

Crack Segmentation on Road Surfaces Using Gabor Filters, Canny Edge Detection, and Morphological Operations

Yasmin Farisah Salma 13522140^{1,2}

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia

¹13522140@std.stei.itb.ac.id, ²yasminfalsalma@gmail.com

Abstract—Road surface crack detection is essential for effective infrastructure maintenance and traffic safety. This paper presents an automated crack segmentation framework that integrates Gabor filters, Canny edge detection, and morphological operations. Image quality is first enhanced using Contrast Limited Adaptive Histogram Equalization (CLAHE) and non-local means denoising. Multi-orientation Gabor filters with different frequencies are then applied to capture crack patterns in various directions, while Canny edge detection extracts structural boundaries. The outputs of both methods are fused using adaptive thresholding and logical operations to improve crack localization. Morphological processing is used to suppress noise and refine crack regions, followed by geometric filtering to distinguish true cracks from non-crack objects. The system also estimates crack coverage and produces visual overlays of detected defects. Experimental results show that the proposed multi-stage approach achieves accurate crack segmentation with reduced false detections, demonstrating its suitability for automated pavement inspection.

Keywords—Canny edge detection, crack segmentation, Gabor filters, image segmentation

I. INTRODUCTION

Road surface crack detection plays a pivotal role in infrastructure maintenance, as early and accurate identification of cracks helps reduce repair costs and improve traffic safety by preventing further pavement deterioration. Traditional manual inspection methods are time-consuming, subjective, and labor-intensive, driving research toward automated visual inspection techniques. Early image processing methods such as thresholding, edge detection, and morphological operations have been widely applied due to their simplicity and interpretability in segmenting crack features from background pavement texture. For instance, morphological operators like dilation and erosion are used to refine binary crack segmentation and suppress noise artifacts in crack extraction pipelines (Hou et al., 2021).

Edge detection techniques are fundamental in highlighting discontinuities in intensity that correspond to

crack boundaries. The Canny edge detector, in particular, remains popular for cracked surface segmentation because it balances edge sensitivity with noise suppression through Gaussian smoothing followed by gradient-based edge localization (Hou et al., 2021). Complementary feature extraction methods such as Gabor filters have also been investigated in crack detection research. Gabor filters act as orientation-selective bandpass filters that effectively extract multi-directional texture patterns associated with cracks and have been shown to improve detection performance over simple gradient-based operators (Abed & Alhamdani, 2024).

Despite these advances, conventional single-method approaches often struggle with complex pavement conditions characterized by varying illumination, crack orientations, and noise, necessitating hybrid and multi-stage frameworks. Research has demonstrated that combining preprocessing enhancement techniques (e.g., Contrast Limited Adaptive Histogram Equalization) with multi-orientation filtering and edge detection can yield more robust crack segmentation results compared to standalone methods (Wang et al., 2019). Additionally, post-processing with morphological operations and geometric filtering is frequently applied to refine segmentation outcomes and eliminate non-crack artifacts.

A multi-stage automated crack segmentation framework is introduced that combines Gabor filtering, Canny edge detection, and morphological processing. This integrated strategy overcomes the limitations of individual techniques by exploiting their complementary capabilities: Gabor filters extract multi-orientation texture patterns, Canny detection identifies precise edge boundaries, and morphological operations enhance the structural continuity of crack regions. The framework further incorporates image enhancement and geometric filtering to reduce false detections and provide reliable crack coverage measurements, making it suitable for practical pavement inspection applications.

II. THEORY

A. Image Processing in Pavement Crack Detection

Pavement crack detection has become an important topic in road infrastructure management because governments and engineers need faster, more accurate, and more affordable ways to monitor road conditions. Checking roads manually is still widely used, but it takes a lot of time, requires many workers, and often gives inconsistent results depending on who is doing the inspection. Because of these limitations, many researchers have turned to computer vision and image processing techniques to automatically find and analyze cracks in road surface images. These methods work by analyzing pixel intensity, patterns, and spatial relationships in an image to separate cracked areas from the surrounding pavement. As explained by Badhon et al. (2024), early image processing steps such as converting images to grayscale, applying histogram equalization, and using thresholding are commonly used to clean the image and improve contrast so that cracks become easier to detect, especially when lighting conditions vary.

A wide variety of image-processing-based techniques have been developed for pavement crack detection. These methods range from simple approaches like thresholding and edge detection to more advanced techniques involving filtering and morphological operations. Edge detection methods focus on identifying sharp changes in intensity, which usually correspond to crack edges on the pavement surface. At the same time, filtering techniques are used to remove noise, while morphological operations help refine the shape and continuity of crack regions. According to Choudhury (2025), combining several of these image processing techniques often leads to better detection results than using just one method, particularly when dealing with complex road textures, shadows, or uneven lighting. This shows that hybrid image processing pipelines are more robust and practical for real-world pavement inspection systems.

B. Gabor Filters for Crack Texture Extraction

Gabor filters are widely used in image processing because they are very good at analyzing textures and repeated patterns in images. Basically, a Gabor filter combines a wave-like pattern with a Gaussian shape, which allows it to respond strongly to certain frequencies and directions in an image. This makes it especially useful for detecting cracks on road surfaces, since cracks usually appear as long, thin structures that run in different directions and have different widths. In pavement crack detection, researchers often use a set of Gabor filters, called a Gabor filter bank, where each filter is tuned to a specific orientation and scale. As discussed by Nguyen (2022), this approach helps highlight crack patterns while suppressing the background texture of the pavement, making the cracks stand out more clearly and easier to segment in later processing stages.

Compared to simpler methods such as basic gradient or

edge operators, Gabor filtering has the advantage of being sensitive not only to edges but also to the orientation and frequency of textures, which is similar to how the human visual system perceives patterns. This allows the algorithm to better distinguish cracks from rough or uneven road surfaces that may also contain strong edges. According to Fawzy (2024), choosing the right combination of frequencies and orientations in a Gabor filter bank is important because it determines how well cracks of different sizes and directions can be detected. Using more orientations usually improves the ability to capture cracks running in many directions, although it also increases the amount of computation required.

C. Edge Detection: Canny Algorithm and Structural Boundaries

Edge detection is one of the most important steps in computer vision, especially for crack detection, because it helps identify where sharp changes in pixel intensity occur. These changes usually represent physical boundaries, such as the edges of cracks on a pavement surface. Among the many edge detection methods available, the Canny edge detector is one of the most popular because it is designed to find clear and accurate edges while reducing noise. As explained by Hou et al. (2021), the Canny algorithm works through several stages, including smoothing the image with a Gaussian filter to remove noise, calculating the gradient to find strong intensity changes, applying non-maximum suppression to thin out the edges, and finally using hysteresis thresholding to connect meaningful edge segments. This multi-step process allows Canny to produce cleaner and more precise edges compared to simpler operators like Sobel or Prewitt.

In pavement crack detection, Canny edge detection is very useful because cracks usually appear as thin lines where the intensity changes sharply between the damaged area and the intact road surface. However, pavement images often contain a lot of texture, shadows, and lighting variations, which can cause the Canny detector to produce broken or noisy edges. For this reason, relying on edge detection alone is usually not enough to get a clean crack segmentation. Choudhury (2025) points out that combining Canny edge detection with other image processing techniques, such as filtering and morphological operations, helps connect fragmented edges and produce more continuous and reliable crack structures.

D. Morphological Operations for Crack Refinement

Mathematical morphology is a group of image processing techniques that focuses on the shape and structure of objects in an image. These operations usually work on binary or grayscale images and analyze how pixels are connected to each other. The main morphological operations include erosion, dilation, opening, and closing, where opening is a combination of erosion followed by dilation, and closing is the opposite.

According to Hou et al. (2021), these techniques are very useful for cleaning up segmentation results because they can remove small unwanted noise, close gaps in crack lines, and make the detected crack patterns more continuous and visually clear.

In practical pavement crack detection systems, morphological operations are usually applied after an initial step such as edge detection or threshold-based segmentation. Erosion helps remove small noisy pixels that do not belong to actual cracks, while dilation is used to expand and connect broken or disconnected crack segments. When combined as opening and closing operations, these methods can smooth the boundaries of cracks and fill in small holes or gaps within the crack regions. Hou et al. (2021) explain that this refinement process greatly improves the overall quality of crack segmentation, especially when the images are affected by rough pavement textures, shadows, or uneven lighting, which might otherwise lead to false detections or broken crack structures.

E. Integrated Approaches and Practical Considerations

In real-world pavement crack detection, relying on just one image processing technique is usually not enough because road images are affected by many factors, such as complex surface textures, different crack directions, uneven lighting, and noise. Because of this, researchers have found that using a combination of several methods in a multi-stage framework works much better. These frameworks typically include steps like image preprocessing, multi-directional filtering, edge detection, and morphological refinement. Choudhury (2025) explains that when Gabor filtering is combined with edge detection and then refined using morphological operations, the resulting crack segmentation becomes more continuous and more accurate than when using simple thresholding or edge detection by itself.

These integrated approaches are also more effective at detecting different types of cracks, such as longitudinal cracks that run along the road, transverse cracks that cross it, and more complex patterns like alligator cracking. By combining multiple techniques, the system can better separate real cracks from background textures, shadows, or other visual disturbances. However, the overall performance still depends heavily on how well the parameters are chosen at each stage. As pointed out by Cubero-Fernandez et al. (2017), settings such as the frequency and orientation of Gabor filters, the Gaussian smoothing level in the Canny detector, and the size of structuring elements in morphological operations must be carefully tuned to match the imaging conditions. Proper parameter selection helps ensure that the crack detection system works reliably across different pavement surfaces and lighting environments.

III. IMPLEMENTATION

The proposed crack detection system is implemented in Python using OpenCV, scikit-image, and NumPy libraries. The implementation follows a modular architecture with a CrackDetector class that encapsulates all processing stages.

A. System Architecture

The system is structured into three main components:

1. Core Detection Module (crack_detector.py): Contains the CrackDetector class implementing all image processing algorithms
2. Command-line Interface (main.py): Provides batch and single-image processing capabilities
3. Web Interface (app.py): Offers interactive parameter tuning through a Streamlit-based dashboard

B. Preprocessing Stage

Image quality enhancement is performed using CLAHE and non-local means denoising to normalize illumination and reduce noise:

```
def preprocess_image(self, image):
    gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)

    clahe = cv2.createCLAHE(clipLimit=2.0,
tileGridSize=(8, 8))
    enhanced = clahe.apply(gray)

    denoised =
cv2.fastNlMeansDenoising(enhanced, None,
h=10,

templateWindowSize=7,

searchWindowSize=21)
    return denoised
```

C. Multi-Orientation Gabor Filtering

Gabor filters are applied across multiple frequencies (0.05, 0.1, 0.15) and orientations (0°, 45°, 90°, 135°) to capture crack patterns:

```
def apply_gabor_filters(self, image):
    gabor_responses = []
    for frequency in self.gabor_frequencies:
        for theta in self.gabor_angles:
            filt_real, filt_imag =
gabor(gray_normalized,

frequency=frequency,

theta=theta,

sigma_x=3, sigma_y=3)
            gabor_magnitude =
np.sqrt(filt_real**2 + filt_imag**2)

gabor_responses.append(gabor_magnitude)
```

```

gabor_combined = np.mean(gabor_responses,
axis=0)
return gabor_combined

```

D. Edge Detection and Fusion

Canny edge detection is applied with Gaussian blur preprocessing, and results are combined with Gabor outputs using adaptive thresholding:

```

def apply_canny_edge_detection(self, image,
low_threshold=50,
high_threshold=150):
    blurred = cv2.GaussianBlur(gray, (5, 5),
1.4)
    edges = cv2.Canny(blurred, low_threshold,
high_threshold)
    return edges

```

The fusion strategy employs adaptive thresholding based on statistical measures:

```

mean_val = np.mean(gabor_result)
std_val = np.std(gabor_result)
adaptive_threshold = min(mean_val + 1.0 *
std_val, 180)
_, gabor_adaptive =
cv2.threshold(gabor_result,
adaptive_threshold,
255,
cv2.THRESH_BINARY)

combined = cv2.bitwise_and(gabor_adaptive,
canny_result)
combined = cv2.bitwise_or(combined,
gabor_adaptive)

```

E. Morphological Refinement and Geometric Filtering

Sequential morphological operations refine crack structures by removing noise and connecting fragmented segments:

```

def apply_morphological_operations(self,
binary_image):
    kernel_open =
cv2.getStructuringElement(cv2.MORPH_ELLIPSE,
(2, 2))
    opened = cv2.morphologyEx(binary_image,
cv2.MORPH_OPEN, kernel_open)

    kernel_close =
cv2.getStructuringElement(cv2.MORPH_RECT, (3,
3))
    closed = cv2.morphologyEx(opened,
cv2.MORPH_CLOSE, kernel_close)

    filtered =
self.filter_crack_structures(closed)

    kernel_dilate =
cv2.getStructuringElement(cv2.MORPH_RECT, (2,
2))
    dilated = cv2.dilate(filtered,
kernel_dilate, iterations=2)
    return dilated

```

Geometric filtering removes non-crack objects based on aspect ratio and area constraints:

```

def filter_crack_structures(self,
binary_image):
    num_labels, labels, stats, _ =
cv2.connectedComponentsWithStats(
cleaned, connectivity=8)

    for i in range(1, num_labels):
        width = stats[i, cv2.CC_STAT_WIDTH]
        height = stats[i, cv2.CC_STAT_HEIGHT]
        area = stats[i, cv2.CC_STAT_AREA]
        aspect_ratio = max(width, height) /
(min(width, height) + 1e-5)

        if aspect_ratio > 1.8 and area > 10:
            filtered[labels == i] = 255

```

F. Crack Coverage Estimation

The system quantifies crack severity by calculating the percentage of cracked pixels relative to the total image area:

```

def calculate_crack_percentage(self,
binary_result):
    total_pixels = binary_result.shape[0] *
binary_result.shape[1]
    crack_pixels = np.sum(binary_result > 0)
    percentage = (crack_pixels /
total_pixels) * 100
    return percentage

```

A threshold of 3% crack coverage is used to classify images as cracked or non-cracked, providing a binary decision for automated inspection systems.

IV. RESULTS & DISCUSSION

A. Experimental Setup

The system was evaluated on a dataset containing both positive (cracked) and negative (non-cracked) road surface images. Default parameters were set as follows: Gabor frequencies [0.05, 0.1, 0.15], orientations [0°, 45°, 90°, 135°], Canny thresholds (50, 150), and morphological kernel sizes of 2-3 pixels. The crack detection threshold was set at 3% coverage.

B. Detection Performance

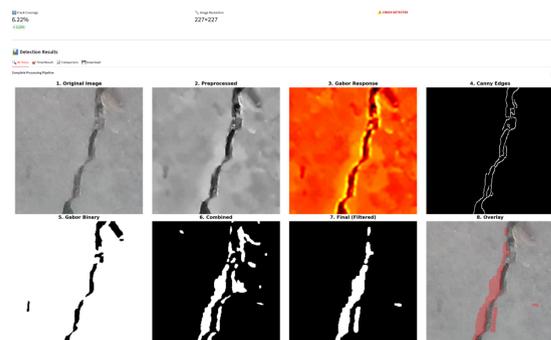


Figure 1 Result

Figure 1 presents the complete crack detection pipeline applied to a sample pavement image with a resolution of 227×227 pixels. In this example, the system estimated a crack coverage of 6.22%, which shows that it was able to successfully detect a clear longitudinal crack on the road surface. This figure is useful because it visually demonstrates how the raw pavement image is gradually transformed into a clean and accurate crack segmentation through a series of processing steps.

The process begins with the preprocessing stage, where CLAHE (Contrast Limited Adaptive Histogram Equalization) is applied to enhance the contrast of the image. This step is very important because pavement images often have uneven lighting and low contrast, which can make cracks difficult to see. By increasing the contrast locally, CLAHE makes the crack more distinguishable from the surrounding road texture.

Next, the Gabor filter response is displayed as a heat map. In this stage, the filters respond strongly to elongated and textured patterns that match the shape of cracks. The bright areas along the crack path show that the Gabor filters are successfully capturing the crack's direction and structure across multiple orientations. This helps separate crack-like patterns from random pavement textures.

After that, the Canny edge detection stage highlights the boundaries of the crack. This step focuses on finding sharp intensity changes, which usually correspond to the edges of cracks. At the same time, Canny suppresses a lot of unnecessary background texture, allowing the main crack outline to stand out more clearly.

The Gabor binary image is then produced using Otsu thresholding. This converts the Gabor response into a binary segmentation where crack candidates are separated from the background. However, at this stage, some noise and unwanted regions are still present, which is normal in early segmentation steps.

To improve the result, the Gabor output is combined with the Canny edge map. This fusion step strengthens the crack structure by keeping regions that are supported by both texture and edge information. As a result, the crack becomes more continuous and better defined than when using either method alone. In the final filtered stage, morphological operations and geometric filtering are applied. These operations remove small noisy regions, fill gaps, and connect broken parts of the crack. The geometric filtering further removes objects that do not match the expected shape of cracks, such as small round spots or irregular background patterns, while keeping the long, thin crack structures. Finally, the detected crack is overlaid in red on the original pavement image. This overlay clearly shows that the detected crack aligns well with the real crack on the road surface, confirming that the system is able to locate and segment pavement cracks accurately. Overall, Figure 1

demonstrates how each processing step contributes to turning a noisy road image into a clean and reliable crack detection result.

C. Strengths of the Multi-Stage Approach

One of the main strengths of this crack detection system is that it does not rely on just one technique, but instead combines several methods that work well together. Gabor filters are very good at capturing the texture of cracks, especially because cracks can appear in many different directions and sizes. At the same time, the Canny edge detector focuses on finding sharp boundaries, which helps locate the exact edges of cracks. When these two methods are used together, the system becomes much more reliable than using either one alone, especially when the pavement has uneven lighting or complex surface patterns.

After the cracks are detected, morphological operations are applied to clean up the results. These operations help connect broken crack lines, remove small noisy pixels, and make the crack shapes look smoother and more continuous. This makes the cracks easier to analyze later. In addition, geometric filtering is used to examine the shape of detected objects. By checking things like aspect ratio and area, the system can tell the difference between long, thin crack shapes and round or irregular objects such as stones, stains, or shadows on the pavement. This step greatly reduces false detections, which are a common problem in simpler crack detection methods.

D. Limitations and Challenges

Even though the system performs well overall, it still has some weaknesses. One major issue is that the results depend heavily on how the parameters are chosen. If the Gabor filter settings or Canny thresholds are not well tuned, the system may either miss real cracks or detect too much noise. Images with strong shadows, very bright or dark lighting, or rough pavement textures can also confuse the system, causing it to detect things that are not actually cracks or to break real cracks into smaller pieces.

Another limitation is that the crack coverage threshold, which is currently fixed at 3%, may not work equally well for all types of pavement or image resolutions. Some roads might naturally have more texture, while others may show cracks more clearly, so a single threshold may not always be accurate. In addition, the system only measures how much area is covered by cracks. It does not yet identify different crack types such as longitudinal, transverse, or alligator cracking, nor does it measure how severe the cracks are. These details are important for deciding what kind of maintenance a road really needs.

E. Experimental Setup

The system is able to produce crack segmentation results that are useful for automated pavement inspection, especially because it provides numerical crack coverage values that can be used to judge road condition more

objectively. The Streamlit web interface makes the system easy to use, allowing users to adjust parameters and immediately see how the results change. This is especially helpful for users who do not have a strong background in image processing. The processing speed is good enough for analyzing batches of images, although further optimization would be needed for real-time video-based inspection.

In the future, the system could be improved in several ways. For example, parameters could be adjusted automatically based on image quality and lighting conditions instead of being set manually. The system could also be expanded to classify different crack types and estimate their severity, which would make it more useful for real maintenance planning. Adding deep learning methods for post-processing could further improve accuracy and reduce false detections, especially in difficult environments. Finally, integrating the system into mobile inspection devices would allow it to be used directly in the field for large-scale road monitoring.

V. CONCLUSION

This work shows that combining several image processing techniques into one multi-stage system can be very effective for detecting cracks on road surfaces. By using Gabor filters, the system is able to capture the texture of cracks in different directions and sizes, which is important because cracks do not always appear in the same shape or orientation. The Canny edge detector then helps find the exact boundaries of these cracks, making their shapes clearer. After that, adaptive thresholding and morphological operations are used to clean up the results by removing small noise, connecting broken crack segments, and producing smoother and more continuous crack patterns. Geometric filtering further improves the detection by separating true cracks from other objects on the road, such as stones, stains, or shadows, based on their shape and size.

One of the useful outputs of this system is crack coverage, which gives a numerical indication of how much of the road surface is damaged. This makes it easier to objectively evaluate road conditions and support maintenance planning. The system is also designed to be practical, since it can be used through both a command-line interface and a web-based dashboard. With the web interface, users can change parameters in real time and immediately see how the results change, making the system easier to use even for people without a strong technical background.

However, the system is not perfect. Its performance depends a lot on choosing the right parameters, and it can struggle when images have strong shadows, extreme lighting, or very rough pavement textures. These conditions may cause false detections or incomplete crack segmentation. In the future, the system could be improved by automatically adjusting parameters based on the input image, classifying different types of cracks, and estimating their severity more accurately. Adding deep

learning methods could also help make the detection more robust in complex real-world environments. Overall, this multi-stage approach shows strong potential for supporting automated and efficient road inspection and maintenance.

VI. ACKNOWLEDGEMENT

This paper would not have been brought to fruition without the resources provided by the IF4073 Digital Image Processing class with the guidance of Dr. Ir. Rinaldi Munir, M.T. as the lecturer of the author in class.

The author would like to thank the developers of OpenCV, scikit-image, and Streamlit for providing open-source libraries that made this implementation possible. Gratitude is also extended to other brilliant researchers in the field of pavement crack detection for sharing their knowledge and findings, without which this paper would not have been complete.

REFERENCES

- [1] A. Cubero-Fernandez, F. J. Rodriguez-Lozano, R. Villatoro, J. Olivares, and J. M. Palomares, "Efficient pavement crack detection and classification," **EURASIP Journal on Image and Video Processing**, vol. 2017, art. no. 39, Jun. 2017.
- [2] D. Choudhury, "Highway Pavement Crack Detection Using Image Processing," **International Journal of Research in Modern Engineering and Emerging Technology (IJRMEET)** Dec. 2025.
- [3] H. I. Abed and I. M. Alhamdani, "A Review of Road Cracks Detection and Classification Based Image Processing and Intelligent Techniques," **Journal of Education and Science (JES)**, vol. 33, no. 3, pp. 45–57, Sep. 2024, doi: 10.33899/edusj.2024.148454.1443.
- [4] M. M. Fawzy, A. S. El Shrakawy, A. A. Hassan, and Y. A. Khalifa, "Enhancing sustainability for pavement maintenance decision-making through image processing-based distress detection," **Innovative Infrastructure Solutions**, vol. 9, art. no. 58, Feb. 2024.
- [5] S. M. S. I. Badhon, S. A. Khushbu, N. C. Saha, A. H. Anik, M. A. Ali, and K. S. M. T. Hossain, "Explainable AI for Skin Disease Classification Using Grad-CAM and Transfer Learning to Identify Contours," **Preprints**, Jul. 2024, doi: 10.20944/preprints202407.2556.v1.
- [6] T. H. N. Nguyen, "Crack Detection and Classification using Digital Image Processing." Ph.D. dissertation, Faculty of Engineering and IT, University of Technology Sydney, Sydney, Australia, Dec. 2022.
- [7] W. Wang, M. Wang, H. Li, H. Zhao, K. Wang, C. He, J. Wang, S. Zheng, and J. Chen, "Pavement crack image acquisition methods and crack extraction algorithms: A review," **Journal of Traffic and Transportation Engineering (English Edition)**, vol. 6, no. 6, pp. 535–556, Dec. 2019, doi: 10.1016/j.jtte.2019.10.001.
- [8] Y. Hou, Q. Li, C. Zhang, G. Lu, Z. Ye, Y. Chen, L. Wang, and D. Cao, "The State-of-the-Art Review on Applications of Intrusive Sensing, Image Processing Techniques, and Machine Learning Methods in Pavement Monitoring and Analysis," **Engineering**, vol. 7, no. 6, pp. 845–856, 2021, doi: 10.1016/j.eng.2020.07.030.

ATTACHMENT

Github: <https://github.com/caernations/surface-crack-detection>

STATEMENT

I hereby declare that this paper is my own original work,
not an adaptation or a translation of someone else's paper,
and that it does not constitute plagiarism.

Bandung, December 24th, 2025,

A handwritten signature in black ink, appearing to read 'Yasmin', with a horizontal line underneath.

Yasmin Farisah Salma, 13522140