Petals to Pixels: An Image Processing Pipeline for Flower Segmentation

Anshana Manoharan
School of Computer Science
University of Nottingham Malaysia
hcyam4@nottingham.edu.my

Anshali Manoharan
School of Computer Science
University of Nottingham Malaysia
hcyam5@nottingham.edu.my

Aravindh Palaniguru
School of Computer Science
University of Nottingham Malaysia
hcyap1@nottingham.edu.my

Varsagasorraj A/L Vasagarajan
School of Computer Science
University of Nottingham Malaysia
hfyvv2@nottingham.edu.my

***Abstract*—An image processing pipeline has been developed for segmentation of flower images, with a focus on isolating the main flower in the focal point of the image. This pipeline leverages the size and contrast characteristics of flower images to accurately identify and segment flowers. Specifically, the pipeline incorporates colour space conversion, gamma correction, foreground extraction, negation, histogram equalisation, Otsu thresholding, region labelling and largest segment selection to preprocess and segment the images. Evaluation using the MioU (Mean Intersection over Union) and Overlap Score (OS) indicates a high level of similarity between the ground truth and predicted segmentation results from the proposed image processing pipeline, reaching 80% and above.**

Keywords—flower, segmentation, foreground extraction, negation, gamma correction, Otsu thresholding, region labelling, processed ground truth

I. Introduction

The task of accurately identifying flower species poses significant challenges, mainly due to the extensive diversity of flower types found globally [1].

With the advancements in computer science, image processing methods have emerged as valuable tools to aid botanists in the analysis of plant images [2]. In our study, we focus on flower image segmentation, which plays an important role in assisting analysis of plant images.

Flower segmentation involves effectively dividing the image into distinct regions corresponding to flowers within an image. The quality of the segmentation greatly influences subsequent tasks such as object recognition and classification [3].

Segmenting flower images obtained from natural environments presents numerous challenges, such as background clutter, variations in brightness, contrast, scale, resolution, orientation [4, 5]. We aim to design an image pipeline which mitigates the impact of these variations and segments out the flower in the focal point of the image accurately.

II. Literature Review

Previous studies in the literature have extensively investigated methods for flower segmentation. Das et al. [6] presented a flower segmentation approach in their study. However, this method begins by deleting pixels from colour classes which are assumed to not represent flowers, such as green, brown, black, and grey. For specific flowers in the Oxford Flower Dataset, this assumption is not valid, as depicted in Fig. 1.

(a) (b)

Fig. 1. Flower images from the Oxford Flower Dataset. (a) shows grey pixels which represent shadows. (b) shows a flower with a green centre.

Najjar et al. [2] proposed a straightforward segmentation approach for isolating flower regions without relying on domain-specific knowledge.

Their method involved converting the RGB image to the Lab colour space, where the components L, a, and b were separated. After pre-processing the image, Otsu thresholding was applied to each component individually.

However, the selection of the optimal segmentation result was guided by a supervised evaluation measure. Although this method is computationally inexpensive, it relies on the presence of ground truth data to choose the best segmentation result.

Another method for extracting flower regions was proposed by Saitoh et al. [7] which involved the use of Intelligent Scissors (IS). This method achieved a flower recognition rate of 90%. However, this method has some shortcomings. It depends on the image containing an unfocused background with the flower in focus, and flower extraction depends on user interaction.

Our study presents a novel flower segmentation approach aimed at addressing the limitations mentioned above. Our algorithm eliminates the need for domain knowledge, reliance on ground truths and the need for user interaction.

The structure of this paper is as follows: Section III outlines the steps of the proposed image pipeline, accompanied by detailed justifications. In Section IV, we

present our evaluation measures, results, and thorough analysis. Finally, we offer insights and conclusions drawn from our work.

III. Methodology

This section begins with a visual representation of the proposed pipeline's steps are depicted in Fig. 2. Then, each step is explored and justified in detail in Sections A to I.

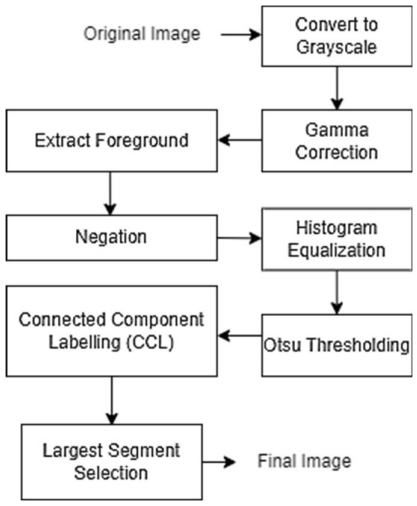


Fig. 2. Ordered steps of the proposed image pipeline.

A. Colour space conversion

The selection of an appropriate colour space holds significance as it directly impacts the outcome of segmentation.

We opted for greyscale, as it reduces the computational requirements by converting a 3 colour-channel image (RGB) into a single channel [8]. This simplification is essential for tasks like edge detection, thresholding, and contour detection, as it streamlines the process. While considering efficiency, processing a single-channel grayscale image is computationally less demanding compared to an RGB image as they require a single intensity value per pixel, compared to the three values for each pixel in an RGB image.

B. Noise reduction

Our image processing pipeline omits noise reduction. Noise reduction techniques such as blurring could compromise the edges and textures that are key to identifying the contours of the flower [9].

By excluding noise reduction, we can preserve the critical high-frequency details that are essential for accurate segmentation of the flower from its background.

C. Gamma correction

According to the study by Goh et al. [10] the key to successful image segmentation depends on object-background intensity difference. Therefore, gamma correction is applied to the converted greyscale image, to improve the contrast between the flower and its background.

This technique non-linearly stretches the pixel intensity values, darkening the darker areas and brightening the lighter areas. This emphasises the differences between foreground and background regions as shown in Fig. 3.

Fig. 3. Effects of gamma correction on greyscale image. (a) before gamma correction. (b) after gamma correction.

This accentuation of the difference between foreground and background aids to extract the foreground in the subsequent steps of the pipeline. However, the choice of value for gamma correction needs to be optimal.

A parameter tuning session was conducted to identify the most optimal value for gamma correction. The average overlap score between the ground truth and the segmented result of all the flower images was computed for each gamma value. As exemplified in Fig. 4, the optimal parameter value was determined to be 4.0.

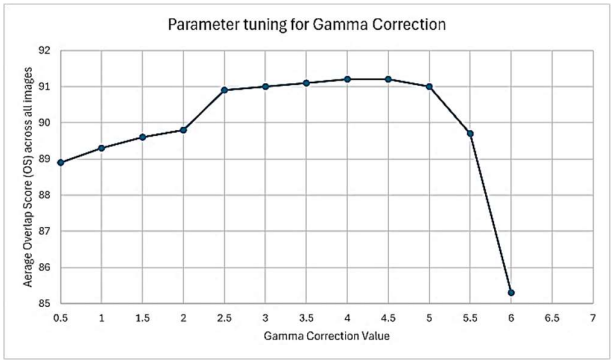


Fig. 4. Graph displaying results of the parameter tuning session for gamma correction.

Nevertheless, there exists a trade-off associated with this optimal gamma correction value. With a high gamma value like 4.0, the darker sections (i.e. shadows) of the flower material tend to darken further, potentially causing some parts of the flower region to be misclassified as background in subsequent steps of the pipeline.

This introduces visual anomalies into the segmented image. Hence, upon visual analysis of the segmented images, and considering both the overlap scores and the presence of these anomalies on the flowers, we found that setting the gamma parameter lower i.e. 3.5 effectively mitigates this issue, as depicted in Fig 5.

(a) (b)

Fig. 5. Effect of two gamma correction values. (a) final segmented image with gamma value 4. (b) final segmented image with gamma value 3.5. Similar observations were made on other flower images.

D. Foreground extraction using GrabCut

After gamma correction, GrabCut is implemented on the image, refer to Fig. 6. As proposed by Rother et al. [11], GrabCut surpasses other image segmentation tools such as Magic Wand, Intelligent Scissors, and GraphCut in terms of directness of user input and quality of results. The superiority in both aspects was why GrabCut was adopted as our foreground extraction tool.

However, the GrabCut technique required the region of interest (ROI), foreground and background regions to be initialised by the user using a bounding box [11]. But since our aim was to eliminate user interaction, we have pre-set the initialisations. This includes setting the ROI as the full dimensions of the image.

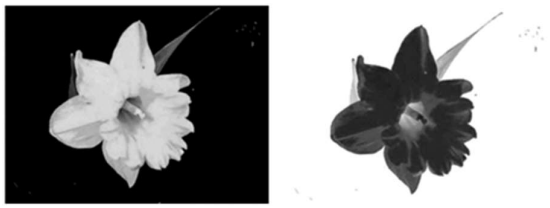
The GrabCut algorithm then iterates between assigning pixels to foreground, background and updating the statistical models of foreground and background regions using the Gaussian Mixture Model (GMM). This process continues until convergence, resulting in a final segmentation mask that delineates foreground objects from the background.

(a) (b)

Fig. 6. Effects of performing GrabCut on the gamma corrected image. (a) before performing GrabCut. (b) after performing GrabCut.

E. Negation

Negation is implemented on the image after extraction of the foreground. Negation is responsible for inverting the intensity values of the image. We used negation to help mitigate the impact of shadows on the flower material. By negating the flower image, we effectively transform these shadows into white areas, as shown in Fig. 7. This aids the subsequent step of the image processing pipeline, histogram equalisation.



(a) (b)

Fig. 7. Effects of negation on the extracted foreground of flower image. (a) before negation. (b) after negation.

F. Histogram equalisation

Histogram equalisation is a technique utilised in image processing to enhance contrast within the spatial domain by using the histogram of the image. This technique is usually implemented to increase the overall contrast of the image and is particularly beneficial for images that are bright or dark [12].

From Section E, it is observed that negation converts shadows on flower material to white areas. Histogram equalization normalises these white areas. This is demonstrated in Fig. 8.

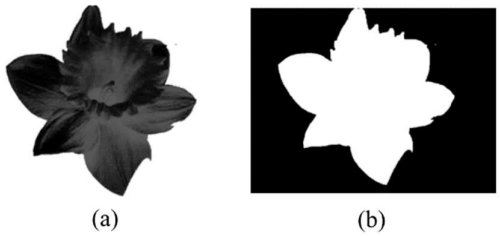
(a) (b) (c) (d)

Fig. 8. Effects of histogram equalisation to two of the negated flower images which have shadows. The images (a) and (c) are before histogram equalisation whereas (b) and (d) are after histogram equalisation.

G. Otsu thresholding

Following histogram equalisation, Otsu thresholding is applied to the image to produce a binary representation. We chose Otsu thresholding due to the success of this technique in prior works [13,14]. The Otsu thresholding is known to give high-quality segmentation results and has a straightforward, easy implementation [2].

The previous pre-processing steps ensured that the flower material is completely distinguishable from the background prior to the application of Otsu thresholding, as shown in Fig. 9.



(a) (b)

Fig. 9. Effects of Otsu thresholding on the equalised image (a) the equalised image (b) the equalised image after applying Otsu thresholding.

H. Region labelling

Region labelling, also known as Connected Component Labelling or Blob Extraction, is an important technique used in image analysis, image understanding, pattern recognition, and computer vision. It is applied on binary images and involves assigning a unique label to each group of connected pixels [15].

In the context of our image processing pipeline, this technique distinguishes the different segments of flower material and labels them. This is demonstrated in Fig. 10.

Fig. 10. CCL performed on the binary representation of processed flower image. Each coloured segment represents a Connected Component (CC) (a) before CCL. (b) after CCL.

Note: The colouring of the segments is to demonstrate the labelling and is not a part of the pipeline.

I. Largest segment selection

Following the implementation of region labelling on the image, an exhaustive calculation of the areas of all labelled regions was conducted. The region with the largest area is retained within the image, while the remaining regions are systematically discarded.

This technique not only effectively eliminates extraneous background flower material but also successfully mitigates any noise that may have infiltrated the foreground. As a result, the primary flower, serving as the focal point of the image, is preserved. The implementation of this technique is depicted in Fig. 11.

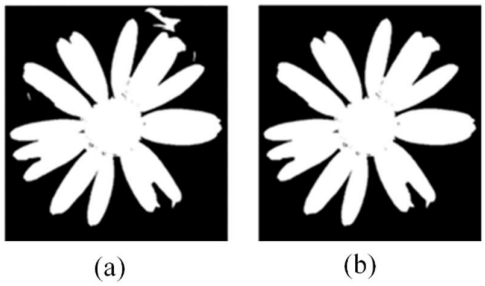


Fig. 11. Extracting the largest segment of the binary image. (a) before largest segment selection. (b) after largest segment selection.

IV. Result and Discussion

The evaluation of our image processing pipeline was done using 9 images from a flower dataset from the Oxford University. The dataset includes various flower images, and their corresponding ground truths.

Section A justifies the processing performed on the ground truths. Section B discusses the evaluation metrics to be used. Lastly, Section C showcases and analyses our results, and the strengths and weaknesses of the proposed pipeline are discussed.

A. Processing ground truths

The interpretation of the ground truth was guided by the study done by Nilsback and Zisserman [16]. This study is referenced in the Oxford flower dataset's README markdown file to aid in the interpretation of the ground truths.

The study explains the ground truth tri-maps, delineating the roles of distinct regions within them. It states that within these tri-maps (refer to Fig. 12), the red region signifies the foreground, while the green region represents the background. The black region is unlabelled and is not utilized for performance evaluation.

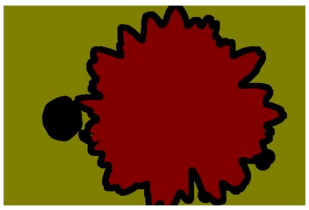


Fig. 12. Sample ground truth from the Oxford flower dataset.

Using this interpretation of the tri-maps, we processed the ground truth images before comparison to the final segmented images. Fig. 13 depicts the processing steps for the ground truths.

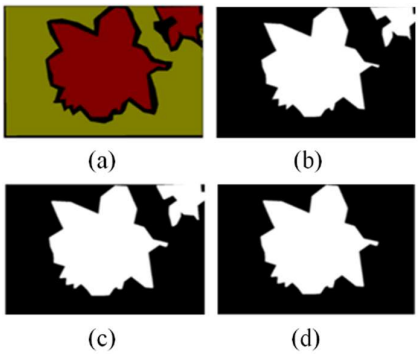


Fig. 13. Ground truth processing steps. (a) original ground truth image. (b) ground truth image after foreground extraction and binary conversion. (c) dilation is applied. (d) only the main flower (largest segment) is extracted. This is the final processed ground truth image used for evaluation.

As our final images from the image processing pipeline are binary images, the ground truth images were converted to binary after extraction of the red region (foreground). Next, dilation was applied to the binary ground truth to improve the similarity to the input image. An example of the effectiveness of dilation on the ground truth image is depicted in Fig. 14.

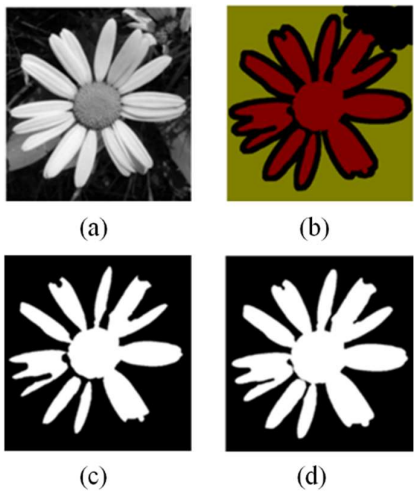


Fig. 14. An example of the effectiveness of dilation on ground truth images. (a) original input image in grayscale for comparison. (b) original ground truth image (c) ground truth image after foreground (red region) extraction and binary conversion. (d) ground truth image after dilation.

After dilation, the final step of ground truth image processing is extracting the largest flower segment from the binary image.

This step is performed as the focus of our image processing pipeline is to extract the flower in the focal point of the image, and background flowers in the ground truth image would interfere with the accuracy.

B. Metrics used for evaluation

The processed ground truth images and the results from the pipeline were compared with three performance metrics: overlap score (OS), mean squared error (MSE) and Mean Intersection over Union (MIoU).

We computed the overlap score (OS) as indicated in (1).

$$OS = \frac{true\ foreground \cap segmented\ foreground}{true\ foreground \cup segmented\ foreground} \quad (1)$$

True foreground indicates the set of pixels that belong to the foreground of the ground truth image. Segmented foreground is the set of pixels that are classified as foreground in the segmentation result obtained from the image processing pipeline algorithm.

We also computed the Mean Squared Error (MSE), the most used tool for estimating image quality measurement, where the values nearer to zero are considered superior [17].

Equation (3) for MSE is referenced from [18]. It measures the mean squared differences between actual ground truth and the segmented output, which is defined as

$$E = \frac{1}{N_v}\sum_{p=1}^{N_v}\left[t_p - y_P\right]^2 \quad (3)$$

In the context of our pipeline, t_p indicates the p^{th} pixel value of the ground truth, and y_p is the p^{th} pixel value of the segmented output. N_v is the total number of pixels, and (t_p-y_p) is the residual error.

Lastly, we computed the Mean Intersection over Union (MIoU), by calculating the Intersection over Union (IoU) for each class (i.e. foreground and background) and computing the mean of the two classes.

C. Analysis of results

Table I. displays the results after applying the above-mentioned evaluation measures.

TABLE I. QUANTITATIVE EVALUATION OF SEGMENTATION ACCURACY

Image Name	Performance Metrics		
	MIoU (%)	***Overlap Score (OS) (%)***	***MSE***
easy_1	94.5	92.8	0.03
easy_2	95.0	93.2	0.02
easy_3	93.9	91.0	0.02
medium_1	86.3	83.1	0.07
medium_2	94.1	93.3	0.03
medium_3	89.4	88.2	0.06
hard_1	92.6	93.0	0.04
hard_2	91.8	91.2	0.04
hard_3	93.9	93.9	0.03

All the images have overlap scores (OS) and MIoU values above 80%. The results show 77% of the images have MIoU values and overlap scores above 90%. Our method gives an average OS of 91%. Additionally, majority of the images have relatively low MSE values, indicating good similarity between the ground truths and processed images.

Despite variations in image difficulty (easy, medium, and hard), the image processing pipeline demonstrates consistent performance, with most images achieving strong segmentation results, highlighting the potential for applications such as flower recognition, where accurate segmentation is crucial for subsequent analysis.

However, there are some drawbacks to our method. In some images, there is a slight drop of accuracy, where the background pixels are similar to the foreground pixels (such as leaves and blurred background flowers). This similarity is maximised by gamma correction, leading to the background pixels being considered as a part of the flower region. Examples of this is highlighted in Fig. 14.

Furthermore, this approach is applicable only to images featuring a prominent large flower. It may not be suitable for images containing clusters of small flowers, as it would solely isolate the largest cluster, disregarding the smaller ones.

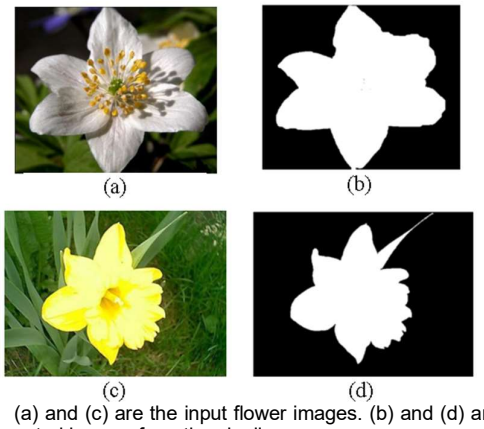


Fig. 14. (a) and (c) are the input flower images. (b) and (d) are the final segmented images from the pipeline.

The strengths of this pipeline include its ability to segment out flower material regardless of interfering light conditions like shadows on flower material. This is shown in Fig. 15.

Fig. 15. (a) is the input image of a flower with shadows whereas (b) is the final segmented image from the pipeline.

Additionally, the pipeline avoids noise-reduction techniques like blurring, and by doing so it accurately preserves important outlines of flower parts such as the stamen (encircled in red). This is shown in Fig. 16.

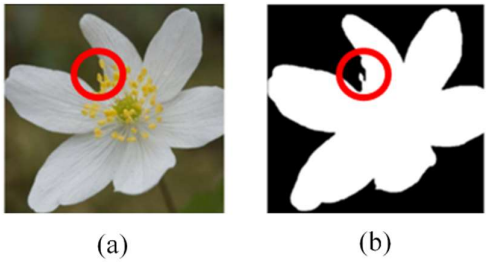


Fig. 15. (a) is the input image whereas (b) is the final segmented image from the pipeline.

CONCLUSION

In conclusion, our study presents a robust image processing pipeline for flower segmentation, using techniques such as colour space conversion, gamma correction, foreground extraction using Grab Cut, negation, histogram equalization, Otsu thresholding, region labelling and largest segment selection.

Our study addresses limitations observed in previous flower segmentation methods documented in the literature. By contrast to the methods reliant on colour-based assumptions, ground truth data, or user interaction, our pipeline offers a more automated approach, with high overlap scores and MioU values, enhancing its applicability across diverse datasets and scenarios especially with interfering light conditions on flower material.

REFERENCES

[1] I. Patel and S. Patel, 'Analysis of Various Image Segmentation Techniques for Flower Images', vol. 6, pp. 1936–1943, 02 2019.

[2] A. Najjar and E. Zagrouba, 'Flower image segmentation based on color analysis and a supervised evaluation', 06 2012.

[3] K. Srikanth and B. Arun Teja, 'FLOWER IMAGE SEGMENTATION: A COMPARISON BETWEEN WATERSHED, MARKER CONTROLLED WATERSHED, AND WATERSHED EDGE WAVELET FUSION', 2006.

[4] K. Prasad, M. V. D. Prasad, R. Prasad, N. Sasikala, and S. Ravi, 'Flower image segmentation using watershed and marker controlled watershed transform domain algorithms', ARPN Journal of Engineering and Applied Sciences, vol. 13, pp. 8441–8455, 01 2018.

[5] M.-E. Nilsback, 'An automatic visual flora-segmentation and classification of flower images', Oxford University, 2009.

[6] M. Das, R. Manmatha, and E. M. Riseman, 'Indexing flower patent images using domain knowledge', IEEE Intelligent Systems and their Applications, vol. 14, no. 5, pp. 24–33, 1999.

[7] T. Saitoh, K. Aoki, and T. Kaneko, 'Automatic recognition of blooming flowers', in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., 2004, vol. 1, pp. 27-30 Vol.1.

[8] C. Kanan and G. W. Cottrell, 'Color-to-grayscale: does the method matter in image recognition?', PloS one, vol. 7, no. 1, p. e29740, 2012.

[9] C. Knaus and M. Zwicker, 'Progressive image denoising', IEEE transactions on image processing, vol. 23, no. 7, pp. 3114–3125, 2014.

[10] T. Y. Goh, S. N. Basah, H. Yazid, M. J. A. Safar, and F. S. A. Saad, 'Performance analysis of image thresholding: Otsu technique', Measurement, vol. 114, pp. 298–307, 2018.

[11] Rother, C., Kolmogorov, V., & Blake, A. (2004). " GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, *23*(3), 309-314.

[12] O. Patel, Y. P. S. Maravi, and S. Sharma, 'A comparative study of histogram equalization based image enhancement techniques for brightness preservation and contrast enhancement', arXiv preprint arXiv:1311. 4033, 2013.

[13] B. Yu, A. K. Jain, and M. Mohiuddin, 'Address block location on complex mail pieces', in Proceedings of the fourth international conference on document analysis and recognition, 1997, vol. 2, pp. 897–901.

[14] E. Prasetyo, R. D. Adityo, N. Suciati, and C. Fatichah, 'Mango leaf image segmentation on HSV and YCbCr color spaces using Otsu thresholding', in 2017 3rd International Conference on Science and Technology-Computer (ICST), 2017, pp. 99–103.

[15] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, 'The connected-component labeling problem: A review of state-of-the-art algorithms', Pattern Recognition, vol. 70, pp. 25–43, 2017.

[16] M.-E. Nilsback and A. Zisserman, 'Delving into the Whorl of Flower Segmentation', in BMVC, 2007, vol. 2007, pp. 1–10.

[17] U. Sara, M. Akter, and M. S. Uddin, 'Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study', Journal of Computer and Communications, vol. 7, no. 3, pp. 8–18, 2019.

[18] J.-M. Muñoz-Ferreras, R. Gómez-García, and C. Li, 'Chapter 5 - Human-aware localization using linear-frequency-modulated continuous-wave radars', in Principles and Applications of RF/Microwave in Healthcare and Biosensing, C. Li, M.-R. Tofighi, D. Schreurs, and T.-S. J. Horng, Eds. Academic Press, 2017, pp. 191–242.