Pengolahan Citra Digital

Topik :

Identifikasi Citra Daun Tanaman Tomat Menggunakan Ekstraksi Fitur GLCM, Klasifikasi KNN dan Menghitung Tingkat Kerusakan Daun menggunakan Metode Laplacian Of Gaussian

Anggota Kelompok

Melania Syafrida P.

18081010011

Dwi Wahyu Effendi

18081010017

Dedy Ramadhan

18081010020

Anggota Kelompok

Amirah Aulia Fitri

18081010029

Quincy Mayferta P.P.A

19081010127

Pendahuluan

Permasalahan pada daun tanaman tomat tidak hanya muncul dari bentuk buah dan kualitas tomat tersebut, tapi permasalahan juga muncul dari beberapa penyakit atau hama yang menyerang dari beberapa bagian tanaman tomat itu sendiri seperti pada bagian daun tomat yang akan mempengaruhi pertumbuhan tanaman tomat tersebut.

ekstraksi fitur GLCM dan Klasifikasi KNN digunakan untuk mengklasifikasi daun tanaman tomat tersebut sehat atau sakit. lalu kita mendeteksi tepi menggunakan metode Laplacian of Gaussian (LoG). Setelah mendeteksi tepi, didapat nilai rata-rata Mean Square Error dari daun tomat sehat dan daun tomat sakit

Metode

Tahapan Metode

1. Desain Program
2. Ekstraksi Fitur dan Klasifikasi KNN
3. Deteksi Tepi (Log) dan Menghitung MSE

Desain Program

Pada Program untuk mengetahui daun tomat tersebut sehat atau rusak, kita melakukan Pengolahan citra dengan tahapan ekstraksi fitur tekstur mengunakan algoritma GLCM dan untuk mengklasifikasi citra tersebut sehat atau rusak kita menggunakan Klasifikasi KNN.

setelah itu untuk menghitung MSE (Mean Square Error) dari citra daun tomat. sehat atau rusak kita harus mengubah citra berwarna ke greyscale. lalu selanutnya kita menggunakan Gaussiaan Filter untuk memperhalus atau (blur) citra daun tomat. lalu selanjutnya kita menggunakan deteksi tepi laplaciaan untuk mendeteksi tepi daun tomat. lalu baru kita bisa menghitung nilai MSE dari citra daun tomat tersebut

Grayscale

$$
\begin{aligned}
\textit{Grayscale} &= (0{,}2989 \times R) + (0{,}5870 \times G) \\
&\quad + (0{,}1140 \times B) \\
&= (0{,}2989 \times 217) + (0{,}5870 \times 224) \\
&\quad + (0{,}1140 \times 219) \\
&= 62{,}8866 + 131{,}448 + 24{,}966 \\
&= 219{,}3006
\end{aligned}
$$

Perhitungan yang sama dilakukan pada baris dan kolom lainnya.

Grayscale adalah warna-warna piksel yang berada dalam rentang gradasi warna hitam dan putih.

Mengubah warna ke grayscale pada pengubahan sebuah gambar menjadi grayscale dapat dilakukan dengan cara mengambil semua pixel pada gambar kemudian warna tiap pixel akan diambil informasi mengenai 3 warna dasar yaitu merah, biru dan hijau (melalui fungsi warnatoRGB), ketiga warna dasar ini akan dijumlahkan kemudian dibagi tiga sehingga didapat nilai ratarata.

GLCM

Gray Level Co-occurence Matrix adalah suatu matrik kookurensi yang elemenelemennya merupakan jumlah kemunculan piksel-piksel yang memiliki nilai tingkat keabuan tertentu, di mana pasangan piksel itu berada pada jarak (d) dan sudut tertentu (Ɵ).

Orientasi sudut yang digunakan adalah yaitu sudut 0°, 45°, 90°, 135°, 180° dst Sedangkan jarak antar piksel biasanya ditetapkan sebesar 1 piksel, atau 2 piksel

Energi

Energi menyatakan ukuran konsentrasi pasangan dengan intensitas keabuan tertentu pada matriks. Nilai energi (E) dapat dihitung dengan persamaan

$$E = \sum_{i,j} (p_{i,j})^2 \qquad (1)$$

Dengan :

i = tingkat keabuan baris ke – i

j = tingkat keabuan kolom ke – j

$p_{i,j}$ = Peluang keabuan baris ke – i, kolom ke – j

Kontras

Kontras menyatakan perbedaan intensitas antara nilai tertinggi (terang) dan nilai terendah (gelap) dari sepasang piksel yang saling berdekatan

$$CON = \sum_{i,j} i - j^2 \, p_{i,j}$$

Homogenitas

Untuk suatu matrik citra, nilai homogenitasnya ditentukan dari derajat keabuan yang sejenis

$$H = \sum_{i,j} \frac{p_{i,j}}{1+|i-j|}$$

Correlation

Mengukur kemungkinan terjadinya gabungan dari pasangan piksel yang ditentukan.

'correlation':

$$\sum_{i,j=0}^{levels-1} P_{i,j} \left[\frac{(i-\mu_i)\,(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right]$$

KNN

Prinsip

Misalnya kita memiliki data-data yang telah terkategori menjadi kategori A (tosca) dan B (kuning). Lalu muncul data baru yang ingin di kategorikan.
Berdasarkan gambar, tentunya kita mengklasifikasikannya dalam Kategori A karena letaknya yang berdekatan dengan kategori tersebut.

Pengertian

K-nearest neighbors atau knn adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (train data sets), yang diambil dari k tetangga terdekatnya (nearest neighbors). Dengan k merupakan banyaknya tetangga terdekat.

Cara Kerja

- Tentukan jumlah tetangga (K) yang akan digunakan untuk pertimbangan penentuan kelas.
- Hitung jarak dari data baru ke masing-masing data point di dataset.
- Ambil sejumlah K data dengan jarak terdekat, kemudian tentukan kelas dari data baru tersebut.

Penentuan nilai K

Untuk menggunakan algoritma k nearest neighbors, perlu ditentukan banyaknya k tetangga terdekat yang digunakan untuk melakukan klasifikasi data baru. Penentuan nilai K ini tidak ada rumus pastinya. Namun satu tips yang dapat dipertimbangkan, yakni jika kelas berjumlah genap maka sebaiknya nilai K-nya ganjil, sebaliknya jika kelas berjumlah ganjil maka sebaiknya nilai K-nya genap . Penentuan nilai k dipertimbangkan berdasarkan banyaknya data yang ada dan ukuran dimensi yang dibentuk oleh data. Semakin banyak data yang ada, angka k yang dipilih sebaiknya semakin rendah. Namun, semakin besar ukuran dimensi data, angka k yang dipilih sebaiknya semakin tinggi.

Klasifikasi

Proses klasifikasi dilakukan dengan mencari titik terdekat dari data baru (nearest neighbor). Goal-nya adalah mendapatkan data sebanyak k dengan jarak terdekat dengan data baru. Teknik pencarian tetangga terdekat yang umum dilakukan dengan menggunakan formula jarak euclidean.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Contoh

$$\sqrt{(8-10)^2+(1-5)^2+(2-9)^2}$$

i	x	y	z	Kelompok
1	8	1	2	2
2	5	5	5	2
3	7	10	7	2
4	6	8	10	2
5	2	5	3	1
6	4	2	2	1
7	2	4	1	1
8	3	2	7	1
9	10	5	9	?

Distance	K		
	3	5	7
8,306623863		2	2
6,403124237	2	2	2
6,164414003	2	2	2
5,099019514	2	2	2
10			1
9,695359715			1
11,35781669			
7,874007874		1	1
Kelompok	2	2	2

Filter Gaussian

filter Gaussian sangat baik untuk menghilangkan noise yang bersifat sebaran nomal

filter Gaussian menempatkan warna transisi yang signifikan dalam sebuah image, kemudian membuat warna-warna pertengahan untuk menciptakan efek lembut pada sisi-sisi sebuah image

Contoh :

1/16

1	2	1
2	4	2
1	2	1

Kernel 3x3

1/273

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Kernel 5x5

1/1003

0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0

Kernel 7x7

Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Operator turunan kedua disebut juga operator Laplace.

Operator Laplace mendeteksi lokasi tepi lebih akurat khususnya pada tepi yang curam.

Pada tepi yang curam, turunan keduanya mempunyai persilangan nol (zero-crossing), yaitu titik di mana terdapat pergantian tanda nilai turunan kedua dari positif ke negative atau sebaliknya.

Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

kernel konvolusi operator laplacian

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \end{bmatrix}$$

(i) Citra semula

$$\begin{bmatrix} * & * & * & * & * & * & * \\ * & 0 & +4 & -4 & 0 & 0 & * \\ * & 0 & +4 & -4 & 0 & 0 & * \\ * & 0 & +4 & -4 & 0 & 0 & * \\ * & * & * & * & * & * & * \end{bmatrix}$$

(ii) Hasil konvolusi

Laplacian

- Contoh berikut memperlihatkan memperlihatkan pendeteksian tepi vertikal dengan operator Laplace:

$$\begin{bmatrix} 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \end{bmatrix}$$

(i) Citra semula

$$\begin{bmatrix} * & * & * & * & * & * & * \\ * & 0 & +4 & -4 & 0 & 0 & * \\ * & 0 & +4 & -4 & 0 & 0 & * \\ * & 0 & +4 & -4 & 0 & 0 & * \\ * & * & * & * & * & * & * \end{bmatrix}$$

(ii) Hasil konvolusi

Satu baris dari hasil pendeteksian tepi: $0 \quad +4 \mid -4 \quad 0 \quad 0$

persilangan nol

Mean Square Error (MSE)

Mean Squared Error (MSE) adalah Rata-rata Kesalahan kuadrat diantara nilai aktual dan nilai peramalan. Metode Mean Squared Error secara umum digunakan untuk mengecek estimasi berapa nilai kesalahan pada peramalan. Nilai Mean Squared Error yang rendah atau nilai mean squared error mendekati nol menunjukkan bahwa hasil peramalan sesuai dengan data aktual dan bisa dijadikan untuk perhitungan peramalan di periode mendatang. Metode Mean Squared Error biasanya digunakan untuk mengevaluasi metode pengukuran dengan model regressi atau model peramalan seperti Moving Average, Weighted Moving Average dan Analisis Trendline.

Cara menghitung Mean Squared Error (MSE) adalah melakukan pengurangan nilai data aktual dengan data peramalan dan hasilnya dikuadratkan (squared) kemudian dijumlahkan secara keseluruhan dan membaginya dengan banyaknya data yang ada.

$$MSE = \frac{\sum_{t=1}^{n}(At - Ft)^2}{n}$$

Kesimpulan

Dari pembahasan kita dapat mengetahui citra daun mana yang sehat dan citra daun mana yang rusak melalui proses ekstrasi fitur GLCM dan Klasifikasi KNN.

Dari pembahasan sebelumnya juga dapat disimpulkan bahwa deteksi tepi daun tanaman kopi pada citra dapat dilakukan dengan metode Laplacian of Gaussian (LoG) dan operator Laplacian.

Dari proses deteksi tepi terhadap daun kopi didapat nilai rata-rata yaitu Mean Square Error yang di gunakan untuk mengetahui tingkat kerusakan pada daun kopi

Nilai MSE akan semakin besar berdasarkan tingkat kerusakan daun.

Demo Program

Kalian siap?

Terima kasih!

Ada Pertanyaan?