

PENERAPAN ALGORITMA CANNY UNTUK DETEKSI TEPI MENGGUNAKAN PYTHON DAN OPENCV

Saluky^{1*}, Yoni Marine²

¹ Jurusan Tadris Matematika, FITK, IAIN Syekh Nurjati Cirebon

² Etunas Sukses Sistem

Email: luke4line@gmail.com

Abstrak: Deteksi objek merupakan salah satu langkah dalam sebuah pengenalan objek pada bidang computer vision. Tepi dari gambar mencirikan batas yang membedakan dengan objek lainnya dan karena itu menjadi masalah yang sangat penting dalam pemrosesan gambar. Deteksi tepi gambar yang akurat secara signifikan dapat mengurangi jumlah data dan memfilter informasi yang tidak berguna, dengan tetap mempertahankan properti struktural penting dalam gambar. Karena deteksi tepi merupakan yang terdepan dalam pemrosesan gambar untuk deteksi objek, sangat penting untuk memiliki pemahaman yang baik tentang algoritme deteksi tepi. Dalam penelitian ini menerapkan metode canny untuk deteksi tepi dengan menggunakan python. Hasilnya algoritma canny untuk deteksi tepi memiliki performa yang lebih baik dibandingkan dengan algoritma lainnya seperti LoG (Laplacian of Gaussian), Robert Cross Operator, Prewitt dan Sobel.

Kata Kunci: Deteksi Tepi, Canny, Computer Vision

Abstract: Object detection is one step in an object recognition in the field of computer vision. The edges of the image characterize the boundaries that distinguish it from other objects and are therefore a very important problem in image processing. Accurate image edge detection can significantly reduce the amount of data and filter out useless information, while retaining important structural properties in the image. Since edge detection is at the forefront of image processing for object detection, it is very important to have a good understanding of edge detection algorithms. In this study, applying canny edge detection using python and opencv and also compared with other image processing methods. The result is that canny edge detection has better performance compared to other algorithms such as LoG (Laplacian of Gaussian), Robert Cross Operator, Prewitt and Sobel.

Keywords : Edge Detection, Canny, Computer Vision

1. PENDAHULUAN

Deteksi objek merupakan salah satu dasar dalam computer vision, salah satu yang penting adalah mengidentifikasi objek kemudian mengklasifikasi dalam kelas objek [1][2][3]. Informasi dasar yang dibutuhkan dari sebuah objek secara sederhana adalah informasi tepi gambar yang merupakan salah satu yang paling penting sehingga dapat menggambarkan garis besar target, posisi relatif dalam area target, dan informasi penting lainnya [4]. Deteksi tepi adalah salah satu yang paling penting proses dalam pemrosesan gambar, dan hasil deteksi akan langsung mempengaruhi analisis gambar[5]. Deteksi tepi mengacu pada proses mengidentifikasi dan menemukan diskontinuitas tajam dalam suatu gambar [6].

Diskontinuitas adalah perubahan tiba-tiba dalam intensitas piksel yang mencirikan batas-batas objek dalam sebuah adegan. Metode klasik deteksi tepi melibatkan

penggabungan gambar dengan operator (filter 2-D), yang dibangun agar peka terhadap gradien besar pada gambar sambil mengembalikan nilai nol di wilayah yang seragam [7]. Ada banyak sekali operator deteksi tepi yang tersedia, masing-masing dirancang agar peka terhadap jenis tepi tertentu. Variabel yang terlibat dalam pemilihan operator deteksi tepi meliputi orientasi Tepi, lingkungan Kebisingan, dan struktur Tepi. Geometri operator menentukan arah karakteristik yang paling sensitif terhadap tepi. Operator dapat dioptimalkan untuk mencari horizontal, vertikal, atau tepi diagonal [8]. Deteksi tepi sulit dilakukan pada gambar yang berderau, karena derau dan tepian mengandung konten frekuensi tinggi. Upaya untuk mengurangi kebisingan menghasilkan tepi yang kabur dan terdistorsi. Operator yang digunakan pada gambar dengan noise biasanya memiliki cakupan yang lebih besar, sehingga rata-rata data dapat cukup untuk mengurangi piksel

noise yang dilokalkan [9]. Ini menghasilkan lokalisasi tepi yang terdeteksi kurang akurat. Tidak semua sisi melibatkan langkah perubahan intensitas. Efek seperti pembiasan atau fokus yang buruk dapat menghasilkan objek dengan batasan yang ditentukan oleh perubahan intensitas secara bertahap Operator harus dipilih agar responsif terhadap perubahan bertahap dalam kasus tersebut. Jadi, ada masalah deteksi tepi yang salah, tepi yang hilang, lokalisasi tepi, waktu komputasi yang tinggi dan masalah karena kebisingan dan lainnya. Oleh karena itu, tujuannya adalah untuk melakukan perbandingan berbagai teknik deteksi tepi dan menganalisis kinerja berbagai teknik di kondisi yang berbeda sehingga didapatkan model deteksi yang efektif dan akurat. Pada makalah ini deteksi tepi diterapkan dengan menggunakan Python dan library opencv.

2. TINJAUAN PUSTAKA

A. Python dan OpenCV

Python adalah bahasa pemrograman berorientasi objek yang digunakan dalam berbagai domain aplikasi [10]. Ini menawarkan dukungan kuat untuk integrasi dengan bahasa dan alat lain, dan dilengkapi dengan ekstensif perpustakaan standar. Lebih tepatnya, berikut adalah beberapa fitur yang membedakan Python:

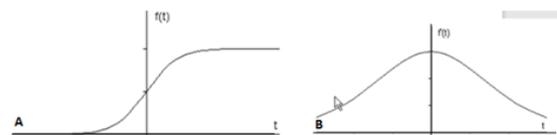
- Sintaks yang sangat jelas dan mudah dibaca.
- Kemampuan introspeksi yang kuat.
- Modularitas penuh.
- Penanganan kesalahan berbasis pengecualian.

OpenCV (Open Source Computer Vision Library) adalah pustaka pengolahan citra digital secara real-time yang dibuat oleh Intel. Program ini bersifat open source sehingga semua programmer python ataupun C dapat dengan bebas menggunakan pustaka opencv. Pustaka ini didedikasikan sebagai besar untuk pengolahan citra digital secara real time.

B. Deteksi Tepi Canny

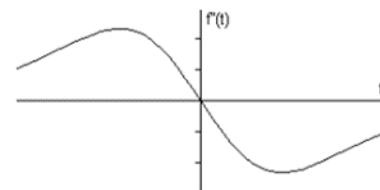
Ada banyak cara untuk melakukan deteksi tepi. Namun, sebagian besar metode yang berbeda dapat dikelompokkan menjadi dua kategori yaitu 1. deteksi tepi berbasis gradien yang merupakan metode gradien mendeteksi tepi dengan mencari maksimum dan minimum pada turunan pertama dari

gambar. 2. Deteksi Tepi berbasis Laplacian yaitu metode Laplacian mencari zero crossing pada turunan kedua dari citra untuk menemukan edge. Sebuah edge memiliki bentuk satu dimensi dari ramp dan menghitung turunan dari gambar dapat menyorot lokasinya. Misalkan kita memiliki sinyal berikut, dengan tepi yang ditunjukkan oleh lompatan intensitas. Jika kita mengambil gradien dari sinyal ini (yang, dalam satu dimensi, hanyalah turunan pertama terhadap t) kita mendapatkan yang berikut:



Gambar 1: A. Penentuan tepi berdasarkan 2 dimensi, B. Turunan dari fungsi t dengan rangkaian filter gradient

Selanjutnya, ketika turunan pertama maksimum, turunan kedua adalah nol. Akibatnya, alternatif lain untuk menemukan lokasi sebuah sisi adalah dengan mencari nol pada turunan kedua sehingga dapat ditunjukkan pada gambar 2.



Gambar 2: Turunan ke 2 dari fungsi t

Berikut ini langkah pada deteksi tepi:

B.1 Penyaringan Gambar

Langkah pertama dari algoritma deteksi tepi tradisional adalah gambar halus. Canny menyimpulkan turunan pertama dari Gaussian fungsi, yang merupakan perkiraan terbaik dari tepi optimal operator deteksi. Pilih fungsi Gaussian yang sesuai untuk menghaluskan gambar sesuai dengan baris dan kolom masing-masing, yaitu menjalankan operasi konvolusi ke gambar matriks. Karena operasi konvolusi memenuhi komutatif hukum dan hukum asosiatif, algoritma Canny umumnya menggunakan fungsi Gaussian dua dimensi (seperti yang ditunjukkan pada (1)) untuk menghaluskan gambar dan menghilangkan kebisingan.

$$G(x,y)=\exp[-(x^2+y^2)/2\sigma^2]/2\pi\sigma^2 \quad (1)$$

di mana σ singkatan dari parameter filter Gauss, dan itu mengontrol perluasan gambar smoothing. Sedangkan x dan y adalah piksel.

B.2 Perhitungan Gradien Gambar

Langkah kedua adalah menghitung besaran dan arah gradien gambar. Algoritma Canny tradisional mengadopsi perbedaan terbatas dari 22 area tetangga untuk dihitung nilai dan arah gradien citra [11]. Urutan pertama pendekatan turunan parsial pada arah x dan y dapat diperoleh dari rumus berikut ini:

$$Ex[i,j] = (I[i+1, j] - I[i, j] + I[i+1, j+1] - I[i, j+1])/2 \quad (2)$$

$$Ey[i,j] = (I[i, j+1] - I[i, j] + I[i+1, j+1] - I[i+1, j])/2 \quad (3)$$

Dimana Ex adalah garis pada x dan i, j adalah perambatan dari titik.

Oleh karena itu, template operator perhitungan gradien gambar adalah:

$$G_x = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad (4)$$

$$G_y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \quad (5)$$

Dimana G_x dan G_y adalah operator untuk perhitungan gradient gambar.

Dari persamaan diatas didapatkan magnitude gradient sebagai berikut:

$$M(i,j) = \sqrt{E_x[i,j]^2 + E_y[i,j]^2} \quad (6)$$

Dimana M adalah magnitud gradient, sedangkan Ex dan Ey adalah garis pada x dan y .

Dengan azimuth gradient gambar sebagai berikut:

$$\theta(i,j) = \arctan\left(\frac{E_x[i,j]}{E_y[i,j]}\right) \quad (7)$$

Dimana θ adalah azimuth gradient.

B.3 Non-maximum Suppression(NMS)

Non Maximum Suppression (NMS) adalah teknik yang digunakan dalam banyak tugas visi komputer. Ini adalah kelas algoritme untuk memilih satu entitas (mis., Kotak pembatas) dari banyak entitas yang tumpang tindih. Kita dapat memilih kriteria seleksi

untuk sampai pada hasil yang diinginkan. Kriteria yang paling umum adalah beberapa bentuk angka probabilitas dan beberapa bentuk ukuran yang tumpang tindih (misalnya Intersection over Union) [12].



Gambar 3: Penerapan Intersection of Union pada Non Maximum Suppression

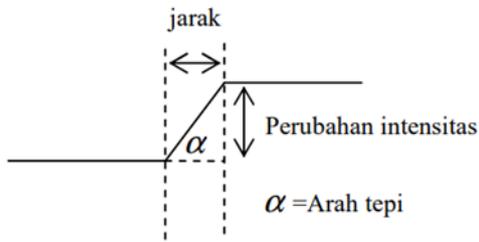
Setelah memperoleh gambar besaran gradien $M[i,j]$, perlu dilakukan penekanan non-maksimum pada gambar untuk memposisikan tepi secara akurat. Proses NMS dapat membantu menjamin bahwa setiap sisi memiliki lebar satu piksel. Algoritma Canny menggunakan area bertetangga 3×3 yang terdiri dari delapan arah untuk melakukan interpolasi terhadap besaran gradien sepanjang arah gradien. Jika besarnya $M[i,j]$ lebih besar dari dua hasil interpolasi pada arah gradien, itu akan menjadi ditandai sebagai titik tepi kandidat, jika tidak maka akan ditandai sebagai titik bukan tepi. Oleh karena itu, citra tepi kandidat adalah diperoleh melalui proses.

B.4 Pengecekan dan Menghubungkan Tepi

Algoritma Canny mengadopsi metode ambang ganda untuk pilih titik tepi setelah melakukan non-maksimum penekanan. Piksel yang besaran gradiennya di atas ambang tinggi akan ditandai sebagai titik tepi, dan itu yang besarnya gradien berada di bawah ambang rendah akan ditandai sebagai titik non-tepi, dan sisanya akan ditandai sebagai calon titik tepi. Kandidat poin tepi yang terhubung dengan titik tepi akan ditandai sebagai titik tepi. Ini metode mengurangi pengaruh kebisingan di tepi final gambar tepi.

Tepi (edge) adalah perubahan nilai intensitas derajat keabuan yang mendadak besar dalam jarak yang dekat. Suatu titik (x,y) dikatakan sebagai tepi bila titik tersebut mempunyai perbedaan nilai piksel yang tinggi dengan nilai piksel tetangganya.

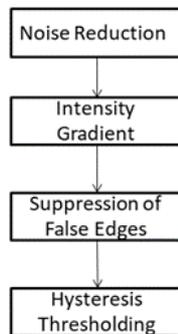
Gambar 4 menunjukkan salah satu model tepi untuk satu dimensi.



Gambar 4: Model tepi satu dimensi

3. METODE

Canny Edge Detection adalah salah satu metode deteksi tepi paling populer yang digunakan saat ini karena sangat kuat dan fleksibel. Algoritme itu sendiri mengikuti proses tiga tahap untuk mengekstraksi tepi dari suatu gambar. Tambahkan ke gambar kabur, langkah preprocessing yang diperlukan untuk mengurangi kebisingan. Secara umum ditunjukkan pada flowchart berikut:



Gambar 5: Flowchart Deteksi Tepi

Dalam prosesnya deteksi tepi dibagi menjadi empat tahap, yang meliputi:

A. Noise Reduction

Piksel gambar mentah sering kali dapat menyebabkan tepian yang bising, jadi sangat penting untuk mengurangi kebisingan sebelum menghitung tepian [13]. Dalam metode canny dalam deteksi tepi, filter Gaussian blur digunakan untuk menghilangkan atau meminimalkan detail yang tidak perlu yang dapat menyebabkan tepian yang tidak diinginkan

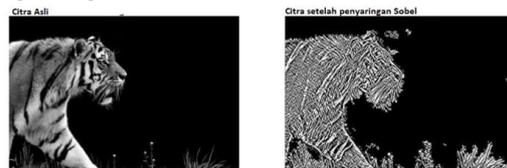


Gambar 6: Perbandingan citra asli dengan citra setelah proses deblur dengan Gaussian blur

B. Perhitungan Intensity Gradient dari Citra

Setelah gambar dihaluskan (kabur), itu difilter dengan kernel Sobel, baik secara horizontal maupun vertical [14]. Hasil dari operasi pemfilteran ini kemudian digunakan untuk menghitung besaran gradien intensitas (G), dan arah (θ) untuk setiap piksel, seperti yang ditunjukkan pada rumus (6)(7).

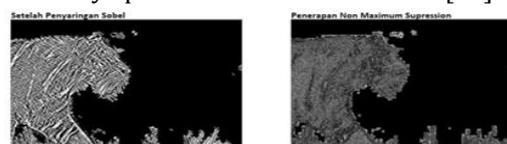
Arah gradien kemudian dibulatkan ke sudut 45 derajat terdekat. Gambar 7. Menunjukkan hasil dari langkah pemrosesan gabungan ini.



Gambar 7: Citra setelah penyaringan sobel

C. Penyaringan Tepi Palsu

Setelah mengurangi noise dan menghitung gradien intensitas, algoritme dalam langkah ini menggunakan teknik yang disebut non-maximum suppression of edge untuk memfilter piksel yang tidak diinginkan. Untuk melakukannya, setiap piksel dibandingkan dengan piksel tetangganya dalam arah gradien positif dan negatif. Jika besarnya gradien dari piksel saat ini lebih besar dari piksel tetangganya, itu dibiarkan tidak berubah. Jika tidak, besarnya piksel saat ini disetel ke nol [15].

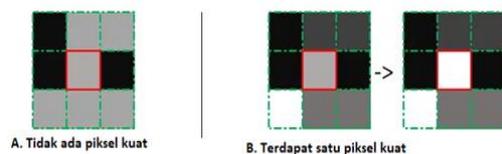


Gambar 8: Penerapan Non Maximum Supression setelah penyaringan Sobel

D. Hysteresis Thresholding

Pada langkah terakhir Deteksi Tepi Canny ini, besaran gradien dibandingkan dengan dua nilai ambang, yang satu lebih kecil dari yang lain.

- Jika nilai besaran gradien lebih tinggi dari nilai ambang batas yang lebih besar, piksel tersebut diasosiasikan dengan tepi padat dan dimasukkan dalam peta tepi akhir.
- Jika nilai besaran gradien lebih rendah dari nilai ambang yang lebih kecil, piksel akan ditekan dan dikecualikan dari peta tepi akhir.
- Semua piksel lainnya, yang besaran gradiennya berada di antara dua ambang ini, ditandai sebagai tepi 'lemah' (yaitu menjadi kandidat untuk disertakan dalam peta tepi akhir).
- Jika piksel 'lemah' terhubung ke piksel yang terkait dengan tepi padat, piksel tersebut juga disertakan dalam peta tepi akhir.



Gambar 9: Histeresis terdiri dari mengubah piksel lemah menjadi piksel kuat

4. HASIL DAN PEMBAHASAN

Pada penerapan metode Canny dalam deteksi tepi dengan menggunakan python dan library opencv dengan listing code sebagai berikut:

```
import cv2
```

Gambar 10: Import library opencv

Sebelum menggunakan pustaka yang ada pada opencv dilakukan import cv2 untuk dapat menggunakan semua pustaka yang ada didalam opencv. Pada penggunaan google colab semua pustaka telah diinstal tetapi jika pada laptop maka perlu diinstall dulu dengan perintah pip install opencv atau conda install opencv

```
# Read the original image
img = cv2.imread('tiger.jpg')

# Display original image
cv2.imshow('Original', img)

cv2.waitKey(0)
```

Gambar 11: Baca data citra dan tampilkan citra aslinya

Untuk pengolahan citra maka hanya 1 frame yang diolah namun untuk pengolahan video maka data video dikonversi menjadi *frame* dan dibaca per *frame*.

```
# Convert to grayscale
img_gray = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)

# Blur the image for better edge
detection

img_blur = cv2.GaussianBlur(img_gray,
(3,3), 0)
```

Gambar 12: Konversi BGR ke Gray

Pembacaan data standar memberikan informasi BGR(*blue, green, red*) yang kemudian dikonversi menjadi *grayscale*. Untuk mendapatkan citra yang akan dideteksi tepinya maka citra di blur menggunakan *Gaussian blur*.

```
# Sobel Edge Detection

# Sobel Edge Detection on the X axis

sobelx = cv2.Sobel(src=img_blur,
ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5)

# Sobel Edge Detection on the Y axis

sobely = cv2.Sobel(src=img_blur,
ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5)

# Combined X and Y Sobel Edge Detection

sobelxy = cv2.Sobel(src=img_blur,
```

```
ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5)  
  
# Display Sobel Edge Detection Images  
cv2.imshow('Sobel X', sobelx)  
  
cv2.waitKey(0)  
  
cv2.imshow('Sobel Y', sobely)  
  
cv2.waitKey(0)  
  
cv2.imshow('Sobel X Y using Sobel() function',  
sobelxy)  
  
cv2.waitKey(0)
```

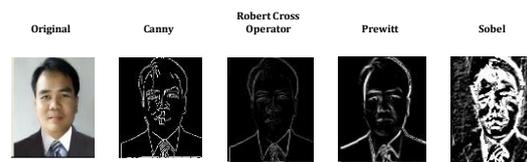
Gambar 13: Deteksi Tepi dengan algoritma Sobel

Deteksi tepi menggunakan metode sobel digunakan untuk mendapatkan semua garis yang ditimbulkan dari citra *gray* yang telah di blur. Hasilnya adalah garis putih yang menunjukkan intensitas yang lebih kuat.

```
# Canny Edge Detection  
  
edges = cv2.Canny(image=img_blur,  
threshold1=100, threshold2=200) # Canny  
Edge Detection  
  
# Display Canny Edge Detection Image  
cv2.imshow('Canny Edge Detection', edges)  
  
cv2.waitKey(0)  
  
cv2.destroyAllWindows()
```

Gambar 14: Deteksi Tepi menggunakan metode canny

Algoritma canny untuk deteksi tepi digunakan untuk mendapatkan garis tepi yang menunjukkan objek, pada pengembangannya objek yang telah dideteksi dapat dikenali secara sederhana dengan menggunakan template yang telah dilatih sebelumnya. Hasil perbandingan metode deteksi tepi dengan metode lainnya seperti LoG (*Laplacian of Gaussian*), *Robert Cross Operator*, *Prewitt* dan *Sobel* ditunjukkan pada gambar 15.



Gambar 15: Perbandingan hasil deteksi tepi

5. KESIMPULAN DAN SARAN

Deteksi tepi merupakan teknik pemrosesan gambar yang penting dan berfokus untuk mengetahui dua algoritme terpentingnya (Sobel Edge Detection dan Canny Edge Detection). Pada penggunaannya di OpenCV, penerapan blur merupakan langkah prepemrosesan yang penting untuk mendapatkan deteksi tepi yang lebih baik. Dari hasil perbandingan deteksi tepi dapat terlihat algoritma canny memiliki performa yang lebih baik dari algoritma deteksi tepi lainnya.

DAFTAR REFERENSI

- [1] Saluky Saluky; Suhono Harso Supangkat; Fetty Fitriyanti Lubis, "Moving Image Interpretation Models to Support City Analysis," *2018 Int. Conf. ICT Smart Soc.*, 2018.
- [2] H. Park, S. Park, and Y. Joo, "Detection of Abandoned and Stolen Objects Based on Dual Background Model and Mask R-CNN," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2990618.
- [3] T. Hafsia;, A. Belhaj;, H. Tlijani;, and K. Nouri, "Implementing Canny Edge Detection Algorithm for Different Blurred and Noisy Images," *2022 IEEE 21st Int. Ccnference Sci. Tech. Autom. Control Comput. Eng.*, 2022, doi: 10.1109/STA56120.2022.10019120.
- [4] S. S. Al-amri, "Image segmentation by using edge detection," *Int. J. Comput. Sci. Eng.*, vol. 02, no. 03, pp. 804–807, 2010.
- [5] A. W. K. Johannes Widagdho Yodha, "PENGENALAN MOTIF BATIK MENGGUNAKAN DETEKSI TEPI CANNY DAN K-NEAREST NEIGHBOR," *Techno.COM*, vol. 13, no. 4, pp. 251–

- 262, 2014.
- [6] M. G. Nayagam and K. Ramar, "A survey on Real time Object Detection and Tracking Algorithms A survey on Real time Object Detection and Tracking Algorithms," *Int. J. Appl. Eng. Res.*, no. April, 2015.
- [7] Z. Chen, M. Li, and H. Yu, "A 3D Circular Object Detection Method Based on Binocular Stereo Vision," *2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics (CISP-BMEI 2017)*, 2017, doi: 10.1109/CISP-BMEI.2017.8301990.
- [8] D. J. Muchammad Al Amin, "KLASIFIKASI KELOMPOK UMUR MANUSIA BERDASARKAN ANALISIS DIMENSI FRAKTAL BOX COUNTING DARI CITRA WAJAH DENGAN DETEKSI TEPI CANNY," *MATHunesa J. Ilm. Mat.*, vol. 2, no. 6, 2017.
- [9] M. Hörhan and H. Eidenberger, "An Efficient DCT template-based Object Detection Method using Phase Correlation," *2016 50th Asilomar Conf. Signals, Syst. Comput.*, pp. 444–448, 2016, doi: 10.1109/ACSSC.2016.7869078.
- [10] P. Jain, N. Delhi, N. Chopra, N. Delhi, V. Gupta, and N. Delhi, "Automatic License Plate Recognition using OpenCV," *Int. J. Comput. Appl. Technol. Res.*, vol. 3, no. 12, pp. 756–761, 2014.
- [11] W. He and K. Yuan, "An Improved Canny Edge Detector and its Realization on FPGA," *Proc. 7th World Congr. Intell. Control Autom.*, pp. 6561–6564, 2008.
- [12] S. Saluky, "Moving Object Detection on CCTV Surveillance Using the Frame Difference Method," *Inf. Technol. Eng. Journals*, vol. 4, pp. 114–122, 2019.
- [13] R. Bentler and L. Chiou, "Digital Noise Reduction: An Overview," *Trends Amplif.*, pp. 67–82, 2006, doi: 10.1177/1084713806289514.
- [14] B. Pan, Z. Lu, and H. Xie, "Mean intensity gradient: An effective global parameter for quality assessment of the speckle patterns used in digital image correlation," *Opt. Lasers Eng.*, vol. 48, no. 4, pp. 469–477, 2010, doi: 10.1016/j.optlaseng.2009.08.010.
- [15] B. Magnier, H. Abdulrahman, and P. Montesinos, "A Review of Supervised Edge Detection Evaluation Methods and an Objective Comparison of Filtering Gradient Computations Using Hysteresis Thresholds," *J. Imaging*, 2018, doi: 10.3390/jimaging4060074.