

## Aplikasi Pengekstrak Gambar Ke Excel dan Uji Ekstraksi dengan Kirsch Untuk Deteksi Tepi

<sup>1)</sup> Yasir Hasan

Universitas Budi Dharma, Medan, Sumatera Utara, Indonesia  
E-Mail: [yasirhasan.kom@gmail.com](mailto:yasirhasan.kom@gmail.com)

<sup>2)</sup> Hery Sunandar

Universitas Budi Dharma, Medan, Sumatera Utara, Indonesia  
E-Mail: [herysun1975@gmail.com](mailto:herysun1975@gmail.com)

### ABSTRACT

Pixel extractor application to retrieve image pixel values and save them in Excel format. Matlab is often used for pixel extraction, but the transfer process to Excel is difficult and long if the image resolution is large. The placement of pixel values in Excel is useful for knowing the process of image processing mathematical formulas. The solution of this application is that the user can select images to be processed and the pixel extraction results are stored in three separate Excel sheets, namely red (R), green (G), blue (B), and Grayscale values. Convenience This is useful for analyzing pixel data for users. In this study, the extracted pixel values were tested using the Kirsch operator for edge detection. Doing a test of one Kirsch kernel on a grayscale sheet. This application is built using the Python programming language and the PySimpleGUI library to create an easy-to-use user interface.

**Keyword :** Excel, Edge\_detection, Image processing, Kirsch, Python.

### PENDAHULUAN

Aplikasi pengekstraksi piksel gambar ke Excel untuk analisis dalam pemrosesan gambar adalah program yang dibuat untuk mengekstraksi nilai piksel dari gambar dan menyimpannya dalam format Excel [1]. Dalam pemrosesan gambar, ekstraksi piksel merupakan langkah penting untuk memahami gambar, dan menggunakan Excel untuk menyimpan data piksel akan memudahkan analisis selanjutnya [2], [3]. File Excel yang menyimpan nilai piksel akan digunakan untuk uji coba deteksi tepi menggunakan kernel Kirsch [4]. Untuk mendapatkan visual yang menarik dari hasil ekstraksi piksel dan hasil deteksi tepi Kirsch di Excel, hanya sedikit perlu setting data pada sheet kerja menggunakan fitur Conditional Formatting.

Solusi yang dihadirkan oleh aplikasi ini adalah pengguna dapat memilih gambar yang ingin diolah dan hasil ekstraksi piksel akan disimpan dalam tiga sheet Excel terpisah, masing-masing untuk warna merah (R), hijau (G), biru (B), dan abu-abu. (Grayscale) nilai piksel. Ini memudahkan pengguna untuk menganalisis data piksel dalam pemrosesan gambar, untuk perbaikan gambar, histogram, dan pemrosesan gambar digital tingkat lanjut. Aplikasi ini dibangun menggunakan bahasa pemrograman Python dan library PySimpleGUI untuk membuat antarmuka pengguna yang mudah digunakan [5].

Seperti diketahui, aplikasi pengolah citra yang sering digunakan untuk mengekstrak piksel citra dapat dilakukan dengan Matlab. Aplikasi Matlab sangat populer untuk

implementasi pengolahan citra dan pengolahan data berbasis matematika lainnya [6], [7]. Tapi aplikasi ini berbayar untuk penggunaannya. Tentu sangat memberatkan bagi pelajar yang ingin mengetahui nilai piksel dari gambar digital. Matlab sangat cocok untuk mengekstraksi nilai piksel berwarna RGB dan Grayscale [1]. Namun, hasil ekstraksi Matlab tidak mudah ditempatkan dan ditata di aplikasi Excel.

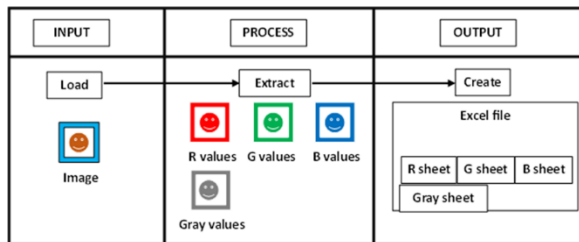
Aplikasi ekstraksi piksel gambar ke Excel adalah solusi yang tepat untuk melakukan ekstraksi piksel dalam pemrosesan gambar. Dengan menggunakan antarmuka pengguna yang mudah digunakan dan menyimpan hasil ekstraksi piksel dalam format Excel [1], [8], Aplikasi ini dapat membantu pengguna melakukan analisis data piksel dengan lebih efisien dan akurat. Diharapkan aplikasi ini dapat membantu mempermudah proses pengolahan citra bagi para peneliti dan praktisi di bidang pengolahan citra [9], [10], [11]. Disini bagian pengolahan citra yang diuji terhadap nilai piksel yang telah diekstrak adalah deteksi tepi menggunakan operator Kirsch.

### BAHAN DAN METODE

Penelitian ini bertujuan untuk mengembangkan aplikasi Python menggunakan PysimpleGUI dan openpyxl untuk membaca nilai piksel dari gambar dan menyimpannya ke file Excel [5], [12]. Aplikasi ini akan dapat membaca nilai piksel RGB dan grayscale untuk setiap piksel pada gambar dan menyimpan hasilnya ke dalam file Excel dalam format yang dapat diproses dan dipahami [1], [13].

## 2.1. Skema aplikasi

Skema penelitian ini meliputi input, proses, dan output. Visualisasi skema aplikasi pada gambar di bawah ini:



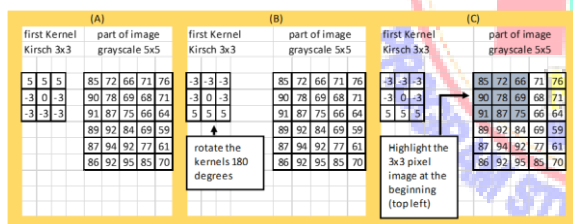
**Gambar 1.** Skema aplikasi

Alur kerja aplikasi didasarkan pada definisi input, proses, dan output, yaitu:

1. Input. Bagian ini melakukan jpg, bmp, png, dan jenis input gambar digital lainnya
2. Process. Bagian ini termasuk
  - a. Dapatkan nilai warna dengan menjiplak isi data file gambar
  - b. Pisahkan nilai warna untuk setiap bagian R, G, B, dan Grayscale
3. Outputs. Bagian ini menulis nilai RGB dan abu-abu dan mencetaknya ke file Excel

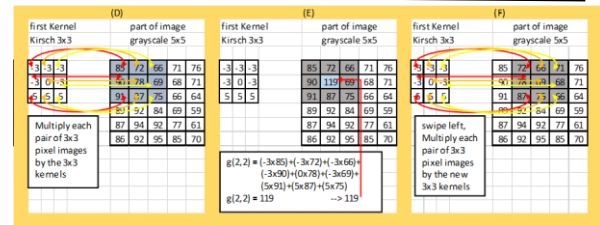
Skema pengujian file Excel dengan menerapkan operator Kirsch untuk deteksi tepi diatur berdasarkan konvolusi, lihat Gambar 2 (A), sebagai berikut:

- A. Siapkan matriks gambar input 6x5 (mode Grayscale) dan kernel Kirsch 3x3.
- B. Lakukan rotasi kernel 180 derajat.
- C. Sesuaikan bagian 3x3 awal (kiri atas) untuk pasangan perkalian setiap tumpang tindih dengan kernel 3x3 dan jumlahkan hasilnya.



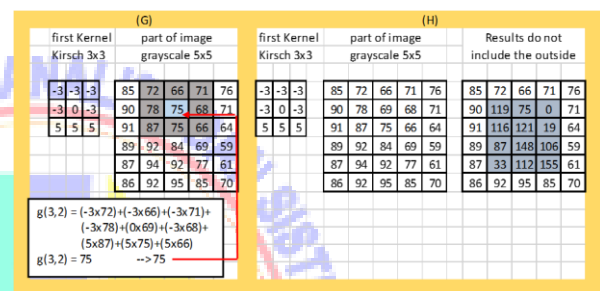
**Gambar 2.** Persiapan konvolusi

- D. Lipat gandakan setiap pasangan gambar yang tumpang tindih dengan kernel dan jumlahkan hasilnya. Lihat Gambar 3 (D).
- E. Tempatkan hasil konvolusi 119 sehingga piksel baru berada di titik tengah gambar 3x3. Lihat Gambar 3(E). Perlu diketahui bahwa jika hasil konvolusi kurang dari 0 maka nilai hasilnya adalah 0, jadi jika hasil konvolusi lebih dari 255 maka nilai hasilnya adalah 255.
- F. Geser satu kolom ke kanan pada gambar input dan sesuaikan 3x3 untuk pasangan perkalian. Lihat Gambar 3 (F).



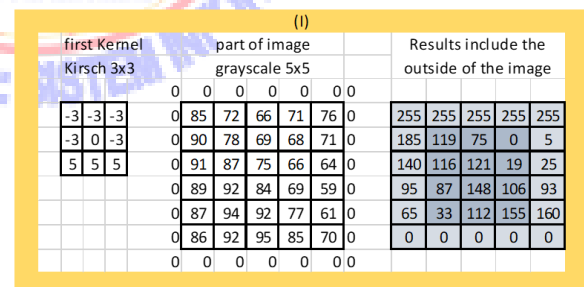
**Gambar 3.** Geser satu kolom

- G. Lakukan konvolusi lagi dengan cara yang sama seperti sebelumnya. Setelah mendapatkan hasil 75, kemudian buat nilai piksel baru. Lihat Gambar 4 (G).
- H. Lakukan metode konvolusi yang sama untuk piksel yang telah selesai. kunjungi piksel dari kolom demi baris. Setelah semuanya mendapatkan hasil konvolusi dengan kernel kirsch, selanjutnya dibuat menjadi output pixel image. Lihat Gambar 4 (F) hasil konvolusi berwarna biru terang dan bagian luarnya tidak berubah karena tidak sesuai dengan matriks kernel.



**Gambar 4.** Hasil tidak termasuk bagian luar

- i. Untuk mendapatkan hasil lilitan luar, yaitu dengan menjumlahkan kolom dan baris dari luar dan masing-masing diberi nilai 0 lagi. sehingga nilai luar dapat berbelit-belit. Lihat Gambar 5 (I) yang menyertakan nilai output gambar luar.



**Gambar 5.** Hasil termasuk bagian luar

Pengujian di atas hanya melibatkan 6x5 matriks grayscale dan bukan hasil dari seluruh matriks. Selain itu, pengujian juga dapat diterapkan pada analisis lain dalam pengolahan citra.

## 2.2. Kode Python

Pengembangan program menggunakan bahasa pemrograman Python dengan memanfaatkan beberapa library seperti PySimpleGUI,

OpenCV, dan Openpyxl [1], [14]. Pustaka PySimpleGUI digunakan untuk membuat antarmuka program yang interaktif dan mudah digunakan [12], [14], [15]. Berikutnya library OpenCV digunakan untuk membaca gambar dan mengambil nilai piksel [16] [17], [18]. Sedangkan openpyxl digunakan untuk menulis data pengukuran ke dalam format file Excel [1].

Kode Python yang akan digunakan sangat sederhana, hal ini dikarenakan Python menyingkat beberapa perintah agar mudah digunakan dan banyak diminati oleh programmer. Di bawah ini adalah penjelasannya:

1. Import library, bagian ini menerapkan library OpenCV, Openpyxl, dan PySimpleGui di bidang kerja Python.  

```
import cv2
import openpyxl
import PySimpleGUI as sg
```
2. Membuat tampilan GUI menggunakan PySimpleGUI  

```
sg.theme('SandyBeach')
layout = [[sg.Text('Please select an image file:'), sg.Input(key='-IN FILE-'), sg.FileBrowse()],
[sg.Text('Please select the storage directory:'), sg.Input(key='-OUT FOLDER-'), sg.FileBrowse()],
[sg.Submit('Process'), sg.Cancel()]]
```
3. Membuat windows  

```
window = sg.Window('Create Pixel Value in Excel (Yasir Hasan-Indonesia)', layout)
#Opens a window and waits for input from the user
event, values = window.read()
# If the 'Cancel' button is pressed or the window is closed then the program exits if event == sg.WINDOW_CLOSED or event == 'Cancel':
exit()
```
4. Baca file gambar dan periksa formatnya  

```
img = cv2.imread(values['-IN FILE-'])
if img is None:
sg.popup_error('The file cannot be read or is not supported.')
exit()
```
5. Buat berkas Excel baru  

```
workbook = openpyxl.Workbook()
# Create a sheet for each RGB color and for grayscale values
sheets = {'Red':
workbook.create_sheet('Red'),
'Green':
workbook.create_sheet('Green'),
'Blue': workbook.create_sheet('Blue'),
'Grayscale':
workbook.create_sheet('Grayscale')}
```
6. Mengambil nilai piksel dari setiap piksel pada gambar dan menuliskannya ke dalam file Excel  

```
for x in range(img.shape[1]):
for y in range(img.shape[0]):
```

```
b, g, r = img[y,x]
gray = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)[y,x]
```

7. Tuliskan nilai piksel ke dalam sheet yang sesuai  

```
sheets['Red'].cell(row=y+1,
column=x+1, value=r)
sheets['Green'].cell(row=y+1,
column=x+1, value=g)
sheets['Blue'].cell(row=y+1,
column=x+1, value=b)
sheets['Grayscale'].cell(row=y+1,
column=x+1, value=gray)
```
8. Simpan file Excelnya  

```
workbook.save(values['-OUT FOLDER-'] +
'file_output.xlsx')
```
9. Tutup jendela dan keluar dari program  

```
window.close()
```

## HASIL DAN PEMBAHASAN

Pengembangan program untuk membaca nilai piksel dari gambar dan menampilkannya di layar menggunakan pustaka PySimpleGUI, OpenCV, dan Openpyxl. Program yang dibuat dapat membaca gambar dalam format grayscale atau warna, dan menampilkan nilai piksel dalam format RGB atau grayscale.

### 3.1. GUI display

Tampilan GUI aplikasi ini sangat sederhana, tampilannya dapat dilihat pada Gambar 6. Berdasarkan tampilan GUI aplikasi dapat dijelaskan. tampilan gui termasuk:

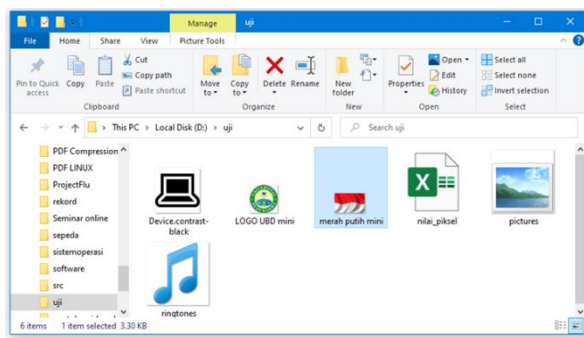
1. Tekan tombol browse atas untuk memilih gambar yang akan diekstrak, setelah gambar dipilih, jalur file gambar tercantum di kotak teks pemilihan file gambar.
2. Tekan tombol browse kedua untuk memilih folder/direktori yang akan digunakan sebagai file output Excel dan disana file tersebut akan disimpan dengan nama file\_output.xlsx yang sudah berisi sheet RGB dan Grayscale.
3. Tekan tombol proses untuk mengekstrak atau tekan tombol batal untuk membatalkan ekstraksi



Gambar 6. GUI display

Pada saat menekan tombol browse untuk memilih file gambar Merah Putih Mini, akan muncul jendela dan pilih file gambar seperti pada Gambar 7 dibawah ini.

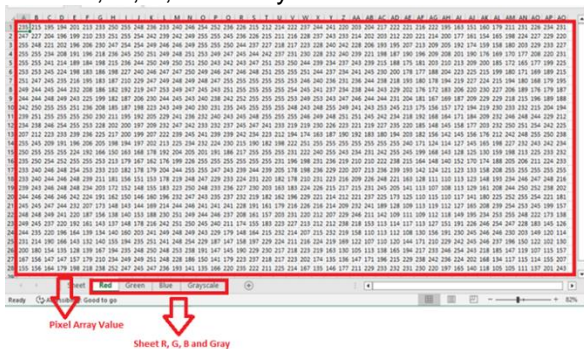




**Gambar 7.** Pilih file gambar Merah Putih Mini

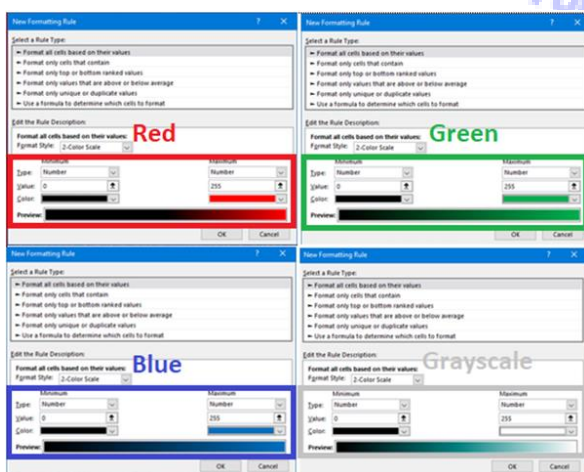
### 3.2. Sheet mewarnai R, G, B, dan Gray

Setelah selesai, file file\_output.xlsx dapat dibuka dengan aplikasi Microsoft Excel. Pada Gambar 8 dapat dilihat nilai piksel dalam bentuk array. Nilai larik piksel 2d ditemukan di setiap sheet R, G, B, dan Grayscale.



**Figure 8.** Nilai dan sheet larik piksel 2d

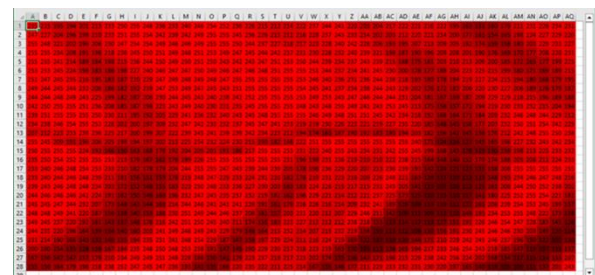
Agar setiap sheet tampak memiliki warna yang berbeda, hal ini dapat dilakukan dengan fitur Conditional Formatting pada menu home di Microsoft Excel [19]. Sebuah jendela muncul untuk membuat perubahan format warna baru. Pada Gambar 9, buat perubahan untuk setiap sheet dengan menyatel nilai terkecil ke 0 dan terbesar ke 255, dan memilih warna untuk nilainya [20]. Untuk mempermudah bagian ini, Anda dapat melihat kotak spidol berwarna pada Gambar 9 di bawah ini.



**Figure 9.** New formatting rule color

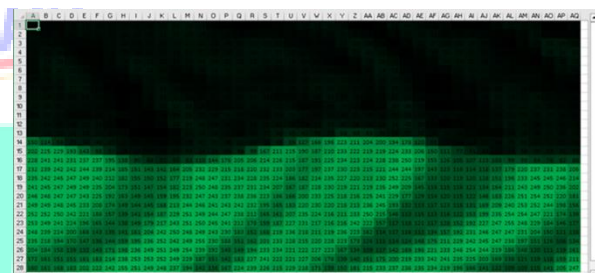
Setelah semua sheet sudah diberi warna untuk nilai pikselnya, maka ketika itu bisa

digunakan untuk analisis nilai citra, seperti filtering, peningkatan citra, histogram, atau algoritma lainnya dalam pengolahan citra. [19], [21]. Gambar sheet warna dapat dilihat pada Gambar 5, 6, 7, dan 8 di bawah ini.



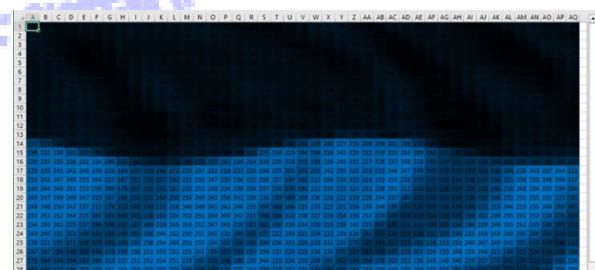
**Figure 10.** Red sheet

Gambar 10 dapat dilihat dari tampilan Excel setelah dilakukan format baru dengan ketentuan nilai warna terkecil adalah 0 dengan warna hitam dan nilai warna tertinggi adalah 255 dengan warna merah. Sehingga hasil tampilan area kerja Excel berwarna merah berdasarkan nilai intensitasnya.



**Figure 11.** Green sheet

Gambar 11 dapat dilihat dari tampilan Excel setelah dilakukan format baru dengan ketentuan nilai warna terkecil adalah 0 dengan warna hitam dan nilai warna tertinggi adalah 255 dengan warna hijau. Sehingga hasilnya menampilkan bidang kerja Excel berwarna hijau berdasarkan nilai intensitasnya.



**Figure 12.** Blue sheet

Gambar 12 dapat dilihat dari tampilan Excel setelah dilakukan format baru dengan ketentuan nilai warna terkecil adalah 0 dengan warna hitam dan nilai warna tertinggi adalah 255 dengan warna biru. Sehingga hasilnya menampilkan bidang kerja Excel berwarna biru berdasarkan nilai intensitasnya.

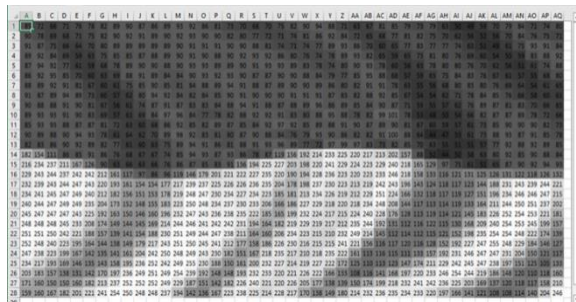


Figure 13. Grayscale sheet

Pada Gambar 13 di atas lakukan hal yang sama seperti sebelumnya yaitu lakukan format baru dengan syarat nilai warna terkecil 0 dengan warna hitam dan nilai warna tertinggi 255 dengan warna putih. Sehingga hasilnya menampilkan bidang kerja Excel grayscale berdasarkan nilai intensitasnya.

### 3.3. Pengujian deteksi tepi untuk piksel sheet Grayscale

Eksperimen ini tidak lengkap jika tidak diuji dengan teknik analisis image processing. Untuk pengujian dilakukan deteksi tepi karena perubahan warna terlihat berdasarkan nilai piksel dan perhitungannya juga tidak rumit, tetapi dapat disimulasikan melalui Excel. Deteksi tepi akan disampling pada sheet grayscale dan operator deteksi tepi yang digunakan adalah operator Kirsch. Konvolusi dengan kernel Kirsch pertama sebanyak 8 kernel Kirsch dan bentuk perhitungannya sama dengan teknik konvolusi konvensional.

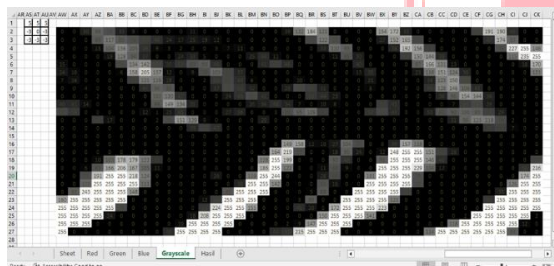


Figure 14. Hasil pengujian deteksi tepi dari ekstraksi piksel grayscale dengan operator Kirsch

Pada Gambar 14 di atas posisi kernel Kirsch yang digunakan adalah [4] :

5	5	5
-3	0	-3
-3	-3	-3

Dengan menggunakan kernel ini, dilakukan konvolusi dengan nilai piksel grayscale pada Gambar 9 dan menghasilkan deteksi tepi pada Gambar 14. Untuk hasil yang mengalami perubahan, dapat dilakukan perhitungan teknis lainnya pada pengolahan citra. Hal menarik yang dapat dilihat adalah nilai dari bilangan piksel tersebut dapat digunakan untuk tahap selanjutnya.

## KESIMPULAN

Pengembangan program menggunakan bahasa pemrograman Python dengan memanfaatkan beberapa library seperti PySimpleGUI, Openpyxl, dan OpenCV untuk membaca nilai piksel dari suatu gambar dan menampilkannya dalam format RGB atau grayscale pada sheet kerja Excel. Hasil pengujian menunjukkan bahwa program yang dibuat dapat membaca gambar dengan berbagai jenis format seperti JPEG, PNG, dan BMP dengan baik dan dapat membaca nilai pixel dari gambar dengan benar. Program yang dibuat juga mampu mengatasi beberapa keterbatasan seperti mengolah gambar berukuran besar dan membaca gambar dengan format yang kompleks. Selain itu, programnya.

## DAFTAR PUSTAKA

- [1] Y. Colella, A. S. Valente, L. Rossano, T. A. Trunfio, A. Fiorillo, and G. Improta, "A Fuzzy Inference System for the Assessment of Indoor Air Quality in an Operating Room to Prevent Surgical Site Infection," *Int J Environ Res Public Health*, vol. 19, no. 6, Mar. 2022, doi: 10.3390/ijerph19063533.
- [2] P. Chandana, M. Borkakoty, and A. Professor, "CONVERSION OF IMAGE TO EXCEL USING OCR TECHNIQUE," 2022. [Online]. Available: [www.irjmets.com](http://www.irjmets.com)
- [3] P. A. Larbi, "RGBEXCEL: An RGB Image Data Extractor and Exporter for Excel Processing," *Signal Image Process*, vol. 7, no. 1, pp. 01–09, Feb. 2016, doi: 10.5121/sipij.2016.7101.
- [4] C. WANG, Y. LI, and Y. QI, "Comparison Research of Capability of Several Edge Detection Operators," 2015. doi: 10.2991/itms-15.2015.188.
- [5] A. L. S. Saabith, T. Vinothraj, and M. Fareez, "POPULAR PYTHON LIBRARIES AND THEIR APPLICATION DOMAINS," *International Journal of Advance Engineering and Research Development*, vol. 7, no. 11, 2020, [Online]. Available: <https://www.researchgate.net/publication/349828209>
- [6] S. Mukherjee and A. Singh Poonia, "MATLAB Based Vehicle Number Plate Recognition," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 3, no. 2, pp. 25–39, 2015, [Online]. Available: [www.ijraset.com](http://www.ijraset.com)
- [7] J. FathimasonJ, Bibis.S, Aswanth.R, and Gayatri S, "Underwater Image Restoration Using UICCS Method in Matlab," 2018. [Online]. Available: [www.ijntr.org](http://www.ijntr.org)



- [8] K. Padmavathi and K. Thangadurai, "Implementation of RGB and grayscale images in plant leaves disease detection - Comparative study," *Indian J Sci Technol*, vol. 9, no. 6, 2016, doi: 10.17485/ijst/2016/v9i6/77739.
- [9] J. Wang and S. Lee, "Data augmentation methods applying grayscale images for convolutional neural networks in machine vision," *Applied Sciences (Switzerland)*, vol. 11, no. 15, Aug. 2021, doi: 10.3390/app11156721.
- [10] A. N. Parikh, "Automatic Image Colourizer," 2021. [Online]. Available: [www.ijert.org](http://www.ijert.org)
- [11] Mrs. A. K H, M. B E, and Mrs. C. A S, "Python - Based Image Processing," *International Journal of Scientific Research and Management*, vol. 9, no. 11, pp. 635–638, Nov. 2021, doi: 10.18535/ijser/v9i11.ec03.
- [12] V. Nara, H. Pottipati, and R. Athavale, "MACHINE INTELLIGENCE FOR BRAIN SEGMENTATION: A TOOL TO IDENTIFY BRAIN ILLNESSES THROUGH SEGMENTATION," *International Journal of Social Science and Economic Research*, vol. 07, no. 09, pp. 3115–3125, 2022, doi: 10.46609/IJSSER.2022.v07i09.023.
- [13] D. S. Shreya, "DIGITAL IMAGE PROCESSING AND RECOGNITION USING PYTHON," 2021. [Online]. Available: <http://www.ijeast.com>
- [14] S. K. Rajamani and R. S. Iyer, "Machine Learning-Based Mobile Applications Using Python and Scikit-Learn," 2023, pp. 282–306. doi: 10.4018/978-1-6684-8582-8.ch016.
- [15] V. Kocheta, S. Shinde, K. Nadkar, S. Jambhulkar, and S. Wankhede, "Dictation Module Using Automatic Speech Recognition in Machine Learning," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 10, no. 5, 2021, doi: 10.17148/IJARCCCE.2021.105123.
- [16] P. Kaler, "Study of Grayscale image in Image processing," *IJRITCC*, vol. 4, no. 11, pp. 309–311, 2016, [Online]. Available: <http://www.ijritcc.org>
- [17] A. Willis and K. Hasan, "Simple Computer Vision Algorithm Production Using OpenCV for 'Virtual Ecosystem' Project," *23 IJNMT*, vol. VII, no. 1, 2020.
- [18] B. Hangün and Ö. Eyecioğlu, "Performance Comparison Between OpenCV Built in CPU and GPU Functions on Image Processing Operations," 2017.
- [19] DrSenbagavalliM, DrVinodhaK, and DrJesudasT, "Visualization of Enterprise Grade Cloud Architecture for the Most Cost-Effective Application Deployment using Mining Tool," *International Journal for Research in Engineering Application & Management (IJREAM)*, vol. 05, no. 10, pp. 2454–9150, 2020, doi: 10.35291/2454-9150.2020.0036.
- [20] K. Yamini, K. Sai Swetha, P. L. Prasanna, M. Rupa, V. Swathi, and V. Rao Maddumala, "IMAGE COLORIZATION WITH DEEP CONVOLUTIONAL OPEN CV," vol. 11, 2020, [Online]. Available: [www.jespublication.com](http://www.jespublication.com)
- [21] A. Singh and S. Anekar, "Text to Image using Deep Learning; Text to Image using Deep Learning," *www.ijert.org*, vol. 10, 2021, [Online]. Available: [www.ijert.org](http://www.ijert.org)