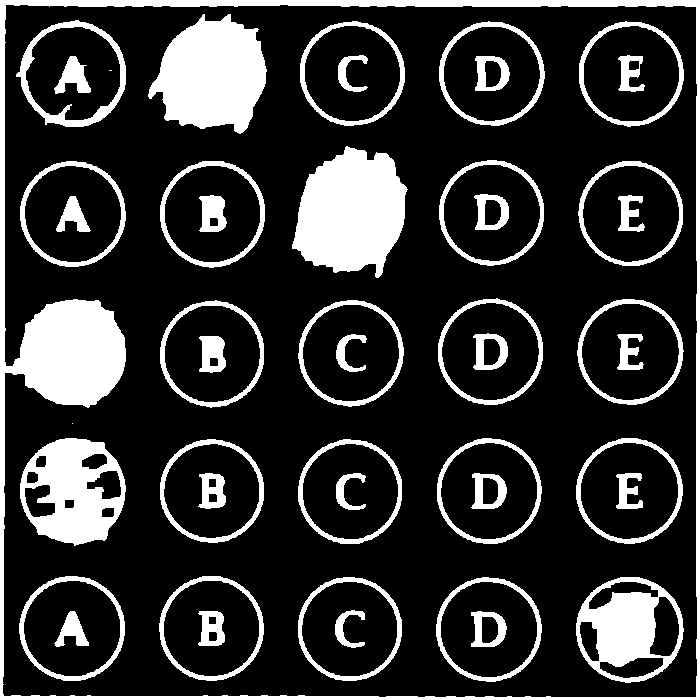


Figure 75: Image closed with kernel sized 7x7

1. GRADIENT

The value of the output pixel is the maximum value of all pixels in the neighborhood. In a binary image, a pixel is set to 1 if any of the neighboring pixels have the value 1.

Morphological dilation makes objects more visible and fills in small holes in objects. Lines appear thicker, and filled shapes appear larger.

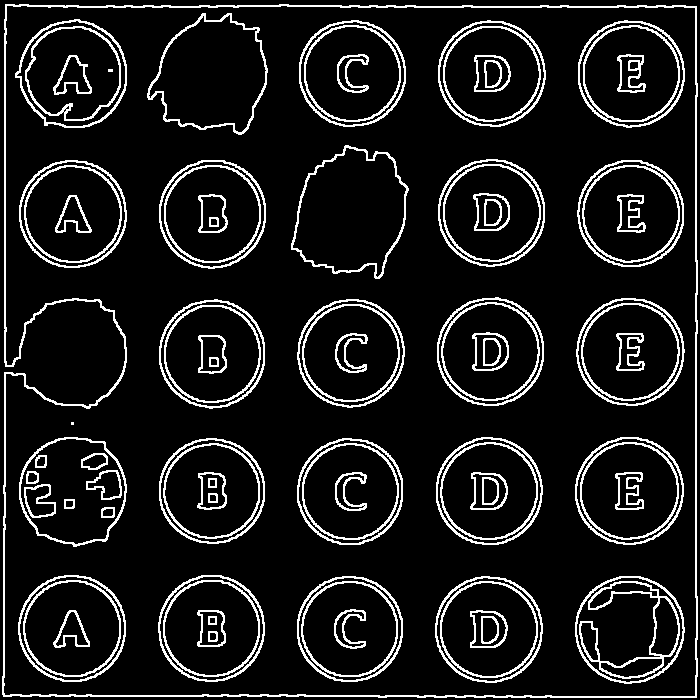
****

Figure 76: Gradient using kernel size 3x3

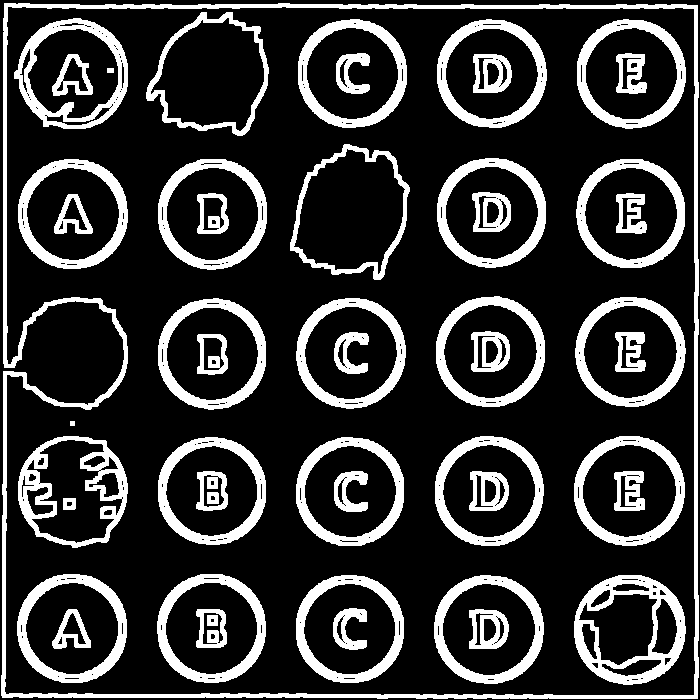


Figure 77: Gradient using kernel size 5x5

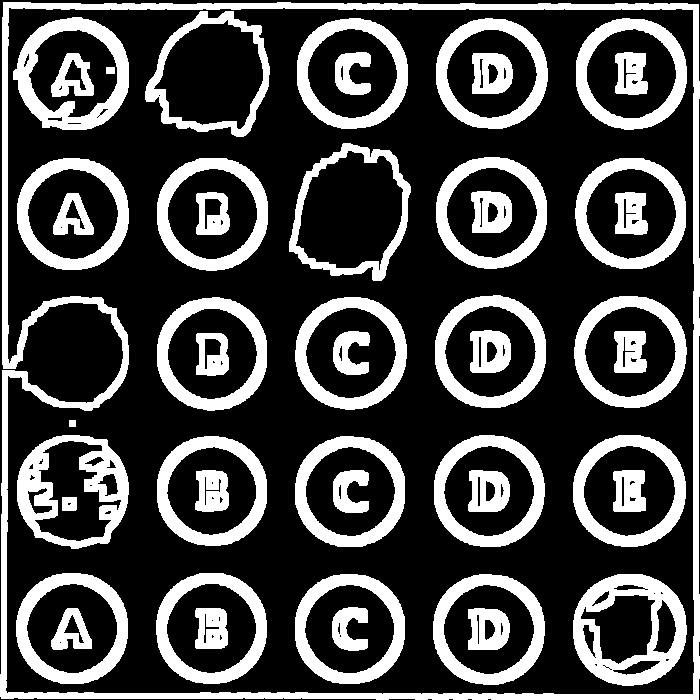


Figure 78: Gradient using kernel size 7x7

1. TOP HAT

Perform a morphological top-hat transform. The top-hat transform opens an image, then subtracts the opened image from the original image.

The top-hat transform can be used to enhance contrast in a grayscale image with nonuniform illumination. The transform can also isolate small bright objects in an image. It is used for Feature Extraction.



Figure 79: Top-hat using kernel size 3x3

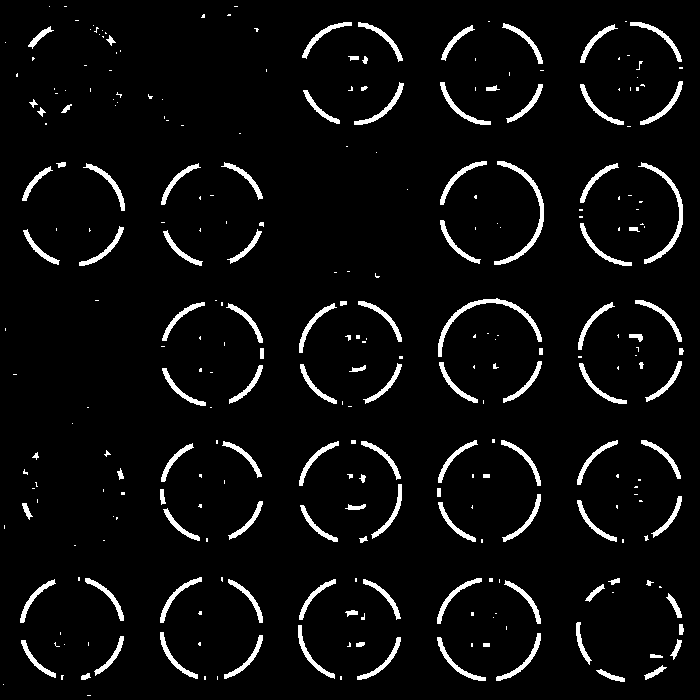


Figure 80: Top-hat using kernel size 5x5

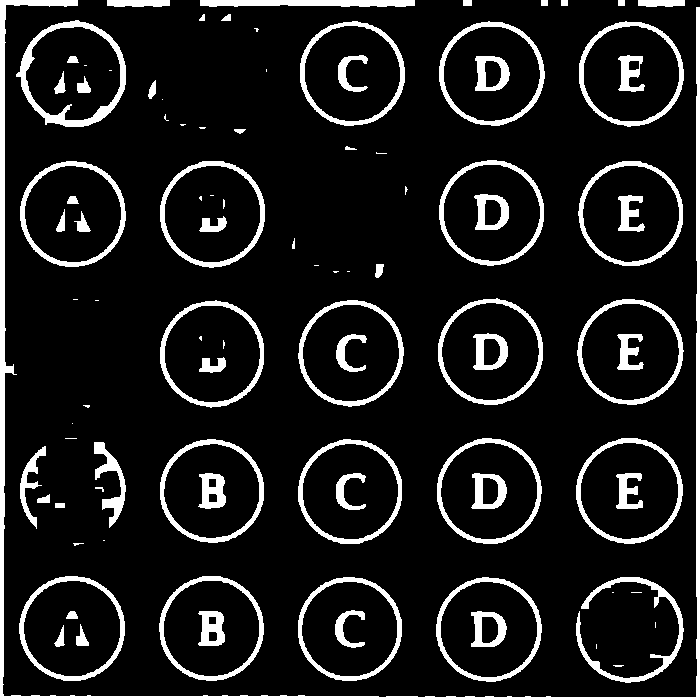


Figure 81: Top-hat using kernel size 13x13

1. BLACK HAT

Perform a morphological black-hat transform. The black-hat transform closes an image, then subtracts the original image from the closed image.

The black-hat transform isolates pixels that are darker than other pixels in their neighborhood. Therefore, the transform can be used to find intensity troughs in a grayscale image. It is used to enhance dark objects of interest in a bright background.

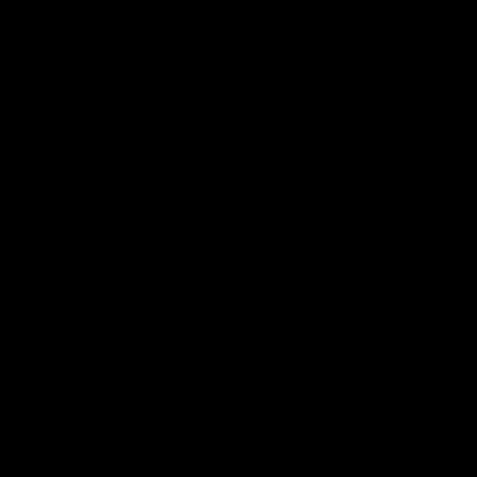
****

Figure 82: Black-hat using kernel size 5x5

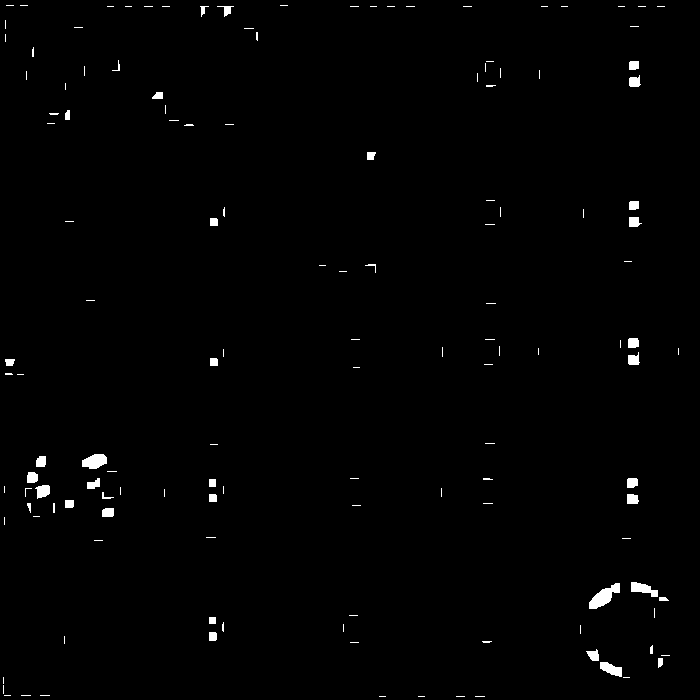


Figure 83: Black-hat using kernel size 11x11

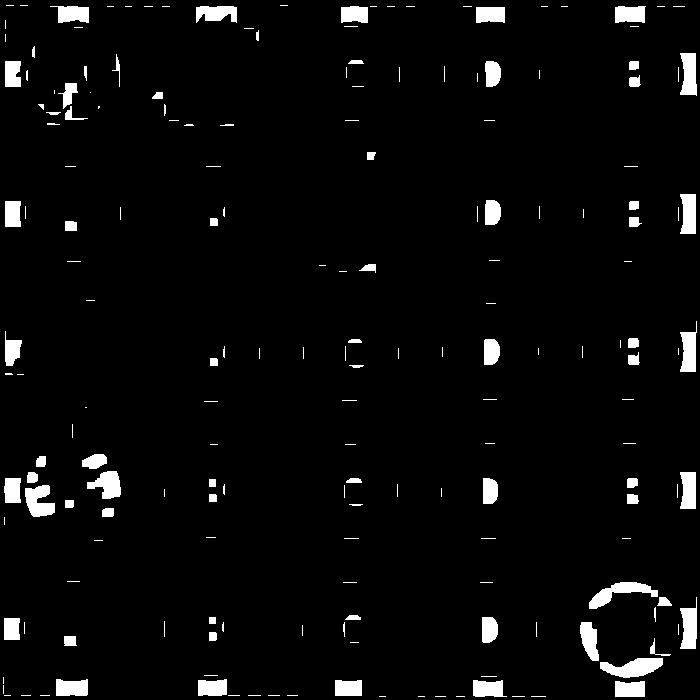


Figure 84: Black-hat using kernel size 17x17

For our problem we use image closed with a kernel of size 7x7.

**7. REGION SPLITTING**

To look through every option on the sheet and check which option has been marked, we divide the sheet into larger number of segments, i.e., as many as [number of question x number of option].

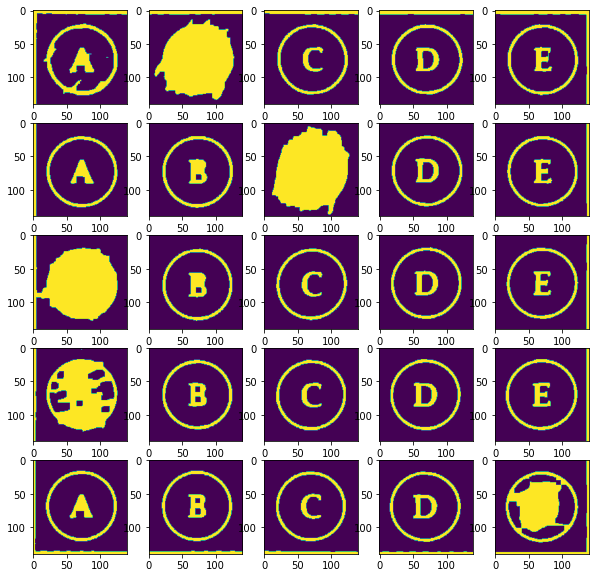
****

Figure 85: Image split into 25 sub-images

Number of white pixels in every sub-image is calculated. The image with highest number of white pixels is the option that has been marked by the user. This information is stored in an array.

A solution array is given as input which contains the correct answers for corresponding question. According to this array, the sheet is colored, i.e., option is colored green if the answer marked is correct, and option is colored red if the answer marked is incorrect.

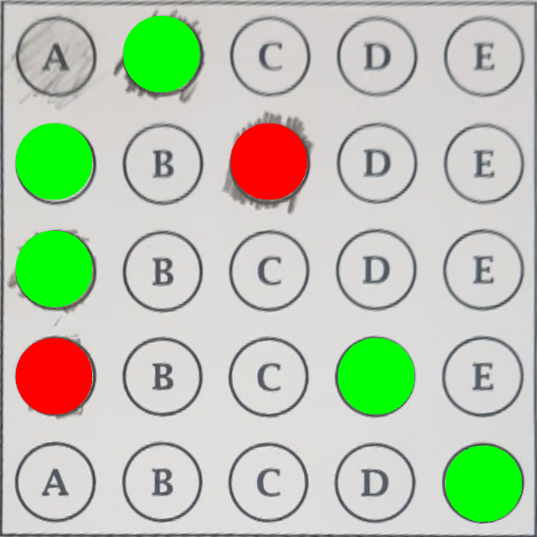


Figure 86: Image with feedback

The background of this image is changed to black and the image’s perspective is transformed to the original perspective. The feedback image is super imposed on the original image which results in the colored feedback to appear on the original image.

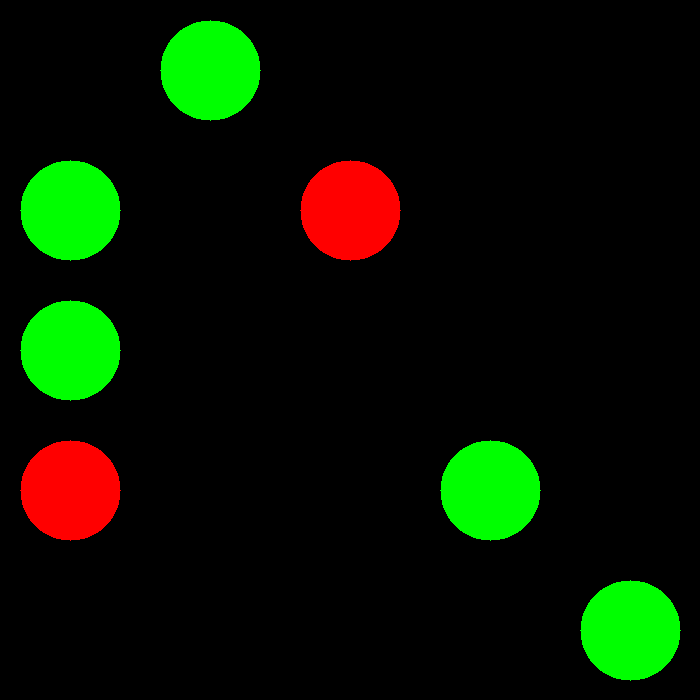


Figure 87: Feedback on black background

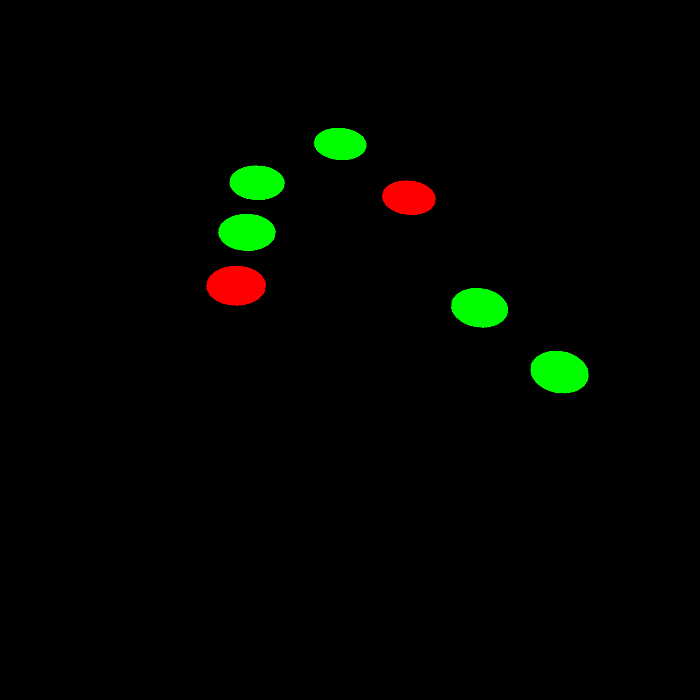


Figure 88: Perspective transformed to original perspective

Similarly, the grade box is segmented out of the original image and its perspective is changed.

The solution array is compared with the array made while counting the number of white pixels in every cell and the result is calculated in percentage.

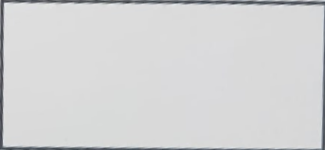


Figure 89: Image of grade box

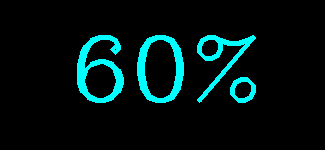


Figure 90: Grade box with the score printed on it



Figure 91: Image perspective changed to original perspective

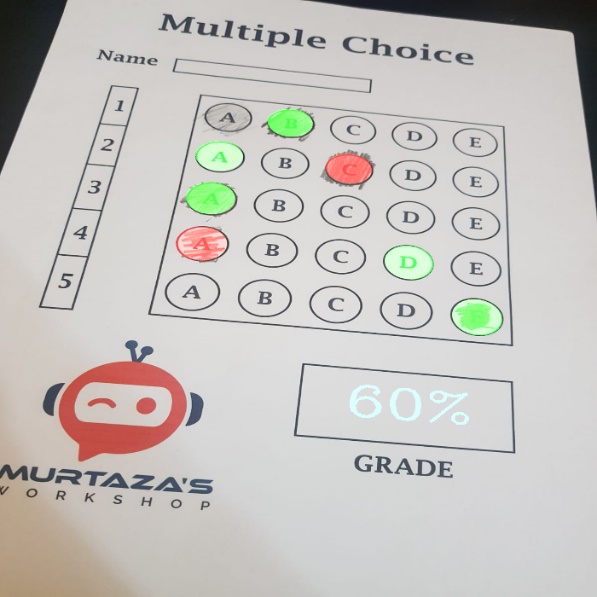


Figure 92: Final result with feedback and score

**8. RESULT AND CONCLUSION**

Optical mark recognition sheet was scanned and checked for required input using OpenCV in python. The input image was preprocessed and processed using various methods in OpenCV to obtain the results and the comparison and results of all methods used are shown. This method is more efficient than the traditional scanners which scan the input and detect the marked regions on basis of amount of light absorbed by the marked regions. Image scanning does not depend on absorption of light and works on the input image hence producing more accurate results.

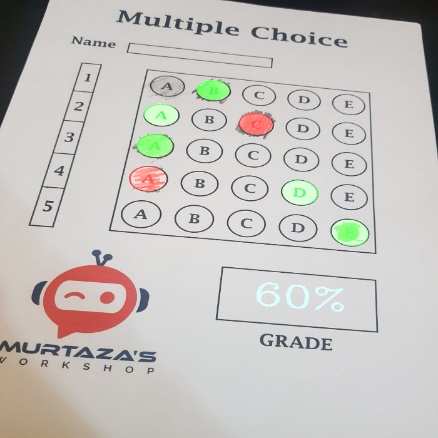


Figure: Final output image with marked feedback and score.

**6. REFERENCES**

1. <https://www.geeksforgeeks.org/python-opencv-morphological-operations/>
2. <https://www.geeksforgeeks.org/perspective-transformation-python-opencv/>
3. <https://analyticsindiamag.com/a-guide-to-different-types-of-noises-and-image-denoising-methods/#:~:text=Poisson%20noise%20is%20produced%20by,value%2C%20this%20expression%20is%20utilized>.
4. <https://www.geeksforgeeks.org/matlab-butterworth-lowpass-filter-in-image-processing/>
5. <https://www.geeksforgeeks.org/matlab-ideal-lowpass-filter-in-image-processing/>
6. <https://analyticsindiamag.com/a-guide-to-different-types-of-noises-and-image-denoising-methods/>