

Peningkatan Kualitas Citra Tradisional vs Deep Learning Pada Citra Objek Rumah

Pengolahan Citra Digital (PCD) sangat bergantung pada kualitas citra input. Proyek ini mengeksplorasi metode peningkatan citra untuk mengatasi degradasi dan dampaknya pada segmentasi objek.



Anggota Tim



Boyke Ramadha



Rahmat Hidayat



Muhammad Hafiz

Tantangan Kualitas Citra

Citra seringkali mengalami penurunan kualitas karena berbagai faktor seperti pencahayaan buruk, sensor kamera, gerakan, dan noise. Noise membuat citra berbintik, detail kabur, dan batas objek tidak jelas.



Penurunan kualitas ini berdampak negatif pada tahapan PCD selanjutnya seperti segmentasi dan deteksi fitur. Solusinya adalah **Image Enhancement** untuk memperbaiki tampilan visual.





Pendekatan Peningkatan Citra

1

Metode Tradisional

Menggunakan teknik filter dan manipulasi kontras (Non-Local Denoising + CLAHE). Implementasi sederhana, cepat, tanpa pelatihan model.

2

Metode Berbasis Pembelajaran

Mempelajari pola kompleks dari data. Menggunakan **patch extraction** dan **supervised regression** untuk hasil adaptif.

Tujuan & Manfaat Proyek

Tujuan

- Implementasi penambahan noise & enhancement (tradisional & learning-based).
- Perbandingan visual & kuantitatif kedua metode.
- Uji dampak enhancement pada segmentasi objek (SLIC Superpixel).
- Ukur kualitas segmentasi dengan Dice Score & IoU.

Manfaat

- Pemahaman penerapan enhancement citra.
- Referensi perbandingan metode tradisional vs learning-based.
- Mengetahui hubungan kualitas enhancement & segmentasi.
- Dasar pengembangan metode deep learning lebih lanjut.



LIBRARY

//SET UP ENVIRONMENT

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

from skimage import restoration, filters, segmentation, metrics, morphology, color
from skimage.util import random_noise
from google.colab import files
%matplotlib inline
```

Implementasi Kode

Pemuatan Data

Memuat dataset citra objek rumah beserta anotasi, menyiapkan untuk tahap pelatihan dan validasi model.

```
uploaded = files.upload()

# Ambil satu file gambar
image_path = list(uploaded.keys())[0]

# Load image (OpenCV default BGR + convert ke RGB)
img = cv2.imread(image_path)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(6,6))
plt.imshow(img_rgb)
plt.title("Citra Rumah Asli (RGB)")
plt.axis("off")
```

Metode Peningkatan Tradisional

Mengimplementasikan algoritma peningkatan citra tradisional seperti CLAHE dan denoising mean

```
DENOISING TRADITIONAL + CLAHE
denoise_trad = cv2.bilateralFilter(noisy_img, 9, 75, 75)

# CLAHE RGB
def clahe_rgb(img):
    clahe = cv2.createCLAHE(2.0, (8,8))
    channels = cv2.split(img)
    out = [clahe.apply(c) for c in channels]
    return cv2.merge(out)

enh_trad = clahe_rgb(denoise_trad)

plt.figure(figsize=(5,5))
plt.imshow(enh_trad)
plt.title("Enhancement Konvensional (Bilateral + CLAHE)")
plt.axis("off")
```

Penambahan Noise

Menerapkan Noise pada gambar

```
# PENAMBAHAN GAUSSIAN NOISE

noisy_img = random_noise(img_rgb, mode='gaussian', var=0.01)
noisy_img = (noisy_img * 255).astype(np.uint8)

plt.figure(figsize=(5,5))
plt.imshow(noisy_img)
plt.title("Citra Rumah Terdegradasi (Noise RGB)")
plt.axis("off")
```

Persiapan Data Latih

Menyiapkan data latih model Deep Learning

```
# PERSIAPAN DATA LATIH
# Parameter patch
PATCH_SIZE = 7
STRIDE = 3

def extract_patches(img, patch_size, stride):
    patches = []
    h, w, c = img.shape
    for y in range(0, h - patch_size, stride):
        for x in range(0, w - patch_size, stride):
            patch = img[y:y+patch_size, x:x+patch_size, :]
            patches.append(patch.flatten())
    return np.array(patches)

# Input training: noisy image
X_train = extract_patches(noisy_img, PATCH_SIZE, STRIDE)

# Target training: enhanced reference (CLAHE output sebagai proxy GT)
Y_train = extract_patches(enh_trad, PATCH_SIZE, STRIDE)

print("Training samples:", X_train.shape)

Training samples: (144875, 147)
```

Implementasi Kode

Pelatihan Model

Melatih model Deep learning untuk memaksimalkan peningkatan citra

```
PELATIHAN MODEL
# Tambahkan bias
X_aug = np.hstack([X_train, np.ones((X_train.shape[0], 1))])

# Least Squares Training
W = np.linalg.lstsq(X_aug, Y_train, rcond=None)[0]

print("Model trained: Weight matrix shape =", W.shape)

Model trained: Weight matrix shape = (148, 147)
```

Inferensi Citra

Menerapkan peningkatan kualitas berdasarkan data latih dan memaksimalkan denoising + kualitas gambar

```
INFERENSI ENHANCEMENT
def apply_learned_enhancement(img, W, patch_size):
    h, w, c = img.shape
    out = img.copy().astype(np.float32)
    count = np.zeros_like(out)

    for y in range(0, h - patch_size, 1):
        for x in range(0, w - patch_size, 1):
            patch = img[y:y+patch_size, x:x+patch_size, :].flatten()
            patch_aug = np.append(patch, 1)
            pred = patch_aug @ W
            pred = pred.reshape((patch_size, patch_size, 3))

            out[y:y+patch_size, x:x+patch_size, :] += pred
            count[y:y+patch_size, x:x+patch_size, :] += 1

    return np.clip(out / (count + 1e-8), 0, 255).astype(np.uint8)

enh_dl = apply_learned_enhancement(noisy_img, W, PATCH_SIZE)

plt.figure(figsize=(5,5))
plt.imshow(enh_dl)
plt.title("Learned RGB Enhancement (Trained Model)")
plt.axis("off")
```

Segmentasi Citra

Menerapkan segmentasi SLIC Superpixel pada citra yang ditingkatkan

```
SEGMENTASI OBJEK
# Gunakan slic superpixel
segments_trad = segmentation.slic(
    enh_traditional, n_segments=300, compactness=10, start_label=1
)

segments_dl = segmentation.slic(
    enh_dl, n_segments=300, compactness=10, start_label=1
)

fig, ax = plt.subplots(1,2, figsize=(12,6))

ax[0].imshow(segmentation.mark_boundaries(enh_traditional, segments_trad))
ax[0].set_title("Segmentasi (Tradisional)")
ax[0].axis("off")

ax[1].imshow(segmentation.mark_boundaries(enh_dl, segments_dl))
ax[1].set_title("Segmentasi (Learning-based)")
ax[1].axis("off")

# Segmentasi sederhana sebagai ground truth pendekatan
gt = cv2.inRange(img_rgb, (100, 100, 100), (255, 255, 255))
gt = morphology.remove_small_objects(gt.astype(bool), min_size=500)
gt = gt.astype(np.uint8)
```

Evaluasi Model

Mengevaluasi hasilnya menggunakan Dice Score serta IoU.

```
EVALUASI KUALITAS GAMBAR
def evaluate_segmentation(pred, gt):
    """
    pred, gt : binary mask (True/False atau 0/1)
    """
    pred = pred.astype(bool)
    gt = gt.astype(bool)

    intersection = np.logical_and(pred, gt).sum()
    union = np.logical_or(pred, gt).sum()

    dice = (2 * intersection) / (pred.sum() + gt.sum() + 1e-8)
    iou = intersection / (union + 1e-8)

    return dice, iou

# Ambil satu region dominan sebagai objek (pendekatan studi kasus)
seg_trad_binary = segments_trad == np.bincount(segments_trad.flatten()).argmax()
seg_dl_binary = segments_dl == np.bincount(segments_dl.flatten()).argmax()

dice_trad, iou_trad = evaluate_segmentation(seg_trad_binary, gt)
dice_dl, iou_dl = evaluate_segmentation(seg_dl_binary, gt)

print("== Evaluasi Segmentasi ==")
print(f"Tradisional + Dice: {dice_trad:.3f}, IoU: {iou_trad:.3f}")
print(f"Learning-based + Dice: {dice_dl:.3f}, IoU: {iou_dl:.3f}")

== Evaluasi Segmentasi ==
Tradisional + Dice: 0.052, IoU: 0.027
Learning-based + Dice: 0.129, IoU: 0.069
```

```
# PERBANDINGAN TRADITIONAL VS DEEP LEARNING
```

```
plt.figure(figsize=(16,4))

plt.subplot(1,4,1)
plt.imshow(img_rgb)
plt.title("Citra Asli")
plt.axis("off")

plt.subplot(1,4,2)
plt.imshow(noisy_img)
plt.title("Noisy Input")
plt.axis("off")

plt.subplot(1,4,3)
plt.imshow(enh_trad)
plt.title("Konvensional")
plt.axis("off")

plt.subplot(1,4,4)
plt.imshow(enh_dl)
plt.title("Deep Learning Enhancement")
plt.axis("off")

plt.tight_layout()
plt.show()
```

IMPLEMENTASI PERBANDINGAN HASIL

Akuisisi Data & Metodologi

Akuisisi Data

Data diambil menggunakan kamera HP dengan variasi jarak, sudut, dan posisi objek. Objek diletakkan di atas alas biru untuk memudahkan segmentasi.



Alur Sistem

Input citra RGB, preprocessing, penambahan Gaussian noise, enhancement (tradisional/learning-based), segmentasi SLIC, evaluasi Dice & IoU, analisis perbandingan.



Landasan Teori: Noise & Enhancement

Pengolahan Citra Digital

Memanipulasi dan menganalisis citra digital untuk meningkatkan kualitas atau mengekstrak informasi.

Noise Gaussian

Gangguan acak yang membuat citra "berpasir", disimulasikan untuk menguji performa enhancement.

Image Enhancement

Proses peningkatan kualitas citra untuk mengurangi noise, memperbaiki kontras, dan memperjelas detail.



Metode Enhancement Mendalam

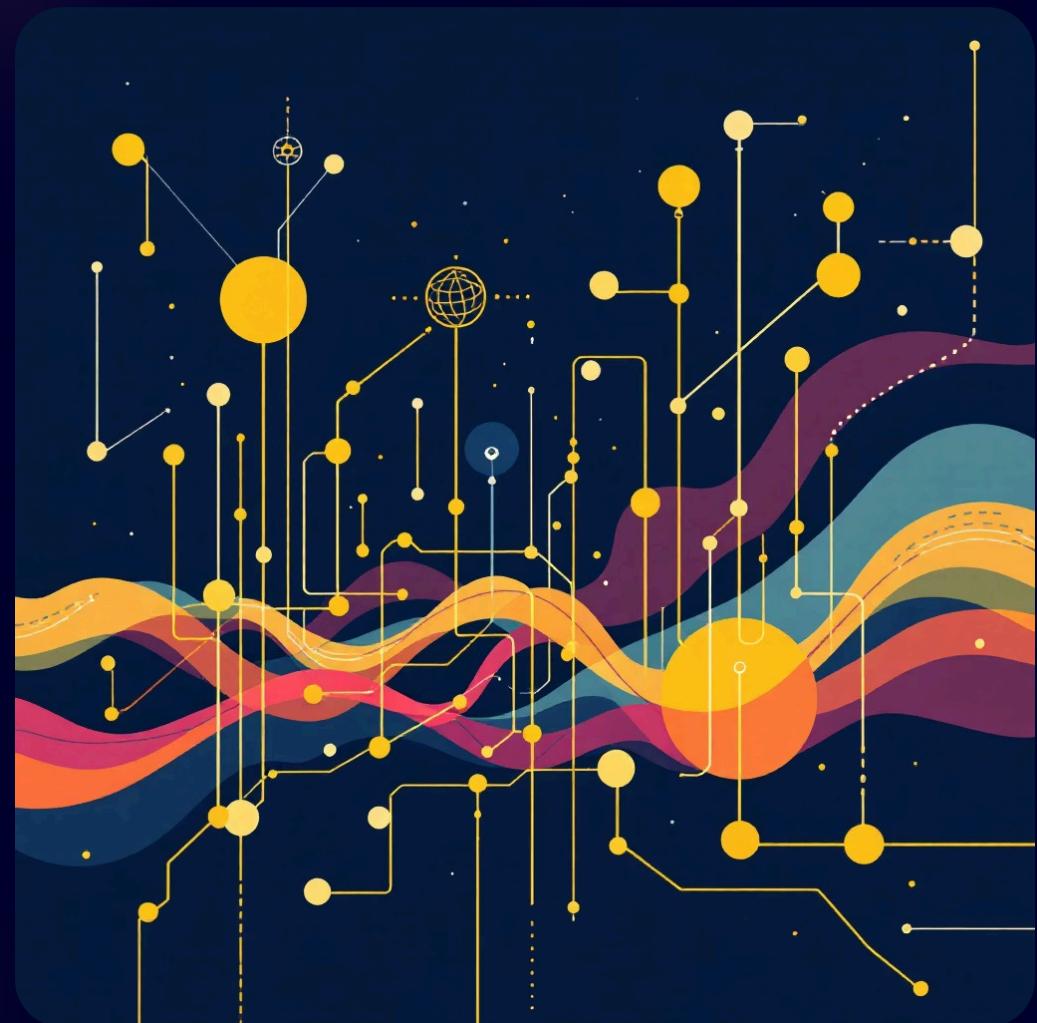
Tradisional: Non-Local Denoising + CLAHE

- **Non-Local Denoising:** Mengurangi noise dengan mencari kesamaan pola, menjaga detail objek.
- **CLAHE:** Meningkatkan kontras lokal secara adaptif, mencegah noise semakin terlihat.



Learning-based: Patch Regression

- **Patch Extraction:** Membagi citra menjadi potongan kecil untuk data latih.
- **Supervised Regression:** Model belajar hubungan citra noisy ke enhanced.
- **Sliding Window:** Menerapkan enhancement ke seluruh citra, digabungkan dengan averaging.



Segmentasi & Evaluasi

Segmentasi SLIC Superpixel

Membagi citra menjadi region kecil (superpixel) berdasarkan kedekatan warna dan lokasi. Kualitas enhancement sangat mempengaruhi pemisahan objek dari background.

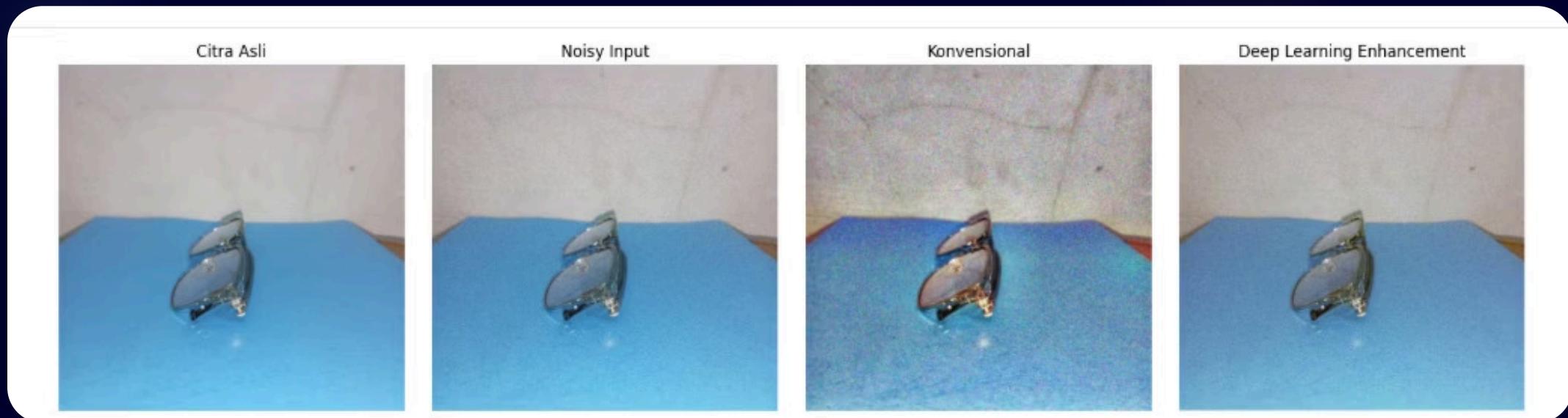


Evaluasi: Dice Score & IoU

Mengukur tingkat kesamaan antara hasil segmentasi prediksi dengan **Ground Truth (GT)**. Nilai mendekati 1 menunjukkan segmentasi sangat baik.



Studi Kasus: Perbandingan Hasil



Citra Asli



Noisy Input



Konvensional



Deep Learning Enhancement

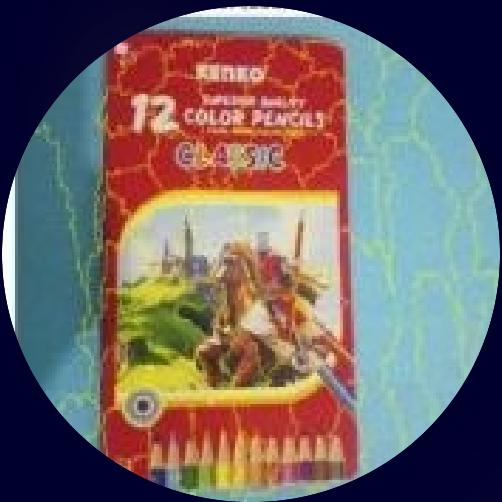


Studi Kasus: Perbandingan Hasil



Kacamata

Learning-based lebih stabil untuk objek tipis/reflektif. Dice: 0.737, IoU: 0.583.



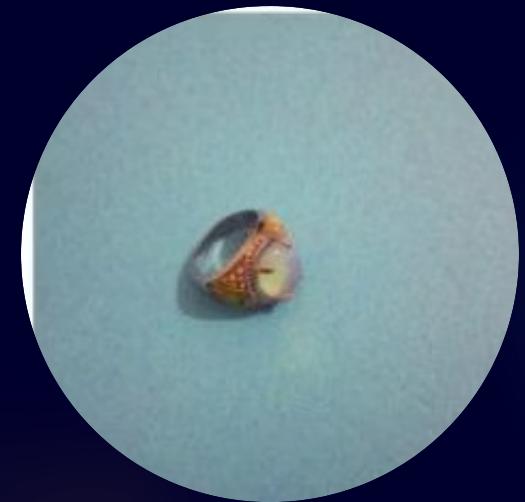
Pensil Warna

Learning-based lebih konsisten menjaga kestabilan citra. Dice: 0.106, IoU: 0.056.



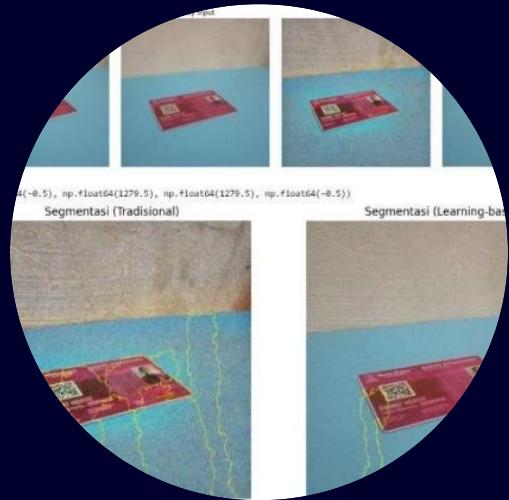
Gelas

Learning-based unggul mengurangi gangguan background. Dice: 0.848, IoU: 0.735.



Cincin

Learning-based lebih baik untuk objek kecil. Dice: 0.987, IoU: 0.974.



Kartu Mahasiswa

Learning-based lebih bersih, tradisional lebih tajam. Dice: 0.880, IoU: 0.786.



Tas

Learning-based lebih stabil, noise removal lebih baik. Dice: 0.792, IoU: 0.655.



Kesimpulan & Saran

Kesimpulan

- Gaussian noise menurunkan kualitas citra & segmentasi SLIC.
- Enhancement tradisional mengurangi noise & meningkatkan kontras, tapi bisa memperjelas background.
- Enhancement learning-based menghasilkan citra lebih stabil & halus, mendukung segmentasi objek.
- Learning-based umumnya lebih baik & konsisten (Dice & IoU) untuk berbagai objek.

Saran

- Gunakan metode deep learning yang lebih kuat (CNN, U-Net).
- Tingkatkan kualitas ground truth (mask manual presisi).
- Tambah variasi dataset & kondisi pengambilan gambar.
- Lakukan tuning parameter enhancement & segmentasi.
- Uji metode segmentasi lain sebagai pembanding.