

## Foreground Extraction pada Citra Daun Melon dengan Bantuan Deep Neural Network

Muhammad Fajar Estu Nugroho<sup>1</sup>, Nurlana Sanjaya<sup>2</sup>, Ayu Shafira Tubagus<sup>3</sup>,  
M Rayhan Rizqullah Syarif<sup>4</sup>, Chaerur Rozikin<sup>5</sup>

<sup>1,2,3,4,5</sup>Teknik Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

<sup>1</sup>fajar.estu17147@student.unsika.ac.id, <sup>2</sup>nurlana.sanjaya17167@student.unsika.ac.id,

<sup>3</sup>ayu.shafira17064@student.unsika.ac.id, <sup>4</sup>m.rayhan17012@student.unsika.ac.id,

<sup>5</sup>chaerur.rozikin@staff.unsika.ac.id

---

### Abstrak

Ekstraksi fitur merupakan tahapan dalam melakukan proses pengolahan citra. Ada banyak metode ekstraksi fitur yang dilakukan untuk proses pengolahan citra, salah satunya adalah ekstraksi *foreground*. Ekstraksi *foreground* yang merupakan segmentasi pada gambar diperlukan untuk memisahkan obyek utama pada citra yang akan diproses dalam pengolahan citra diperlukan untuk menyeleksi obyek utama dengan *background*-nya sehingga proses pengolahan citra bisa berfokus pada obyek utama. Ada beberapa algoritma yang bisa diterapkan untuk melakukan ekstraksi *foreground*, salah satu yang paling populer adalah algoritma *Grabcut*. Di dalam jurnal ini, kami mencoba melakukan ekstraksi *foreground* pada obyek daun melon sebagai obyek utama dengan harapan hasil dari ekstraksi *foreground* dapat lebih lanjut dimanfaatkan terutama dalam proses pengembangan aplikasi yang berhubungan dengan daun melon, seperti misalnya pendeteksian dini terhadap penyakit daun melon berbasis citra. Dalam jurnal ini ekstraksi *foreground* dilakukan dengan bantuan algoritma *Grabcut* dengan bantuan *deep neural network* yang didasarkan pada algoritma *holistically-nested edge detection* dan diaplikasikan sekaligus pada data obyek daun melon yang banyak. Algoritma yang diusulkan bertujuan untuk meningkatkan akurasi dari *grabcut* dan tetap menjaga *user input* dengan seminimal mungkin. Dengan melakukan pengujian berdasarkan pengujian kasus yang dibuat, hasilnya pada pengujian sebanyak 351 citra daun melon, terdapat 68% citra yang dapat diekstraksi citra daunnya dengan sempurna.

**Kata kunci** : Deep Neural Network, Grabcut, Ekstraksi Foreground

---

### 1. Pendahuluan

Melon (*Cucumis melo* L) merupakan tanaman hortikultura yang biasa ditanam di dalam rumah kaca merupakan tanaman penting di seluruh dunia (Huang et al., 2017). Dalam proses penanamannya, tanaman melon tidak terlepas dari penyakit baik dari virus maupun dari bakteri. Salah satu bagian tanaman yang dapat terkena penyakit adalah daun. Dalam (Pineda et al., 2018) deteksi awal dari penyakit daun pada melon dapat menggunakan deteksi menggunakan data gambar. Tentunya data gambar yang digunakan harus diolah dahulu dengan pengolahan citra digital sehingga pemrosesan dapat berfokus hanya pada daun.

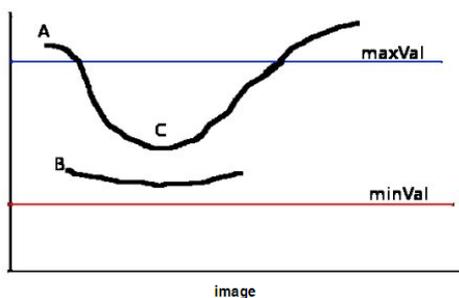
Pengolahan citra digital memiliki beberapa tujuan, salah satunya untuk mengambil ciri dari citra sehingga dapat dikenali yang biasa disebut dengan ekstraksi ciri (Riantama et al., 2019). Ekstraksi fitur (*feature extraction*) adalah suatu metode yang digunakan untuk mengambil ciri objek pada suatu citra dengan tujuan mengenali objek tersebut (Suryawibawa et al., 2015). Salah satu tahapan dalam

ekstraksi fitur adalah akuisisi yaitu sebuah proses mendapatkan data citra dari analog ke bentuk digital (Wirdiani et al., 2018). Pemisahan *background* dari *foreground* menjadi hal penting dalam pemrosesan aplikasi berbasis *computer vision* (V et al., 2020).

Pengambilan fitur *foreground* dapat dilakukan dengan pendekatan algoritma *GraphCut*. Sebuah algoritma yang secara efisien memisahkan *background* dengan *foreground* (Yi & Moon, 2012). Namun algoritma *Grabcut* lalu muncul sebagai pengganti algoritma *GraphCut* dengan kelebihan – kelebihan yang disediakan oleh *Grabcut* (Boykov & Jolly, 2001). Sehingga untuk pendekatan selanjutnya adalah dengan metode *Grabcut*, sebuah algoritma *foreground extraction* dimana pengguna memberikan input berupa citra disertai dengan daerah *foreground* yang dipilih menggunakan memberikan *bounding box* yang menunjukkan area lokasi pasti *foreground* atau dengan *mask* yang menunjukkan *foreground* dan *background* dan kemudian algoritma akan memisahkan *foreground* dari *background* (Rother et al., 2004). *Grabcut* merupakan algoritma populer untuk melakukan segmentasi pada gambar dengan

mengurangi kebutuhan input dari pengguna (Li et al., 2018). Namun algoritma ini mempunyai kecepatan yang relatif lambat dan memiliki keterbatasan ketika objek *background* terlalu kompleks (Li et al., 2018). Dalam (Riantama et al., 2019) menggunakan 40 buah citra dan dilakukan pengujian sebanyak 3 kali iterasi untuk mendapatkan waktu rata-rata proses segmentasi citra. Citra hasil segmentasi tidak selalu bersih dari *noise* atau bagian dari *background* karena piksel warna berada di dalam daerah *bounding box* dianggap sebagai bagian *foreground*. *Grabcut* juga memiliki keterbatasan untuk citra dengan kontras yang rendah (Aykut & Akturk, 2018).

Pendekatan selanjutnya adalah dengan algoritma *canny edge* dengan menganalisis tepi dari objek (Canny, 1986) untuk mendapatkan bagian *foreground* dari citra. Algoritma *canny edge* pada dasarnya menemukan titik tepi pada gambar *grayscale* dengan perubahan nilai intensitas gradien yang paling besar, daerah ini ditemukan dengan menentukan gradien pada gambar (Kaehler & Gary, 2008). Tepi adalah perubahan nilai intensitas derajat keabuan yang besar dalam jarak yang singkat. Deteksi tepi juga merupakan algoritma fundamental dalam pengolahan citra, *machine vision*, dan *computer vision* (Umbaugh, 2010). Namun algoritma *canny edge* memiliki beberapa kelemahan diantaranya adalah tidak mempunyai memisahkan *background* dan target deteksi tepi (Xuan & Hong, 2018) dalam beberapa kasus dimana setiap objek memiliki kontras yang hampir serupa dan *hysteresis thresholding* yang diperlukan dalam algoritma ini membutuhkan proses manual yang membutuhkan percobaan ulang dan membutuhkan validasi secara visual. Masalah selanjutnya dengan *hysteresis thresholding* yang membutuhkan dua variabel *threshold*, yaitu *minimum value* dan *maximum value* dimana jika intensitas gradien lebih dari *maximum value* maka ditentukan menjadi sebuah tepi dan intensitas gradien dibawah *minimum value* ditentukan sebagai bukan tepi (Kaehler & Gary, 2008), adalah setiap gambar membutuhkan *threshold* yang berbeda – beda sehingga perbedaan pencahayaan juga membutuhkan *threshold* yang berbeda juga (Xie & Tu, 2015). *Hysteresis Thresholding* dijelaskan melalui gambar 1.



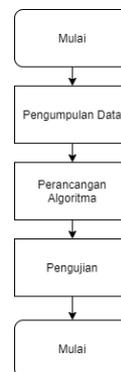
Gambar 1 Hysteresis Thresholding (OpenCV, n.d.)

Dalam (Xie & Tu, 2015) peneliti mencoba memberikan inovasi deteksi tepi dengan *deep neural*

*network* untuk menghasilkan deteksi tepi yang lebih baik. Algoritma tersebut disebut *Holistically-Nested Edge Detection* (HED) yang bisa menjadi pendekatan untuk melakukan deteksi *foreground* pada jurnal ini. Tujuan mendeteksi tepi sendiri adalah untuk melakukan segmentasi pada bagian – bagian pada citra. Dengan bantuan adanya *deep neural network*, yaitu algoritma pemodelan tingkat tinggi pada data dengan fungsi transformasi non-linear yang tertata dengan berlapis – lapis (Deng & Yu, 2013) yang diterapkan pada deteksi tepi, penelitian ini berfokus pada proses ekstraksi *foreground* dan diharapkan algoritma deteksi tepi menggunakan bantuan HED dapat lebih baik dalam melakukan deteksi tepi dan dapat memisahkan antara *foreground* dengan *background* dengan lebih baik dibanding jika hanya menggunakan algoritma *Grabcut*. Citra daun dipilih sehingga penelitian kedepannya dapat menggunakan algoritma dalam jurnal ini untuk mengekstraksi fitur – fitur pada daun dan menggunakannya seperti untuk deteksi penyakit pada daun berbasis *computer vision*.

## 2. Metode Penelitian

Tahapan metode penelitian yang dilakukan sama dengan tahapan pengolahan citra digital pada umumnya. Metodologi penelitian dimulai dari pengumpulan data, perancangan algoritma dan pengujian.



Gambar 2 Metodologi Penelitian

### 2.1 Pengumpulan Data

Data yang digunakan adalah data berupa gambar citra daun melon yang diambil dari kebun – kebun melon yang berada di Purwakarta dan Karawang. Data gambar berasal dari tanaman – tanaman dengan kondisi yang bervariasi mulai dari yang sehat sampai yang sakit dengan posisi pengambilan gambar yang berbeda – beda. Gambar diambil sendiri oleh peneliti pada tahun 2019 menggunakan kamera *handphone*.

### 2.2 Perancangan Algoritma

Untuk melakukan ekstraksi *foreground* dengan *deep neural network*, dibutuhkan bahasa pemrograman yang lengkap dari segi fitur dan pustakanya. Dengan pustaka OpenCV, Python

menjadi bahasa yang dipilih untuk melakukan uji coba algoritma. Sebelum melakukan penulisan program, algoritma terlebih dahulu digambarkan menggunakan *flowchart* diagram untuk kemudahan penulisan program.

### 2.2.1 Algoritma Program Keseluruhan

Program secara keseluruhan akan menggabungkan antara algoritma *grabcut* dengan algoritma HED. Gambar 2, menunjukkan proses kerja dari program yang diusulkan secara keseluruhan.



Gambar 3 *Flowchart* Diagram untuk Algoritma Ekstraksi *Foreground*

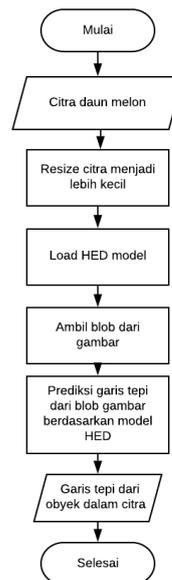
Proses dari program secara keseluruhan adalah sebagai berikut :

1. Muat gambar yang akan diproses
2. Lakukan deteksi tepi sederhana dengan *canny edge* untuk melakukan perbandingan kesuksesan ekstraksi *foreground* dengan HED
3. Hasil dari deteksi tepi akan diperbaiki menggunakan *holistically-nested edge detection* (Xie & Tu, 2015).
4. Hasil dari perbaikan tadi akan dijadikan sebagai *mask* untuk digunakan ke dalam *grabcut*.

Program akan mengeluarkan *output* berupa gambar yang telah terseleksi *foreground*-nya.

### 2.2.2 HED Untuk Deteksi Tepi

Sebelum memasuki tahap *Grabcut*, *mask* dari gambar dibutuhkan untuk menentukan batas – batas antara *foreground* dan *background*. *Mask* dihasilkan dengan menentukan garis tepi dari citra menggunakan HED. Gambar 3, Menunjukkan *flowchart* pengaplikasian HED terhadap citra.



Gambar 4 *Flowchart* deteksi tepi dengan HED

Proses deteksi tepi dengan model HED dijelaskan sebagai berikut :

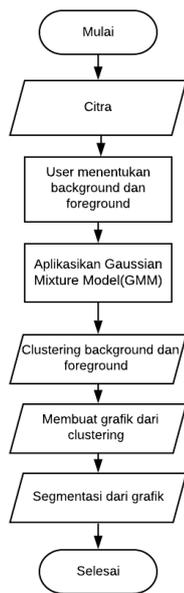
1. Masukkan citra daun melon.
2. Lakukan *resize* terhadap citra untuk menghemat memori.
3. Muat HED model. HED model adalah model *pre-trained* yaitu model pendeteksian garis tepi yang dibuat menggunakan *deep neural network*. Model ini digunakan untuk melakukan prediksi dan deteksi terhadap garis tepi objek – objek pada gambar.
4. Ambil *blob image*.
5. Melakukan prediksi terhadap garis tepi berdasarkan model HED yang sudah dilatih sebelumnya.
6. *Output* berupa gambar garis tepi yang nantinya akan menjadi *mask* untuk melakukan *Grabcut*.

### 2.2.3 Algoritma Grabcut

Algoritma *Grabcut* digunakan untuk mengekstraksi *foreground* dari *background*.

Berikut ini adalah penjelasan dari algoritma *grabcut* dari gambar 5 :

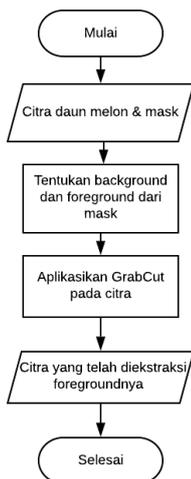
1. Masukkan citra yang akan diolah
2. *User* menentukan *foreground* dan *background*. Penentuan bisa menggunakan masukkan zona *foreground* dalam bentuk koordinat atau *mask*.
3. *Gaussian Mixture Model* (GMM) akan memodelkan *foreground* dan *background*.
4. GMM akan melakukan *clustering* terhadap piksel – piksel untuk menentukan *foreground* dan *background*.
5. Dari distribusi piksel, dibuat grafik.
6. Dengan Algoritma *mincut*, grafik disegmentasi untuk memisahkan antara piksel dari *foreground* dengan *background*.



Gambar 5 Algoritma Grabcut

### 2.2.4 Penerapan Algoritma Grabcut

Algoritma *Grabcut* membutuhkan variabel utama citra dan *mask* untuk memisahkan *foreground* dan *background*. *Mask* merupakan citra abu – abu dengan derajat keabuan berbeda – beda di setiap daerahnya untuk menentukan mana *foreground* dan *background*. *Mask* didapatkan dari algoritma HED. Gambar 5, Menunjukkan *flowchart* dari penerapan algoritma *Grabcut* dengan *mask* dari HED.



Gambar 6 Penerapan Algoritma Grabcut dengan Mask

Penjelasan dari *flowchart* dari gambar 5 adalah sebagai berikut :

1. Masukkan citra daun melon dan *mask* hasil HED
2. Tentukan bagian *foreground* dan *background* dari *mask*. Dalam tulisan ini digunakan *probable foreground* dan *probable*

*background* untuk melakukan pemisahan [17], penentuan *probable foreground* menggunakan daerah *mask* yang mempunyai derajat keabuan kurang dari sama dengan 50 dan daerah *mask* dengan derajat keabuan lebih dari sama dengan 205 sebagai *background*.

3. Aplikasikan *Grabcut* pada citra.
4. *Output* berupa citra yang telah dilakukan *Grabcut*.

### 2.3 Pengujian

Pengujian menggunakan uji berupa *test case* untuk menguji kemampuan program dalam mengekstraksi *foreground* dari *background*.

## 3. Hasil dan Pembahasan

### 3.1 Perisapan Data

Data uji coba menggunakan data citra daun melon yang diambil dari kebun – kebun yang berlokasi di Purwakarta dan Karawang. Data yang digunakan terdapat 351 data gambar yang terdiri dari berbagai macam bentuk dan kondisi daun melon mulai dari yang sakit hingga yang sehat, dari yang berwarna hijau hingga kekuningan karena penyakit. Semua data gambar dimasukkan ke dalam sebuah *folder* dan nantinya akan dilakukan proses ekstraksi *foreground* secara *looping* untuk setiap gambar di dalam *folder*.

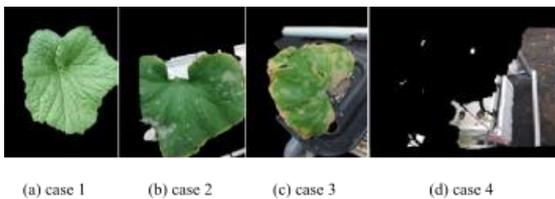
### 3.2 Detail Pelaksanaan

Sebuah *test case* dibuat untuk menentukan hasil dari uji coba. *Test case* tersebut terdiri dari 4 kasus sebagai berikut :

1. Memotong Sempurna  
Algoritma berhasil melakukan pemotongan dengan sempurna dan berhasil mendeteksi tepian dari daun melon dengan rasio luas daun yang terpotong dengan luas daun pada gambar aslinya antara 80% - 100% dan tidak menyisakan obyek *background*.
2. Hampir Memotong Sempurna  
Algoritma berhasil mendeteksi dan melakukan pemotongan pada obyek daun, namun ada beberapa obyek lain yang ikut terdeteksi sebagai daun dengan rasio luas daun yang terpotong dengan luas daun pada gambar aslinya antara 60% - 79% atau menyisakan obyek *background* dengan rasio kurang dari atau sama dengan 20%.
3. Banyak Salah Memotong  
Algoritma mendeteksi dan memotong obyek daun namun ada banyak obyek lain yang terdeteksi sebagai daun dengan rasio luas daun yang terpotong dengan luas daun pada gambar aslinya antara 30% - 59% atau menyisakan obyek pada *background* dengan rasio kurang dari atau sama dengan 50%.
4. Salah Memotong  
Algoritma gagal mendeteksi dan memotong obyek daun dengan rasio *foreground* yang

terekstraksi kurang dari 30% atau obyek *background* yang terbawa lebih dari 50%.

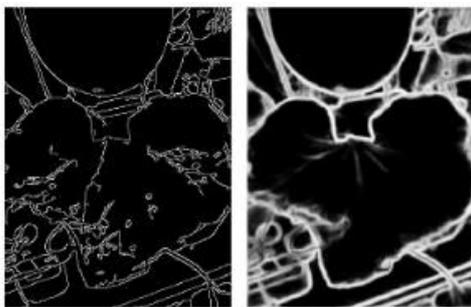
Gambar 6 menunjukkan sampel dari *test case* yang menunjukkan contoh dari kasus – kasus yang ditemukan dalam pengujian dari kiri ke kanan; memotong sempurna, hampir memotong sempurna, banyak salah memotong, salah memotong.



Gambar 7 Contoh Kasus Hasil Pengujian

### 3.3 Hasil Pengujian

Algoritma *Holistically-Nested Edge Detection* memiliki hasil deteksi tepi lebih baik dai algoritma *canny edge* karena dapat dengan jelas memperlihatkan batas – batas dari setiap obyek dengan baik. Gambar 7 menunjukkan perbandingan antara hasil deteksi tepi dengan *canny edge* dengan algoritma HED. Dari kiri ke kanan ; Hasil deteksi tepi dengan *canny edge*, hasil deteksi tepi dengan HED.



Gambar 8 Hasil Deteksi Tepi

Hasil dari deteksi tepi dengan HED selanjutnya dijadikan sebuah *mask* untuk melakukan operasi algoritma *Grabcut*. Pada penelitian ini kami memiliki 351 Citra Daun Melon yang sehat dan memiliki penyakit. Dengan algoritma *Grabcut* yang menggunakan *mask* dari hasil deteksi tepi, menunjukkan adanya peningkatan dari hasil ekstraksi *foreground*. Gambar 8 menunjukkan perbandingan antara gambar awal, hasil pemotongan dengan *Grabcut*, dan hasil pemotongan dengan algoritma *Grabcut* dengan bantuan deteksi tepi dengan model HED. Dari kiri ke kanan; Gambar Awal, Hasil *Grabcut* Tanpa Bantuan HED, Hasil *Grabcut* dengan Deteksi Tepi Menggunakan Bantuan HED.



Gambar 9 Sampel Citra Hasil dari Pemotongan

Data – data hasil pengujian dimasukkan ke dalam tabel pada *software* Microsoft Excel. Hasilnya, dengan bantuan deep learning, persentase *case* ke 1 yaitu memotong sempurna mencapai 68% Dengan bantuan *mask* dari deteksi tepi menggunakan HED, algoritma *Grabcut* dapat lebih baik dalam membedakan *foreground* dan *background*. Tabel 1, menunjukkan hasil perbandingan hasil pengujian dengan bantuan deep neural network dan hanya menggunakan *Grabcut*.

Tabel 1 Hasil Pengujian

Dengan Deep Neural Network		Hanya Menggunakan Grabcut	
Case	Persentase	Case	Persentase
1	68%	1	29%
2	0%	2	18%
3	8%	3	44%
4	25%	4	9%

Persentase *case* 1 dimana hasil pemotongan lebih halus dan lebih baik dalam memotong daun, lebih banyak pada pengujian dengan bantuan HED.

### 4. Kesimpulan

Ekstraksi *foreground* pada objek daun melon dengan bantuan algoritma deteksi tepi *Holistically-Nested Edge Detection* telah dilakukan dalam jurnal ini. Dari hasil penelitian yang dilakukan, akurasi dari penggunaan *Grabcut* dengan *Deep Neural Network* HED untuk melakukan pemisahan *background* dan *foreground* pada citra daun melon telah dilakukan dengan tingkat akurasi mencapai 68%. Meskipun tingkat akurasi kesuksesan hanya mencapai 68%, namun algoritma ini bisa digunakan karena lebih baik menyeleksi *foreground* daripada *Grabcut* dan bisa digunakan untuk proses ekstraksi *foreground* untuk data yang banyak karena algoritma ini membutuhkan *user input* minimal. Kesuksesan dari pemisahan antara *background* dan *foreground* dipengaruhi oleh pencahayaan dan kontras antara *foreground* dan *background*. Harapan untuk penelitian selanjutnya dapat meningkatkan kemampuan algoritma untuk mendeteksi objek lain, tidak hanya terbatas pada daun melon dan mendapatkan tingkat ke akurasi yang lebih tinggi dari penelitian saat ini.

**Daftar Pustaka:**

- Aykut, M., & Akturk, S. M. (2018). An Improvement on Grabcut with CLAHE for the Segmentation of the Objects with Ambiguous Boundaries. *Image Analysis and Recognition*, 116–122. [https://doi.org/10.1007/978-3-319-93000-8\\_14](https://doi.org/10.1007/978-3-319-93000-8_14)
- Boykov, Y. Y., & Jolly, M.-P. (2001). Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. *International Conference on Computer Vision, July*, 105–112.
- Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>
- Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4), 197–387. <https://doi.org/10.1561/20000000039>
- Huang, Y., Li, W., Zhao, L., Shen, T., Sun, J., Chen, H., Kong, Q., Nawaz, M. A., & Bie, Z. (2017). Melon fruit sugar and amino acid contents are affected by fruit setting method under protected cultivation. *Scientia Horticulturae*, 214, 288–294. <https://doi.org/10.1016/j.scienta.2016.11.055>
- Kaehler, A., & Gary, B. (2008). Learning OpenCV--Computer Vision with the OpenCV Library. In *IEEE Robotics & Automation Magazine* (Vol. 16, Issue 3). O'Reilly Media, Inc. <https://doi.org/10.1109/mra.2009.933612>
- Li, Y., Zhang, J., Gao, P., Jiang, L., & Chen, M. (2018). Grab Cut Image Segmentation Based on Image Region. *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 311–315. <https://doi.org/10.1109/ICIVC.2018.8492818>
- OpenCV. (n.d.). *Canny Edge Detection*. [https://docs.opencv.org/master/da/d22/tutorial\\_py\\_canny.html](https://docs.opencv.org/master/da/d22/tutorial_py_canny.html)
- Pineda, M., Pérez-Bueno, M. L., & Barón, M. (2018). Detection of bacterial infection in melon plants by classification methods based on imaging data. *Frontiers in Plant Science*, 9(February), 1–10. <https://doi.org/10.3389/fpls.2018.00164>
- Riantama, G. N. S., Piarsa, I. N., & Sasmita, G. M. A. (2019). Pengaruh Segmentasi Terhadap Hasil Rotasi Citra Menggunakan Metode Minimum Area Rectangle. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*, 7(2), 95–102. <https://doi.org/10.24843/JIM.2019.v07.i02.p01>
- Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut - Interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH 2004 Papers, SIGGRAPH 2004*, 309–314. <https://doi.org/10.1145/1186562.1015720>
- Suryawibawa, I. W. A., Putra, I. K. G. D., & Wirdiani, N. K. A. (2015). Herbs Recognition Based on Android using OpenCV. *International Journal of Image Graphics and Signal Processing*, 2(January), 1–7. <https://doi.org/10.5815/ijigsp.2015.02.01>
- Umbaugh, S. E. (2010). *Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools, Second Edition 2nd Edition*. CRC Press.
- V, R., K, N., & J, I. R. (2020). Foreground algorithms for detection and extraction of an object in multimedia. *International Journal of Electrical and Computer Engineering*, 10(2), 1849–1858. <https://doi.org/10.11591/ijece.v10i2.pp1849-1858>
- Wirdiani, N. K. A., Sukma, S., Sudana, O., & Wibawa, S. (2018). Balinese Papyrus Manuscript Image Segmentation Using DBSCAN Clustering Method. *Journal of Theoretical and Applied Information Technology*, XCVI(17), 5995–6005.
- Xie, S., & Tu, Z. (2015). Holistically-Nested Edge Detection University of California , San Diego. *ICCV*, 1395–1403. <https://doi.org/10.1109/ICCV.2015.164>
- Xuan, L., & Hong, Z. (2018). An improved canny edge detection algorithm. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2017-Novem*, 275–278. <https://doi.org/10.1109/ICSESS.2017.8342913>
- Yi, F., & Moon, I. (2012). Image segmentation: A survey of graph-cut methods. *2012 International Conference on Systems and Informatics (ICSAI2012)*. <https://doi.org/10.1109/ICSAI.2012.6223428>