e- ISSN: 2828–299X

Penggunaan OpenCV untuk Segmentasi Cabai menggunakan Konversi RGB ke HSV

Gilang Dwi Cahyo[1], Nur Ahmadi Thobroni[2], Triasyuko Pinaring Gusti[3], Resty Wulanningrum [4]
[1,2,3,4]Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri
E-mail: *[1]*evermillion921@gmail.com*, [2]*skjthomas11@gmail.com*, [3]*ivonic12@gmail.com,* [4]*restyw@unpkdr.ac.id*

Abstrak *– Cabai merupakan komoditas hortikultura bernilai tinggi, di mana tingkat kematangan memengaruhi kualitas dan harga di pasar. Penelitian ini menggunakan OpenCV dengan analisis ruang warna HSV untuk menilai kematangan cabai secara objektif, menggantikan observasi visual tradisional yang subjektif. Metode yang diterapkan meliputi konversi citra RGB ke HSV, segmentasi untuk mengidentifikasi tingkat kematangan, dan analisis mask untuk menghitung hasil segmentasi. Hasilnya, sistem ini mampu mendeteksi tingkat kematangan dengan akurasi tinggi, menunjukkan jumlah piksel merah (matang) 39.938, kuning/oranye (setengah matang) 643, dan hijau (mentah) 6.115. Pendekatan ini menawarkan solusi efisien, akurat, dan konsisten, mendukung modernisasi sektor pertanian dan meningkatkan daya saing produk di pasar.*

Kata Kunci *— Cabai, Komputer Vision, OpenCV, Segmentasi*

1. PENDAHULUAN

Cabai adalah salah satu komoditas hortikultura yang memiliki nilai ekonomi tinggi di berbagai negara, terutama di Indonesia, di mana cabai banyak digunakan dalam berbagai kuliner lokal. Faktor utama yang menentukan kualitas cabai adalah tingkat kematangannya, yang mempengaruhi rasa, tekstur, serta kandungan nutrisinya. Menurut penelitian oleh [1], tingkat kematangan cabai berperan penting dalam menentukan waktu panen yang tepat sehingga dapat meningkatkan kualitas dan nilai pasar cabai.

Metode tradisional untuk menentukan tingkat kematangan cabai umumnya dilakukan secara manual melalui observasi visual, yang bisa menyebabkan ketidakakuratan karena bergantung pada persepsi subjektif tenaga kerja. Hal ini disampaikan dalam studi oleh [2], yang mengungkapkan bahwa penilaian visual memiliki tingkat variabilitas yang tinggi dan cenderung menyebabkan ketidakkonsistenan dalam hasil akhir, sehingga berdampak pada kualitas dan harga cabai di pasar, dan untuk penelitian ini menggunakan OpenCv.

OpenCV adalah pustaka perangkat lunak open-source untuk computer vision dan image processing, menyediakan berbagai metode untuk memudahkan analisis citra. Menurut penelitian oleh [3], OpenCV memungkinkan identifikasi tingkat kematangan cabai secara otomatis dengan menggunakan analisis spektrum warna, seperti model warna HSV (Hue, Saturation, Value), yang mampu mendeteksi perubahan warna pada cabai sebagai indikator kematangan.

Beberapa penelitian terdahulu menunjukkan efektivitas OpenCV dalam analisis tingkat kematangan buah dan sayuran. Studi oleh [4] menggunakan metode analisis warna berbasis OpenCV untuk mendeteksi kematangan tomat, yang dapat menjadi dasar pendekatan serupa pada cabai. Hasil penelitian menunjukkan bahwa algoritma deteksi warna yang dikembangkan mampu mencapai akurasi hingga 92% dalam mengidentifikasi kematangan buah.

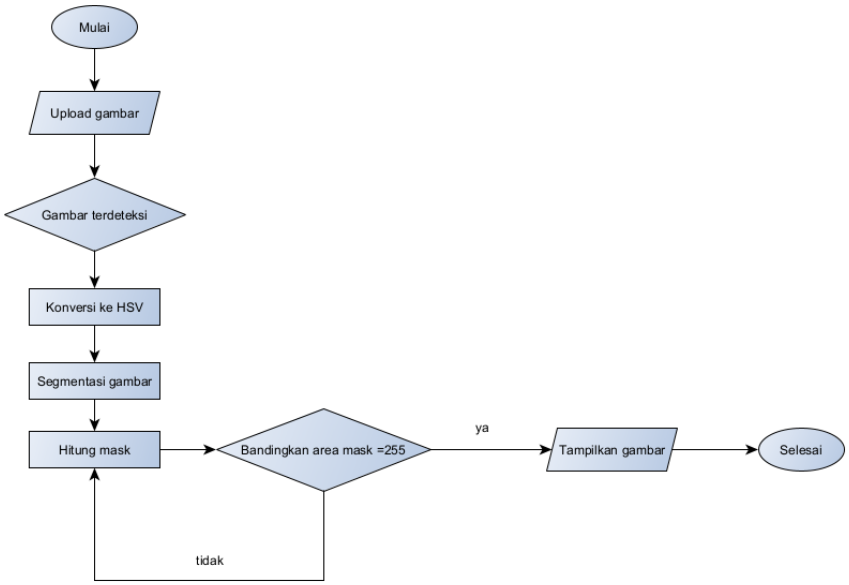
Dari berbagai penelitian yang telah dilakukan, dapat disimpulkan bahwa penggunaan OpenCV dalam mengukur tingkat kematangan cabai adalah pendekatan yang efisien dan akurat. Metode ini tidak hanya mengurangi subjektivitas dalam penilaian visual, tetapi juga meningkatkan efisiensi proses identifikasi kematangan, seperti yang ditegaskan oleh studi [5] yang menggarisbawahi manfaat computer vision dalam pertanian modern untuk memastikan kualitas produk yang lebih baik.

e- ISSN: 2828–299X

2. METODE PENELITIAN

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Seorang analis sistem menggunakan flowchart sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer. Dengan begitu, flowchart dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, flowchart digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. [6] , [7]

Berikut alur diagram Flowchart yang digunakan untuk penelitian ini :



Gambar 1. Diagram Flowchart

2.1 Upload Gambar

Pada penelitian ini gambar yang dipakai yaitu berukuran 500x265 pixel menggunakan citra RGB Dimana RGB sendiri adalah ruang warna aditif, yang berarti bahwa seluruh warna dimulai dengan warna hitam dan dibentuk dengan menambahkan warna Merah (*Red*), Hijau (*Green*), dan Biru (*Blue*).Melalui gabungan warna merah, hijau dan biru akan terbentuk warnawarna baru. Ruang Warna Hue, Saturation, Value (HSV).[8]selanjutnya akan diproses atau dikonversi ke HSV dan dilanjutkan dengan segmentasi gambar dan proses terakhir dihitung jumlah masknya apakah sudah sama dengan 255 atau kurang dari 255.

2.2 Konversi ke HSV

HSV adalah singkatan dari Hue, Saturation, dan Value. Ini adalah ruang warna yang merepresentasikan warna dengan cara yang lebih selaras dengan persepsi manusia dibandingkan dengan ruang warna RGB[9].

- ***Hue*** mengacu pada jenis warna, direpresentasikan sebagai sudut dalam roda warna melingkar (misalnya merah, hijau, biru)
- ***Saturation*** menunjukkan kemurnian atau intensitas warna, dengan nilai berkisar antara 0 (abu-abu) hingga 1 (penuh warna)
- ***Value*** mewakili kecerahan warna, mulai dari hitam (0) hingga putih (1) seiring dengan meningkatnya intensitas.

e- ISSN: 2828–299X

2.3 Segmentasi Gambar

Setelah proses dari konversi ke HSV selesai maka akan dilanjutkan ke Segmentasi Gambar dimana Segmentasi gambar adalah teknik mempartisi gambar menjadi bagianbagian sub penyusun, sangat homogen alam fitur dan proses ini mengakui untuk mengekstrak beberapa informasi yang berguna[10] , [11]. Dan menurut penelitian lain segmentasi merupakan proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisis nilai warna dari tiap piksel pada citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan.[12]

2.4 Hitung Mask

Setelah proses segmentasi selesai Langkah terakhir yang digunakan pada penelitian ini adalah menghitung nilai mask. Mask sendiri adalah representasi tingkat piksel yang digunakan dalam segmentasi misalnya untuk menggambarkan batas-batas objek dalam suatu gambar. Setiap topeng berhubungan dengan contoh objek tertentu, yang menunjukkan piksel mana yang termasuk dalam contoh tersebut. Hal ini memungkinkan pelokalan dan kategorisasi objek secara tepat, memungkinkan aplikasi di berbagai bidang seperti mengemudi otonom, robotika, dan pengeditan gambar. [13]

Dalam pemrosesan gambar, "mask" mengacu pada gambar biner yang digunakan untuk memodifikasi bagian gambar lain secara selektif. Pada dasarnya, ini adalah lapisan yang menentukan area gambar mana yang harus diubah atau dipertahankan. Mask biasanya terdiri dari piksel hitam dan putih, di mana hitam mewakili wilayah yang akan dikecualikan (ditutupi) dan putih menunjukkan wilayah yang akan diproses atau disorot.[14]

Misalnya, mask sering digunakan dalam tugas segmentasi untuk mengisolasi bagian tertentu dari suatu gambar untuk analisis atau manipulasi lebih lanjut. Dalam konteks ini, mask gambar dapat membantu memfokuskan pemrosesan pada wilayah tertentu, seperti mengidentifikasi suatu objek di area tertentu sambil mengabaikan yang lain. Mask juga berguna dalam aplikasi seperti pengeditan gambar, yang dapat membantu menerapkan efek hanya pada bagian tertentu dari gambar, seperti mengaburkan latar belakang atau meningkatkan warna pada area yang dipilih[15].

2.5 Tampilkan Gambar

Tampilkan gambar disini ditujukan pada output dari cabai yang sudah disegmentasi.

3. HASIL DAN PEMBAHASAN

Bedasarkan penelitian tersebut diperoleh hasil sebagai berikut :

3.1 Upload Gambar

Langkah pertama yang dilakukan yaitu upload gambar Dimana user harus menggunggah gambar cabai baik itu dari foto langsung maupun dari internet, berikut adalah kode dan hasil untuk proses upload gambar :

```
# Path gambar
image_path = './merah.jpg'
```

Gambar 2. Gambar Kode untuk upload gambar

```
image = cv2.imread(image_path)
if image is None:
    print("Gambar tidak ditemukan. Pastikan jalur file benar.")
    return
```

Gambar 3. Gambar Kode proses pembacaan gambar

e- ISSN: 2828–299X

Gambar 4. Gambar RGB Cabai

Berikut adalah penjelasan dari kode diatas :

- *image_path*: Merupakan variabel string yang menyimpan path file gambar di sistem.
- *cv2.imread(image_path)*: Membaca gambar dari path yang diberikan dan mengembalikan objek array numpy yang merepresentasikan gambar.
- Jika file tidak ditemukan atau jalurnya salah, *cv2.imread* akan mengembalikan None.

3.2 Konversi Ke HSV

Setelah user mengunggah gambar proses selanjutnya akan di konversi ke HSV dimana proses perubahan representasi warna dari model berbasis cahaya ke model berbasis persepsi manusia, berikut adalah hasil gambar yang telah dikonversi ke HSV, Berikut kode untuk proses konversi dari RGB ke HSV.

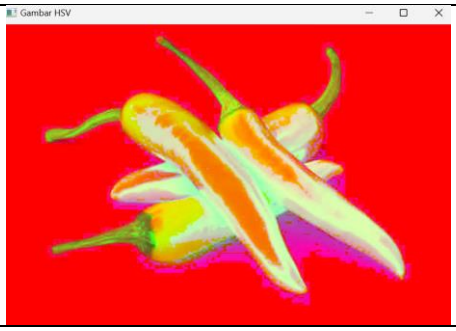
```
# Konversi gambar ke HSV
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

Gambar 5. Gambar Kode proses konversi RGB ke HSV

Berikut adalah penjelasan dari kode diatas :

- *cv2.cvtColor*: Fungsi ini digunakan untuk mengonversi gambar dari satu ruang warna ke ruang warna lainnya.
- *image*: Gambar asli yang dibaca dalam format BGR (karena cv2.imread secara default membaca gambar dalam format BGR).
- *cv2.COLOR_BGR2HSV*: Parameter ini menentukan bahwa konversi dilakukan dari ruang warna BGR ke HSV.

Tabel 1. Perbandingan gambar RGB dan HSV

RGB	HSV
	

Berikut adalah nilai dari masing masing gambar yang ada ditabel diatas :

Tabel 2. Perbandingan Nilai matrix RGB dan HSV

RGB			HSV		
240	220	250	180	150	200
220	106	208	160	120	190
240	198	219	170	140	210

3.3 Segmentasi Gambar

Selanjutnya gambar akan melewati proses segmentasi adalah proses membagi gambar menjadi bagian-bagian yang lebih kecil atau segmen yang lebih bermakna. Berikut adalah kode proses segmentasi dan hasil dari segmentasi gambar

a) Proses Segmentasi

Proses segmentasi dilakukan dengan memisahkan piksel berdasarkan rentang warna di ruang warna HSV. Bagian berikut adalah inti dari proses segmentasi:

```
red_mask = cv2.inRange(hsv, lower_red, upper_red)
yellow_mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
green_mask = cv2.inRange(hsv, lower_green, upper_green)
```

Gambar 5. Gambar Kode untuk membuat Mask untuk Warna Tertentu

Dimana *cv2.inRange* berfungsi untuk menghasilkan mask biner (nilai 0 dan 255) untuk piksel yang berada di dalam rentang warna tertentu (merah, kuning/oranye, atau hijau).

b) Hasil Segmentasi

Hasil segmentasi adalah gambar hasil masking yang menunjukkan area dengan warna dominan berdasarkan status kematangan. Bagian kode yang menghasilkan dan menampilkan hasil segmentasi adalah:

```
result = cv2.bitwise_and(image, image, mask=dominant_color_mask)
```

Gambar 6. Gambar Kode untuk Membuat Gambar dengan Mask Warna Dominan

Dimana *cv2.bitwise_and* fungsi ini mengaplikasikan mask ke gambar asli, sehingga hanya area dengan warna dominan yang ditampilkan.

```
cv2.imshow("Output", result)
```

Gambar 7. Gambar Kode untuk Menampilkan Hasil Segmentasi

Dimana Gambar yang ditampilkan adalah hasil dari result, yang menunjukkan hanya area cabai dengan warna dominan berdasarkan status kematangan.

Gambar 8. Gambar Segmentasi Cabai

Berikut Nilai matrix dari gambar yang sudah disegmentasi

Tabel 3. Nilai matrix gambar Segmentasi

37	37	2
47	59	2
16	17	1

e- ISSN: 2828–299X

3.4 Hitung Mask

Setelah proses segmentasi selesai akan dilanjutkan kedalam proses menghitung jumlah nilai mask, Dimana menggunakan rumus sebagai berikut ini :

$$I'^{(x,y)^n} = \sum_{i=-m}^{m} \sum_{j=-n}^{n} I\,(x+i, y+j) \cdot M(i.j)$$

Dimana :

- $I(x,y)$ adalah nilai pixel dari posisi (x,y)
- $M(i.j)$ adalah nilai mask pada posisi $(i.j)$
- $I'(x,y)$ adalah nilai pixel hasil Konversi

Berikut adalah gambar dari nilai masing-masing mask :

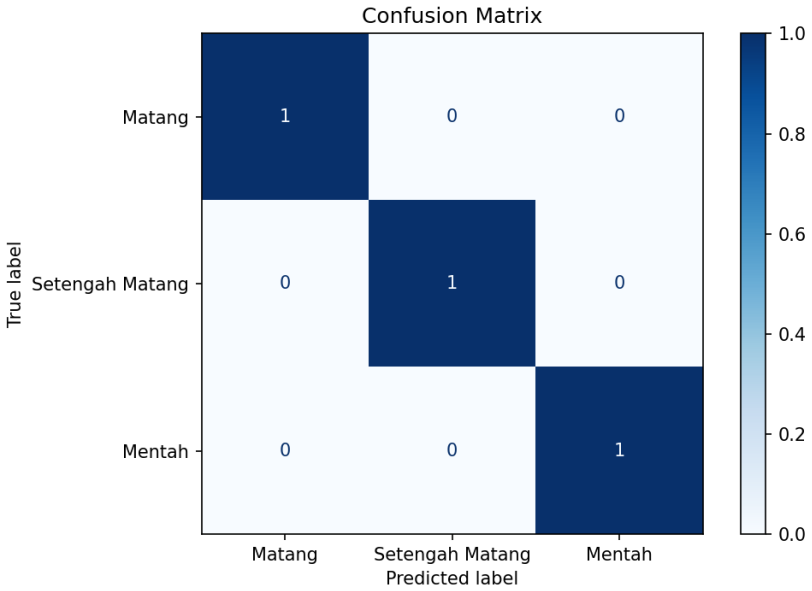
```
Jumlah piksel merah (matang): 39938
Jumlah piksel kuning/orange (setengah matang): 643
Jumlah piksel hijau (mentah): 6115
```

Gambar 9. Gambar nilai Mask

3.5 Tampilkan Gambar

Gambar 10. Gambar Output Cabai

Confusion matrix merupakan alat evaluasi yang sering digunakan untuk menilai performa model klasifikasi. Matriks ini menggambarkan jumlah prediksi benar dan salah yang dibuat oleh model untuk setiap kelas, dibandingkan dengan label sebenarnya pada dataset pengujian. Confusion matrix sangat membantu dalam mengidentifikasi seberapa baik model mengenali setiap kelas, serta memberikan wawasan tentang jenis kesalahan prediksi yang terjadi.



Gambar 11. Gambar Confusion Matrix

Berdasarkan confusion matrix di atas, performa model dalam mengklasifikasikan data menunjukkan hasil yang sempurna. Pada kelas "Matang," sebanyak 1 sampel yang sebenarnya termasuk dalam kelas ini berhasil diprediksi dengan benar tanpa adanya kesalahan prediksi ke kelas lain, yaitu "Setengah Matang" atau "Mentah." Hal serupa juga terjadi pada kelas "Setengah Matang," di mana 1 sampel diprediksi dengan benar sebagai "Setengah Matang" tanpa ada kesalahan klasifikasi ke kelas lainnya. Begitu pula dengan kelas "Mentah," 1 sampel yang sebenarnya termasuk dalam kelas ini diprediksi dengan benar tanpa salah diklasifikasikan sebagai "Matang" atau "Setengah Matang." Secara keseluruhan, model memiliki akurasi sempurna, ditunjukkan oleh nilai diagonal utama confusion matrix yang semuanya bernilai 1, sementara seluruh nilai di luar diagonal utama bernilai 0. Hal ini menunjukkan bahwa model mampu mengenali setiap kelas dengan benar tanpa kesalahan pada dataset pengujian.

4. SIMPULAN

Penelitian ini membuktikan bahwa penggunaan OpenCV dengan ruang warna HSV sangat efektif untuk segmentasi dan identifikasi tingkat kematangan cabai. Proses konversi RGB ke HSV, segmentasi gambar, dan perhitungan nilai mask memberikan hasil yang lebih objektif dibandingkan metode manual. Nilai mask piksel yang diperoleh adalah merah (matang) sebanyak 39.938 piksel, kuning/oranye (setengah matang) sebanyak 643 piksel, dan hijau (mentah) sebanyak 6.115 piksel.

Proses segmentasi menggunakan fungsi cv2.inRange menghasilkan mask biner yang memisahkan piksel berdasarkan rentang warna tertentu dengan presisi tinggi. Nilai matrix RGB dan HSV dari hasil penelitian menunjukkan perbandingan warna dominan yang memberikan gambaran tingkat kematangan cabai secara jelas. Teknik segmentasi berbasis warna ini terbukti mampu mengidentifikasi warna dominan sesuai tingkat kematangan..

5. SARAN

Penelitian lanjutan dapat mengoptimalkan pendekatan dengan memperluas rentang nilai HSV untuk mengurangi noise. Selain itu, integrasi teknik operasi morfologi seperti dilasi atau erosi diharapkan mampu meningkatkan kualitas hasil segmentasi. Pengembangan lebih lanjut ke aplikasi berbasis mobile juga dapat mempermudah penerapan teknologi ini untuk mendukung sektor pertanian modern.

DAFTAR PUSTAKA

[1] D. N. Edowai, S. Kairupan, and D. H. Rawung, “MUTU CABAI RAWIT (CAPSICUM FRUTESCENS L) PADA TINGKAT KEMATANGAN DAN SUHU YANG BERBEDA SELAMA PENYIMPANAN.”

[2] Y. A. Pratama, “Membangun Sistem Identifikasi Kematangan Buah Alpukat menggunakan teknologi Pengolahan Citra Digital,” *Kalijaga : Jurnal Penelitian Multidisiplin Mahasiswa*, vol. 1, no. 3, pp. 102–108, Jul. 2024, doi: 10.62523/kalijaga.v1i3.18.

[3] Susanto, R., et al. (2022). "Penggunaan OpenCV untuk Deteksi Warna dalam Penentuan Kematangan Buah." Jurnal Informatika dan Sains Komputer.

[4] Rahmawati, S., et al. (2021). "Deteksi Kematangan Tomat Menggunakan OpenCV." Jurnal Teknologi Pangan dan Pertanian.

[5] Ardiansyah, D., et al. (2023). "Pemanfaatan Computer Vision dalam Pertanian Modern." Jurnal Teknologi Pertanian Terapan.

[6] R. Rosaly, A. Prasetyo, and M. Kom, "Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan."

[7] A. Zalukhu *et al.*, "PERANGKAT LUNAK APLIKASI PEMBELAJARAN FLOWCHART," *Jurnal Teknologi Informasi dan Industri*, vol. 4, no. 1, 2023.

[8] H. Sanusi, S. H. S., and D. T. Susetianingtias, "PEMBUATAN APLIKASI KLASIFIKASI CITRA DAUN MENGGUNAKAN RUANG WARNA RGB DAN HSV," *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 3, pp. 180–190, 2019, doi: 10.35760/ik.2019.v24i3.2323.

[9] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," in *IEEE International Conference on Image Processing*, 2002. doi: 10.1109/icip.2002.1040019.

[10] K. K. D. Ramesh, G. Kiran Kumar, K. Swapna, D. Datta, and S. Suman Rajest, "A review of medical image segmentation algorithms," *EAI Endorsed Trans Pervasive Health Technol*, vol. 7, no. 27, 2021, doi: 10.4108/eai.12-4-2021.169184.

[11] T. Sulistyorini, E. Sova, N. Sofie, and R. I. Napitupulu, "PENERAPAN HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK (CNN) DALAM MEMBANGUN MODEL SEGMENTASI GAMBAR MENGGUNAKAN ARSITEKTUR U-NET DENGAN TENSORFLOW," *Jurnal Ilmiah Informatika Komputer*, vol. 28, no. 2, pp. 112–121, 2023, doi: 10.35760/ik.2023.v28i2.6959.

[12] I. S. Areni, I. Amirullah, and N. Arifin, "Klasifikasi Kematangan Stroberi Berbasis Segmentasi Warna dengan Metode HSV," *Jurnal Penelitian Enjiniring*, vol. 23, no. 2, pp. 113–116, Nov. 2019, doi: 10.25042/jpe.112019.03.

[13] T. Cheng, X. Wang, L. Huang, and W. Liu, "Boundary-preserving Mask R-CNN," Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.08921

[14] B. Öztürk and M. Özkar, "CREATING AND USING MASK IMAGES FOR SEGMENTATION IN POINT CLOUD DATA," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, International Society for Photogrammetry and Remote Sensing, May 2022, pp. 1133–1138. doi: 10.5194/isprs-archives-XLIII-B2-2022-1133-2022.

[15] K. Lis, M. Koryciński, and K. A. Ciecierski, "Classification of masked image data," *PLoS One*, vol. 16, no. 7 July, Jul. 2021, doi: 10.1371/journal.pone.0254181.