

## OPTIMASI DETEKSI TEPI PADA CITRA DIGITAL MELALUI TUNING HYPERPARAMETER CLAHE DAN FILTER BILATERAL: STUDI KASUS PADA GAMBAR KENDARAAN

Budi Hartanto <sup>1)</sup>, Bramasto Wiryawan Yudanto <sup>2)</sup>, Didik Nugroho <sup>3)</sup>

<sup>1) 2) 3)</sup> STMIK Sinar Nusantara, <sup>1) 1,3</sup> Informatika, <sup>2)</sup> Sistem Informasi

Email : budihartanto@sinus.ac.id <sup>1)</sup>, bramasto@sinus.ac.id <sup>2)</sup>, didiknugroho@sinus.ac.id <sup>3)</sup>

Diterima : 27 Juni 2024 ; Disetujui : 29 Juli 2024 ; Dipublikasikan : 31 Juli 2024

### ABSTRAK

Penelitian ini bertujuan untuk mengoptimalkan deteksi tepi gambar dengan menggabungkan metode Holistically-Nested Edge Detection (HED) dan filter Sobel-Laplacian. Metode ini diterapkan setelah tahap preprocessing yang mencakup Contrast Limited Adaptive Histogram Equalization (CLAHE) untuk peningkatan kontras dan filter bilateral untuk pengurangan noise. Tuning hyperparameter dilakukan untuk meningkatkan performa deteksi tepi. Evaluasi menunjukkan hasil terbaik dengan Precision sebesar 0.1920, Recall 0.5747, F1 Score 0.2878, Accuracy 0.7231, IoU Score 0.1681, dan ROC AUC Score 0.6569. Temuan ini menunjukkan bahwa metode yang diusulkan dapat meningkatkan akurasi deteksi tepi dengan hasil Recall dan Accuracy yang lebih baik dibandingkan metode konvensional, meskipun Precision dan IoU masih menunjukkan potensi untuk perbaikan lebih lanjut. Penelitian ini memberikan wawasan tentang pengaruh preprocessing dan tuning hyperparameter terhadap hasil deteksi tepi, serta aplikasinya dalam berbagai bidang pengolahan citra.

**Kata Kunci** : Deteksi Tepi, Holistically-Nested Edge Detection, Hyperparameter Tuning, CLAHE

### ABSTRACT

*This study aims to optimize edge detection in images by combining Holistically-Nested Edge Detection (HED) with Sobel-Laplacian filters. The approach includes preprocessing steps such as Contrast Limited Adaptive Histogram Equalization (CLAHE) for contrast enhancement and bilateral filtering for noise reduction. Hyperparameter tuning is performed to enhance edge detection performance. The evaluation reveals the following best metrics: Precision of 0.1920, Recall of 0.5747, F1 Score of 0.2878, Accuracy of 0.7231, IoU Score of 0.1681, and ROC AUC Score of 0.6569. The results demonstrate that the proposed method improves edge detection accuracy with better Recall and Accuracy compared to conventional methods, although Precision and IoU still indicate potential for further improvement. This research provides insights into the impact of preprocessing and hyperparameter tuning on edge detection outcomes and its applications in various image processing fields.*

**Keywords** : Edge Detection, Holistically-Nested Edge Detection, Hyperparameter Tuning, CLAHE

## 1. PENDAHULUAN

Deteksi tepi merupakan salah satu teknik fundamental dalam pengolahan citra yang bertujuan untuk mengidentifikasi batas objek dan fitur penting dalam gambar. Teknik ini memainkan peran krusial dalam berbagai aplikasi seperti pengenalan pola, segmentasi gambar, dan analisis citra medis [1]. Deteksi tepi yang akurat dapat meningkatkan pemahaman dan analisis terhadap struktur dan bentuk objek dalam citra, yang penting untuk aplikasi seperti pengawasan, kendaraan otonom, dan diagnosis medis. Metode deteksi tepi klasik seperti Canny, Sobel, dan Laplacian of Gaussian (LoG) telah digunakan secara luas dan terbukti efektif dalam banyak kasus. Canny Edge Detector [2], dikenal karena kemampuannya dalam mendeteksi tepi dengan akurasi tinggi melalui penggunaan filter ganda dan pengurangan noise. Sobel Operator, di sisi lain, memberikan estimasi gradien tepi yang sederhana tetapi efektif [3]. Meskipun metode ini telah banyak digunakan, mereka sering kali mengalami kesulitan dalam mengatasi noise dan perubahan kontras dalam gambar.

Metode deteksi tepi tradisional memiliki keterbatasan dalam menangani tantangan seperti noise tinggi, perubahan pencahayaan, dan variasi kontras yang besar. Penelitian [4] menunjukkan bahwa metode deteksi tepi klasik mungkin tidak selalu memadai untuk gambar dengan noise tinggi dan kontras yang bervariasi, sehingga diperlukan pendekatan yang lebih canggih. Hal ini menekankan pentingnya pengembangan dan penerapan metode deteksi tepi yang lebih modern dan efektif.

Dengan kemajuan teknologi pengolahan citra dan pembelajaran mesin, metode deteksi tepi modern telah diperkenalkan untuk mengatasi keterbatasan metode konvensional. Teknik seperti Structured Edge Detection dan metode berbasis deep learning telah menunjukkan peningkatan signifikan dalam akurasi deteksi tepi [5]. Structured Edge Detection memanfaatkan model pembelajaran mesin untuk mendeteksi tepi dengan lebih akurat dan robust dibandingkan metode deteksi tepi konvensional.

Preprocessing, seperti peningkatan kontras dan pengurangan noise, sangat penting dalam mempengaruhi kualitas deteksi tepi. Teknik seperti Contrast Limited Adaptive Histogram

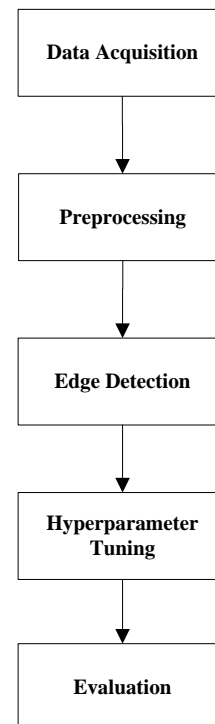
Equalization (CLAHE) digunakan untuk meningkatkan kontras lokal dan memperjelas tepi [6]. Selain itu, tuning hyperparameter adalah teknik yang krusial untuk mengoptimalkan performa model deteksi tepi. Penelitian lain menunjukkan bahwa tuning parameter dapat meningkatkan akurasi deteksi secara signifikan dengan mengoptimalkan parameter yang mempengaruhi preprocessing dan deteksi tepi [7].

Penelitian mempunyai tujuan untuk dapat mengoptimalkan model deteksi tepi dengan menggabungkan Holistically-Nested Edge Detection (HED) dan filter Sobel -Laplacian. Studi ini juga mencakup preprocessing gambar dan tuning hyperparameter untuk meningkatkan performa deteksi tepi. Dengan mengevaluasi metrik kinerja seperti Precision, Recall, F1 Score, dan ROC AUC Score, penelitian ini diharapkan dapat memberikan wawasan tentang metode yang lebih efektif dalam deteksi tepi

## 2. METODE

### 2.1 Tahapan Penelitian

Tahapan dalam proses penelitian ini dilakukan dengan melalui beberapa aktivitas. Adapun tahapan penelitian meliputi tahapan-tahapan yang dapat dilihat pada gambar 1 sebagai berikut ini.



**Gambar 1. Tahapan Penelitian**

Tahapan penelitian di atas dapat dijelaskan sebagai berikut:

#### 1. Data Acquisition (Akuisisi Data)

Pada tahap ini, data gambar yang akan digunakan dalam penelitian dikumpulkan. Proses akuisisi data melibatkan pengambilan atau pengunduhan gambar dari berbagai sumber. Gambar-gambar ini dapat berasal dari database publik, pengambilan gambar secara langsung, atau data satelit. Data yang diambil harus mencakup berbagai kondisi seperti variasi pencahayaan, jenis objek, dan kondisi lingkungan untuk memastikan model yang dikembangkan mampu mengatasi situasi yang berbeda. Dalam konteks deteksi kerusakan jalan, misalnya, gambar yang dikumpulkan mungkin terdiri dari jalan yang memiliki retakan, lubang, atau kerusakan lainnya [8].

#### 2. Preprocessing

Prapemrosesan bertujuan untuk menyiapkan gambar agar lebih mudah dianalisis oleh algoritma deteksi tepi. Langkah ini mencakup peningkatan kontras, pengurangan noise, dan penajaman gambar. Teknik seperti Contrast Limited Adaptive Histogram Equalization (CLAHE) digunakan untuk meningkatkan kontras lokal gambar, yang membantu menonjolkan fitur penting [9]. Selain itu, filter bilateral digunakan untuk mengurangi noise sambil mempertahankan tepi yang penting [10].

#### 3. Edge Detection

Deteksi tepi adalah tahap inti dari penelitian ini, di mana metode Holistically-Nested Edge Detection (HED) dan kombinasi filter Sobel-Laplacian diterapkan untuk mengidentifikasi batas objek dalam gambar. HED merupakan pendekatan deep learning yang dapat mendeteksi tepi dengan presisi tinggi, mengintegrasikan informasi tepi dari berbagai skala [11]. Filter Sobel dan Laplacian digunakan untuk mendeteksi perubahan intensitas dan gradien dalam gambar, memperkuat hasil dari model HED.

#### 4. Hyperparameter Tuning

Hyperparameter tuning adalah langkah penting dalam meningkatkan kinerja model deteksi tepi. Proses ini melibatkan penyesuaian parameter model seperti ukuran kernel, nilai threshold, dan lainnya untuk memaksimalkan performa model. Teknik yang digunakan bisa

berupa grid search atau random search, di mana berbagai kombinasi parameter dieksplorasi untuk menemukan konfigurasi terbaik [12].

#### 5. Evaluation

Tahap evaluasi adalah langkah terakhir di mana kinerja model deteksi tepi dinilai menggunakan berbagai metrik untuk mengukur akurasi dan efektivitasnya. Metrik seperti precision, recall, F1 score, accuracy, IoU score, dan ROC AUC score digunakan untuk memberikan gambaran lengkap tentang performa model [13]. Evaluasi dilakukan untuk memastikan bahwa model yang dikembangkan dapat diandalkan dalam aplikasi dunia nyata dan memberikan hasil yang konsisten di berbagai kondisi [14].

## 3. HASIL DAN PEMBAHASAN

### 3.1 Perancangan Model

Model ini dirancang untuk mendeteksi tepi dalam gambar dengan menggunakan berbagai teknik pemrosesan gambar dan model deteksi tepi berbasis Structured Edge Detection (HED). Proses ini melibatkan pengunduhan gambar dan model, pra-pemrosesan gambar, deteksi tepi, evaluasi kinerja, serta tuning hyperparameter untuk meningkatkan hasil deteksi tepi.

#### 1. Pengunduhan dan Persiapan Data

Kode di bawah menunjukkan dua URL penting dalam proses pemrosesan gambar dan deteksi tepi. `image_url` adalah tautan ke gambar yang akan diproses, yang diunduh dari situs Pexels, dan disimpan dengan nama `car.jpg`. Sedangkan `model_url` adalah tautan ke file model deteksi tepi berbasis Structured Edge Detection (HED) yang diperlukan untuk mendeteksi tepi dalam gambar. Model ini diunduh dari repositori GitHub OpenCV dan disimpan dengan nama `model.yml.gz`. Kedua sumber ini merupakan input penting dalam pipeline pemrosesan gambar, di mana gambar digunakan untuk mendeteksi tepi, sementara model digunakan untuk menghasilkan peta tepi yang mendetail.

```
# URL gambar
image_url = "https://images.pexels.com/photos/3817871/pexels-photo-3817871.jpeg"

# URL model
model_url = "https://github.com/opencv/opencv_extra/raw/master/testdata/cv/ximgproc/model.yml.gz"
model_filename = "model.yml.gz"
```

**Gambar 2. Pengunduhan dan Persiapan**

## 2. Fungsi Kontras dengan CLAHE

Fungsi `apply_clahe` di atas digunakan untuk menerapkan teknik peningkatan kontras pada gambar menggunakan metode CLAHE (Contrast Limited Adaptive Histogram Equalization).

```
# Fungsi untuk peningkatan kontras dengan CLAHE
def apply_clahe(image, clipLimit=2.0, tileGridSize=(8, 8)):
    lab = cv2.cvtColor(image, cv2.COLOR_RGB2LAB)
    l, a, b = cv2.split(lab)
    clahe = cv2.createCLAHE(clipLimit=clipLimit, tileGridSize=tileGridSize)
    cl = clahe.apply(l)
    limg = cv2.merge((cl, a, b))
    return cv2.cvtColor(limg, cv2.COLOR_LAB2RGB)
```

**Gambar 3. Fungsi Kontras**

Proses ini dilakukan dengan langkah-langkah berikut:

- a. Konversi Warna: Gambar input dikonversi dari format RGB (Red, Green, Blue) ke format LAB (Lightness, A, B) menggunakan `cv2.cvtColor`. Format LAB memisahkan komponen kecerahan (L) dari komponen warna (A dan B), yang memudahkan pengolahan kontras.
- b. Pemisahan Komponen: Komponen kecerahan (L) dipisahkan dari komponen warna (A dan B) menggunakan `cv2.split`. Ini memungkinkan aplikasi CLAHE hanya pada komponen kecerahan tanpa mengubah informasi warna.
- c. Penerapan CLAHE: Objek CLAHE dibuat dengan parameter `clipLimit` dan `tileGridSize` yang diatur untuk mengontrol batas kontras dan ukuran grid lokal. Metode `clahe.apply` diterapkan pada komponen kecerahan (L) untuk memperbaiki kontras gambar.
- d. Penggabungan Komponen: Setelah peningkatan kontras, komponen kecerahan yang telah diproses digabungkan kembali dengan komponen warna (A dan B) menggunakan `cv2.merge`.
- e. Konversi Kembali: Gambar yang telah diproses dalam format LAB dikonversi kembali ke format RGB dengan `cv2.cvtColor` untuk mendapatkan gambar akhir dengan kontras yang diperbaiki dalam

format yang bisa ditampilkan atau digunakan lebih lanjut.

## 3. Fungsi Penajaman Gambar

Fungsi `sharpen_image` meningkatkan ketajaman gambar dengan menggunakan kernel tertentu. Jika kernel tidak disediakan, fungsi ini menggunakan kernel penajaman 3x3 yang default, yang menonjolkan tepi dengan mengurangi nilai dari piksel sekeliling dan menambahkannya pada piksel pusat. Fungsi ini memanfaatkan `cv2.filter2D` dari OpenCV untuk menerapkan kernel ini, sehingga menghasilkan gambar yang lebih tajam dengan tepi dan detail yang lebih jelas.

```
# Fungsi untuk pengurangan noise dengan Filter Bilateral
def apply_bilateral_filter(image, d=9, sigmaColor=75, sigmaSpace=75):
    return cv2.bilateralFilter(image, d=d, sigmaColor=sigmaColor, sigmaSpace=sigmaSpace)

# Fungsi untuk penajaman gambar dengan kernel tertentu
def sharpen_image(image, kernel=None):
    if kernel is None:
        kernel = np.array([[[-1, -1, -1], [-1, 9, -1], [-1, -1, -1]], dtype=np.float32)
    return cv2.filter2D(image, -1, kernel)

# Fungsi untuk binarization adaptif
def adaptive_threshold(image):
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    return cv2.adaptiveThreshold(gray, 1, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                cv2.THRESH_BINARY, 11, 2)
```

**Gambar 4. Fungsi Penajaman Gambar**

## 4. Menerapkan Threshold Terbaik

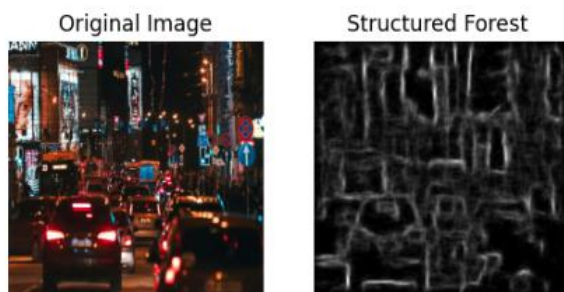
- a. Thresholding: `best_threshold = 0.1` menetapkan nilai threshold tetap sebesar 0.1 untuk binarisasi. Fungsi `cv2.threshold` mengubah gambar `edges_structured` menjadi gambar biner, di mana nilai piksel yang melebihi threshold diatur menjadi 1 (putih) dan yang di bawah threshold diatur menjadi 0 (hitam). Gambar biner ini disimpan dalam `edges_binary_best`.
- b. Konversi Tipe Data: `edges_binary_best = np.uint8(edges_binary_best)` memastikan bahwa data gambar biner berada dalam format `uint8`, yang merupakan tipe data standar untuk pemrosesan gambar di OpenCV.
- c. Operasi Morfologi: Gambar biner menjalani dua operasi morfologi menggunakan fungsi `cv2.morphologyEx` dengan kernel yang telah ditentukan:

### 3.2 Hasil dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap model yang telah dikembangkan hasil dari pemodelan tersebut adalah sebagai Berikut.

#### 1. Perbandingan Gambar Asli dengan Hasil

Pada hasil deteksi tepi, "Original Image" adalah gambar asli yang digunakan sebagai dasar untuk proses pemrosesan dan analisis. Gambar ini belum mengalami perubahan atau pemrosesan dan memberikan konteks visual yang diperlukan untuk membandingkan hasil deteksi tepi. Sementara itu, "Structured Forest" mengacu pada metode deteksi tepi yang menggunakan teknik pembelajaran mesin canggih untuk mendeteksi tepi dalam gambar. Dengan menggunakan model Structured Forest, gambar hasil deteksi menampilkan tepi yang telah diidentifikasi oleh model tersebut. Hasilnya adalah gambar yang menunjukkan area-area di mana tepi terdeteksi dengan intensitas atau warna tertentu. Dengan demikian, perbandingan antara gambar asli dan hasil deteksi tepi dari Structured Forest memungkinkan evaluasi efektivitas metode deteksi tepi dalam mengenali batas-batas objek dalam gambar

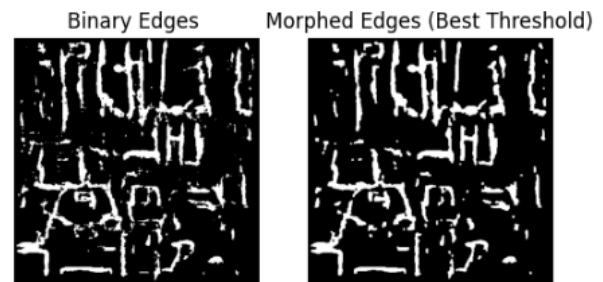


**Gambar 5. Perbandingan Original Image**

#### 2. Binary Edges dan Morphped Edges

Dalam studi kasus deteksi tepi untuk analisis gambar, "Binary Edges" dan "Morphped Edges (Best Threshold)" memiliki peran krusial dalam evaluasi dan peningkatan hasil deteksi tepi. "Binary Edges" menggambarkan hasil awal dari deteksi tepi yang diubah menjadi format biner, memungkinkan visualisasi yang jelas dari tepi yang terdeteksi tanpa gangguan noise. Gambar ini penting untuk memahami area yang dianggap sebagai tepi oleh model deteksi. Sementara itu, "Morphped Edges (Best Threshold)" menunjukkan hasil akhir setelah penerapan teknik morfologi untuk memperbaiki

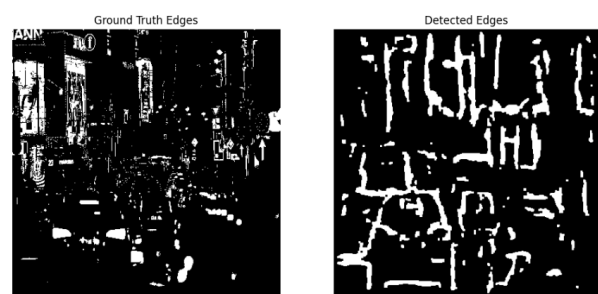
dan membersihkan hasil deteksi tepi. Dengan menghilangkan celah dan mengurangi noise, gambar ini memberikan representasi tepi yang lebih halus dan akurat, yang sangat penting dalam studi kasus yang melibatkan analisis struktur atau fitur spesifik dari gambar. Kedua gambar ini bersama-sama membantu dalam menilai efektivitas metode deteksi tepi dan melakukan penyesuaian yang diperlukan untuk mencapai hasil yang optimal dalam konteks aplikasi spesifik.



**Gambar 6. Binary dan Morphped Edges**

#### 3. Ground Truth Edges dan Detected Edges

Ground Truth Edges adalah gambar referensi yang menggambarkan tepi sebenarnya dari objek atau fitur dalam gambar asli, berdasarkan data yang telah diklasifikasikan secara manual atau oleh sistem yang sangat terpercaya. Ground Truth berfungsi sebagai standar yang harus dicapai atau didekati oleh sistem deteksi tepi. Dalam konteks studi kasus ini, gambar "Ground Truth Edges" menunjukkan batas-batas tepi yang benar yang seharusnya diidentifikasi oleh model deteksi tepi. Detected Edges merupakan hasil deteksi tepi yang diperoleh dari model deteksi tepi yang sedang dievaluasi. Gambar ini menunjukkan tepi-tepi yang diidentifikasi oleh model berdasarkan pemrosesan dan parameter yang diterapkan. Dalam studi kasus ini, "Detected Edges" mencerminkan kemampuan model untuk mengenali dan menandai tepi sesuai dengan data yang terdapat dalam gambar.



**Gambar 7. Ground Truth Edges dan Detected Edges**

4. Hasil Evaluasi Model

a. Recall (0.5747):

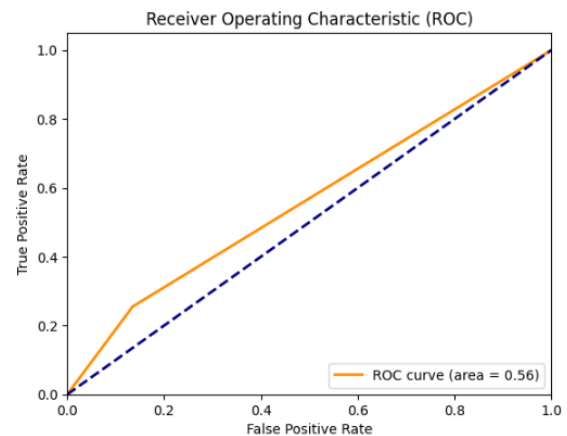
- Recall mengukur seberapa baik model dalam mendeteksi semua tepi yang sebenarnya ada. Secara spesifik, recall adalah rasio dari jumlah tepi yang terdeteksi dengan benar (True Positives) terhadap jumlah total tepi yang seharusnya terdeteksi (jumlah True Positives + False Negatives).
- Implikasi, Nilai recall sebesar 0.5747 menunjukkan bahwa model dapat mendeteksi sekitar 57.47% dari tepi yang sebenarnya ada dalam gambar. Meskipun ini menunjukkan bahwa model cukup baik dalam menemukan sebagian besar tepi yang ada, masih ada ruang untuk perbaikan dalam menangkap semua tepi yang relevan. Ini penting karena dalam banyak aplikasi deteksi tepi, menangkap sebanyak mungkin tepi adalah kunci.

b. Accuracy (0.7231):

- Pengertian: Accuracy mengukur seberapa sering model benar dalam klasifikasi, yaitu jumlah total prediksi yang benar (baik tepi yang terdeteksi dengan benar maupun non-tepi yang diklasifikasikan dengan benar) dibagi dengan jumlah total prediksi. Accuracy dihitung dengan formula:  $(\text{True Positives} + \text{True Negatives}) / \text{Total Predictions}$ .
- Implikasi: Nilai accuracy sebesar 0.7231 menunjukkan bahwa model secara keseluruhan benar dalam sekitar 72.31% dari semua prediksi yang dibuat. Meskipun accuracy memberikan gambaran umum tentang kinerja model, ia mungkin tidak selalu mencerminkan kualitas deteksi tepi jika dataset tidak seimbang atau jika beberapa kelas lebih penting daripada yang lain.

c. ROC AUC Score (0.6569):

- ROC AUC (Receiver Operating Characteristic - Area Under Curve) Score mengukur kemampuan model dalam membedakan antara kelas positif dan negatif pada berbagai threshold. Nilai AUC yang mendekati 1 menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan antara kelas, sementara nilai yang mendekati 0.5 menunjukkan performa yang kurang baik, mirip dengan tebakan acak.
- Implikasi, ROC AUC Score sebesar 0.6569 menunjukkan bahwa model memiliki kemampuan diskriminasi yang cukup baik, tetapi masih ada ruang untuk perbaikan. Dengan skor ini, model menunjukkan kemampuan yang relatif baik dalam membedakan antara tepi dan non-tepi, tetapi belum optimal. Dalam konteks deteksi tepi, AUC yang lebih tinggi diinginkan untuk memastikan bahwa model dapat dengan efektif memisahkan tepi dari latar belakang.



**Gambar 7. Curva Receiver Operating Characteristic**

**4. PENUTUP**

**4.1. Kesimpulan**

Penelitian ini berfokus pada evaluasi dan perbaikan metode deteksi tepi gambar menggunakan model Structured Forest dan berbagai teknik pemrosesan gambar. Berdasarkan hasil evaluasi, model deteksi tepi menunjukkan kinerja yang cukup baik dengan

nilai recall sebesar 0.5747, accuracy sebesar 0.7231, dan ROC AUC Score sebesar 0.6569. Nilai recall yang mencapai 57.47% menunjukkan bahwa model mampu mendeteksi lebih dari setengah dari tepi yang sebenarnya ada dalam gambar, meskipun ada ruang untuk peningkatan agar model dapat mendeteksi lebih banyak tepi yang relevan. Accuracy sebesar 72.31% menunjukkan bahwa model secara keseluruhan akurat dalam klasifikasi, tetapi mungkin tidak cukup baik jika dataset tidak seimbang atau ada variasi signifikan dalam data. ROC AUC Score sebesar 0.6569 mengindikasikan bahwa model memiliki kemampuan diskriminasi yang cukup baik dalam membedakan antara tepi dan non-tepi, namun masih perlu ditingkatkan untuk mencapai performa optimal.

Secara keseluruhan, model Structured Forest yang diterapkan dalam penelitian ini menunjukkan potensi yang baik dalam deteksi tepi, tetapi hasil evaluasi menggarisbawahi kebutuhan untuk peningkatan lebih lanjut. Perbaikan dapat dilakukan dengan mengoptimalkan parameter model, meningkatkan preprocessing gambar, dan memperbaiki metode evaluasi untuk mencapai hasil yang lebih memuaskan. Penelitian ini juga menekankan pentingnya metode deteksi tepi yang lebih canggih dan adaptif untuk mengatasi tantangan seperti noise tinggi, perubahan pencahayaan, dan variasi kontras yang besar. Temuan ini memberikan dasar yang kuat untuk pengembangan dan penerapan teknik deteksi tepi yang lebih efektif di masa depan.

#### 4.2. Saran

Dalam penelitian ini, beberapa langkah tambahan dapat diambil untuk meningkatkan hasil deteksi tepi. Pertama, peningkatan preprocessing gambar seperti penggunaan teknik noise reduction yang lebih canggih dan peningkatan kontras yang lebih spesifik dapat memperbaiki kualitas deteksi. Selain itu, optimasi parameter model Structured Forest melalui teknik tuning hyperparameter yang lebih mendalam bisa membantu mencapai hasil yang lebih baik. Pengujian model dengan berbagai dataset gambar akan membantu mengevaluasi performa dalam berbagai kondisi, sedangkan eksperimen dengan metode deteksi tepi lainnya, termasuk pendekatan deep learning, dapat memberikan wawasan lebih

lanjut tentang efektivitas model. Evaluasi tambahan menggunakan metrik lain seperti precision-recall curve juga penting untuk mendapatkan gambaran komprehensif tentang performa model. Terakhir, integrasi model dengan sistem dunia nyata dan eksplorasi teknik peningkatan kinerja lebih lanjut seperti augmentasi data dan metode ensemble dapat memperkuat aplikasi praktis dan efektivitas deteksi tepi dalam situasi yang lebih kompleks.

#### 5. DAFTAR PUSTAKA

- [1] R. C. Gonzalez, *Digital image processing*. Pearson education india, 2009.
- [2] J. Canny, "A computational approach to edge detection," *IEEE Trans Pattern Anal Mach Intell*, no. 6, pp. 679–698, 1986.
- [3] I. Sobel and G. Feldman, "An isotropic 3x3 image gradient operator for image processing," *Mach. Vis. Three-Dimens. Scenes*, no. June, pp. 376–379, 1968.
- [4] E. Türetken, G. González, C. Blum, and P. Fua, "Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors," *Neuroinformatics*, vol. 9, pp. 279–302, 2011.
- [5] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans Pattern Anal Mach Intell*, vol. 37, no. 8, pp. 1558–1570, 2014.
- [6] V. Stimper, S. Bauer, R. Ernstorfer, B. Schölkopf, and R. P. Xian, "Multidimensional contrast limited adaptive histogram equalization," *IEEE Access*, vol. 7, pp. 165437–165447, 2019.
- [7] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," *Adv Neural Inf Process Syst*, vol. 24, 2011.
- [8] J. Jing, S. Liu, G. Wang, W. Zhang, and C. Sun, "Recent advances on image edge detection: A comprehensive review," *Neurocomputing*, vol. 503, pp. 259–271, 2022.
- [9] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*, 1994, pp. 474–485.

- [10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, IEEE, 1998, pp. 839–846.
- [11] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.
- [12] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Adv Neural Inf Process Syst*, vol. 24, 2011.
- [13] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, 2009.
- [14] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, "What is a good evaluation measure for semantic segmentation?," in *Bmvc*, Bristol, 2013, pp. 10–5244.