

Image Selection Using Morphological Operation

Data Anggota:

1. Faiz Noor Adhytia		22.K3.0006
2. Vincentius Raditya Agung Soedomo		22.K1.0055
3. Dwita Avrilly		22.K.0049

1. Libraries

Menggunakan 3 *libraries*, yaitu:

- Cv2** → `import cv2`
Library OpenCV (cv2) digunakan untuk melakukan berbagai proses pengolahan citra digital. Fungsinya meliputi membaca, menampilkan, dan menyimpan gambar, melakukan konversi warna, peningkatan kualitas citra, hingga deteksi tepi dan kontur. OpenCV berperan sebagai komponen utama dalam manipulasi data citra.
- Numpy** → `import numpy as np`
Library NumPy (np) berfungsi untuk mendukung perhitungan numerik, khususnya dalam bentuk array dan matriks. Gambar yang diolah OpenCV direpresentasikan sebagai array NumPy, sehingga setiap piksel dapat dimanipulasi melalui operasi matematis. NumPy juga mendukung komputasi aljabar linear dan statistik secara efisien.
- Matplotlib.pyplot** → `import matplotlib.pyplot as plt`
Library Matplotlib digunakan untuk menampilkan hasil pengolahan citra. Melalui pyplot, hasil citra asli maupun hasil operasi dapat divisualisasikan, serta dibandingkan secara langsung. Selain itu, Matplotlib juga memungkinkan pembuatan grafik pendukung analisis.

2. Input Image

Input image dilakukan untuk menghubungkan gambar dari device ke dalam program agar siap diolah menggunakan morphological operation. Pendefinisian gambar dilakukan satu per satu, karena setiap gambar memiliki karakteristik berbeda.

- Pada “*bubbles.jpeg*”, terdapat balon dengan saturasi warna tinggi (pink dan kuning), sehingga morphological operation standar dapat menyebabkan background terbaca bias, dan bagian tengah balon salah terdeteksi sebagai foreground.
- Pada “*coins1.jpeg*”, koin saling berhimpitan dengan tingkat saturasi tinggi (cenderung putih), sehingga membutuhkan perlakuan berbeda.
- Untuk “*coins2.jpeg*”, “*coins3.jpeg*”, “*coins4.jpeg*”, koin memiliki variasi bentuk, pencahayaan, dan posisi, sehingga memerlukan kombinasi beberapa teknik morphological.

Berdasarkan karakteristik tersebut, pendefinisian gambar satu-per-satu lebih optimal dibandingkan pendefinisian langsung [1].

Tabel 1. Perbedaan Definisi Langsung dan Satu-per-Satu

Pendefinisian Satu-per-Satu	Pendefinisian Langsung
<pre>img_path = "assets/bubbles.jpeg"</pre>	<pre>images = ["assets/coins2.jpeg", "assets/coins3.jpeg", "assets/coins4.jpeg"]</pre>

3. *Morphological Operation*

3.1. *"bubbles.jpeg"*

a. *Grayscale*:

Mengubah warna gambar (objek & *background* menjadi abu-abu)

b. *HSV (Hue, Saturation, Value)*:

Citra juga diubah dari RGB ke HSV. HSV lebih stabil terhadap pencahayaan karena memisahkan warna (Hue), kejenuhan (Saturation), dan kecerahan (Value). Dengan

`"cv2.inRange(hsv, lower, upper)"`


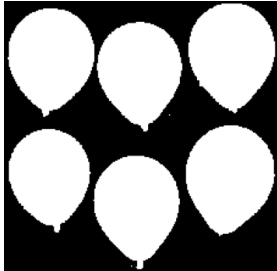
dipilih rentang HSV untuk mendeteksi balon dengan saturasi tinggi (pink, kuning). Hal ini bertujuan agar balon dapat dipisahkan dari background, lalu diproses dalam bentuk citra biner [1, 2].

c. *Morphological*:

- *Structure Element Rectangle*, kernel persegi kecil cukup untuk menutup lubang kecil di bagian tengah balon tanpa merusak bentuk aslinya.
- Menggunakan teknik *Closed*
- Ukuran kernel = 3x3

d. Hasil:

Tabel 2. Hasil *Morphological Operation "bubbles.jpeg"*

Gambar <i>Original</i>	Final
	

3.2. *"coins1.jpeg"*

a. *Grayscale*:

Mengubah warna gambar (objek & *background* menjadi abu-abu)

b. *Binary Inversion*:

Mengubah nilai setiap bit biner dalam pixel menjadi kebalikannya, sebagai contoh jika dalam [1, 2] memiliki nilai 1, maka dengan menggunakan *binary inversion* dapat merubahnya menjadi 0

c. *Otsu*:



Merupakan salah satu teknik merubah warna dari *grayscale* (gambar dengan warna abu-abu) menjadi hitam-putih (*black & white*) [3].

d. *Morphological*:

- *Structure Element Diamond*, Bentuk diamond dipilih karena mendekati bentuk lingkaran. Kernel diamond menjaga keutuhan bentuk bulat koin saat erosi, sekaligus membantu memisahkan koin yang saling berhimpitan.
- Menggunakan teknik *Erosion* dengan kernel (4,4)

e. Hasil:


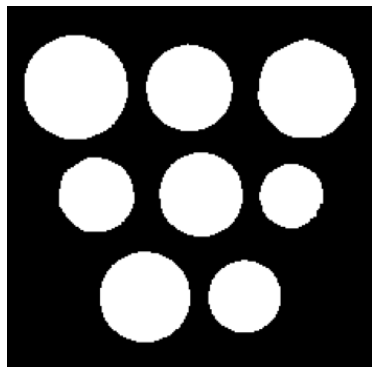
Tabel 3. Hasil *Morphological Operation* “coins1.jpeg”

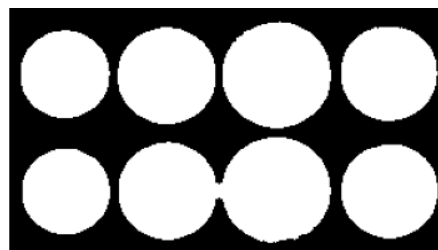
Gambar <i>Original</i>	Final
	

3.3. “coins2.jpeg”, “coins3.jpeg”, & “coins4.jpeg”

- Grayscale*:
Mengubah warna gambar (objek & *background* menjadi abu-abu)
- Binary Inversion*:
Mengubah nilai setiap bit biner dalam pixel menjadi kebalikannya, sebagai contoh jika dalam [1, 2] memiliki nilai 1, maka dengan menggunakan *binary inversion* dapat merubahnya menjadi 0
- Otsu:
Merupakan salah satu teknik merubah warna dari *grayscale* (gambar dengan warna abu-abu) menjadi hitam-putih (*black & white*) [3].
- Morphological:
 - Menggunakan teknik *Erosion*, menghilangkan noise kecil dan memisahkan objek yang saling menempel.
 - Menggunakan teknik *Dilation*, memperbesar area koin setelah erosi agar bentuk kembali utuh.
 - Menggunakan teknik *Closed*, menutup celah kecil di permukaan koin agar terdeteksi sebagai objek penuh.
 - Ukuran kernel = 3x3.
- Hasil:

Tabel 4. Hasil *Morphological Operation* “coins2.jpeg, coins3.jpeg, & coins4.jpeg”

Gambar <i>Original</i>	Final
 <p>“coins2.jpeg”</p>	



"coins3.jpeg"



"coins4.jpeg"

4. References

- [1] OpenCV: Morphological Transformations,
https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html (accessed 21 September 2025).
- [2] Islami F. Implementation of HSV- based Thresholding Method for Iris Detection. *CNAHPC* 2021; 3: 98–104.
- [3] Metode Otsu. *Wikipedia bahasa Indonesia, ensiklopedia bebas*,
https://id.wikipedia.org/w/index.php?title=Metode_Otsu&oldid=26959593 (2025, accessed 21 September 2025).