

KINERJA *RETINEX* SKALA TUNGGAL DAN MULTISKALA PADA PENINGKATAN KUALITAS VISUAL CITRA GELAP

Marlina¹, Bambang Irawan², Ahmad Faqih³, Arif Rinaldi Dikananda⁴, Faturrohman⁵

Program Studi Teknik Informatika¹²³
Program Studi Perangkat Lunak⁴⁵

STMIK IKMI Cirebon
<https://ikmi.ac.id/page/18/?lang=de>

(*) Corresponding Author : mrlina57350@gmail.com;
Published : 30 Januari 2026

Abstract— Improving the visual quality of low-light images is an important challenge in digital image processing, especially when images are used for automatic object classification or detection. In digital agriculture, pest detection based on Convolutional Neural Networks (CNN) often experiences a decrease in accuracy because low lighting makes the visual features of objects less visible. To address this issue, this study applies Retinex image enhancement algorithms, namely Single Scale Retinex (SSR) and Multi Scale Retinex (MSR), as a preprocessing stage on simulated pest images under low-light conditions. The enhanced images are then classified using the MobileNetV2 architecture with a transfer learning approach. The evaluation was conducted using accuracy, precision, recall, F1-score, and loss metrics to assess the effectiveness of SSR and MSR in improving model performance. The results of this study are expected to show that the Retinex algorithm can improve the visual quality of dark images and increase classification accuracy. These findings are expected to support the development of an automatic pest detection system that is more adaptive to variations in field lighting.

Keywords: Retinex Algorithm, Image Enhancement, Low-light Image, Pest Detection, CNN, MobileNetV2, Transfer Learning.

Abstrak— Peningkatan kualitas visual pada citra bercahaya rendah menjadi tantangan penting dalam pengolahan citra digital, terutama ketika citra digunakan untuk klasifikasi atau deteksi objek otomatis. Dalam pertanian digital, deteksi hama berbasis Convolutional Neural Network (CNN) sering mengalami penurunan akurasi karena pencahayaan rendah membuat fitur visual objek kurang terlihat. Untuk mengatasi hal tersebut, penelitian ini menerapkan algoritma peningkatan citra *Retinex*, yaitu Single Scale Retinex (SSR) dan Multi Scale Retinex (MSR), sebagai tahap praproses pada citra hama yang disimulasikan dalam kondisi low-light. Citra hasil peningkatan kemudian diklasifikasikan menggunakan arsitektur *MobileNetV2* dengan pendekatan transfer learning. Evaluasi dilakukan menggunakan metrik akurasi, presisi, recall, F1-score, dan loss untuk menilai efektivitas SSR dan MSR dalam meningkatkan performa model. Hasil penelitian diharapkan menunjukkan bahwa algoritma *Retinex* mampu memperbaiki kualitas visual citra gelap dan meningkatkan akurasi klasifikasi. Temuan ini diharapkan mendukung pengembangan sistem deteksi hama otomatis yang lebih adaptif terhadap variasi pencahayaan lapangan.

Kata Kunci : Algoritma Retinex, Peningkatan Gambar, Citra cahaya rendah, Deteksi Hama, CNN, MobileNetV2, Transfer Learning

INTRODUCTION

Perkembangan teknologi informasi dan kemajuan kecerdasan buatan telah mendorong penerapan sistem berbasis visi komputer di berbagai sektor, termasuk keamanan, kesehatan, agrikultur, dan transportasi cerdas. Keberhasilan sistem tersebut bergantung pada kualitas citra yang dianalisis, karena citra dengan mutu rendah dapat menghambat proses ekstraksi fitur dan menurunkan akurasi model

analisis otomatis [1]. Salah satu tantangan paling signifikan adalah citra dengan pencahayaan rendah, di mana kontras menurun, warna terdistorsi, dan detail visual hilang sehingga memengaruhi performa algoritma pembelajaran mesin dalam mengenali objek secara optimal [2]. Untuk mengatasi kondisi ini, berbagai metode peningkatan citra telah dikembangkan, termasuk algoritma *Retinex* yang meniru mekanisme persepsi cahaya manusia. Dua variannya yang paling umum

digunakan, yaitu Single Scale *Retinex* (SSR) dan Multi Scale *Retinex* (MSR), terbukti mampu meningkatkan pencahayaan dan ketajaman detail pada citra low-light, meskipun performanya masih bervariasi tergantung karakteristik citra dan intensitas iluminasi yang dihadapi [3].

Dalam konteks aplikasi pertanian berbasis visi komputer [4] menyoroti bahwa variasi pencahayaan pada dataset IP102, khususnya citra dengan kecerahan rendah, menjadi salah satu penyebab menurunnya performa deteksi hama. Namun, penelitian tersebut belum mengevaluasi apakah penerapan SSR atau MSR dapat memberikan peningkatan signifikan terhadap hasil klasifikasi. Selain itu, studi komparatif oleh [5] menunjukkan perbedaan dalam stabilitas hasil dan kemampuan mempertahankan detail antarvarian *Retinex*, namun belum menyertakan analisis dampaknya terhadap metrik klasifikasi seperti akurasi dan F1-score. Kesenjangan penelitian ini membuka kebutuhan akan evaluasi yang lebih komprehensif mengenai efektivitas *Retinex* dalam mendukung sistem klasifikasi gambar pada kondisi pencahayaan alami yang kompleks.

Oleh karena itu, penelitian ini melakukan evaluasi komparatif terhadap SSR dan MSR dalam meningkatkan kualitas visual citra low-light serta mengukur pengaruhnya terhadap performa klasifikasi menggunakan arsitektur *MobileNetV2*, yang dikenal efisien dan optimal untuk pemrosesan citra skala besar [6]. Pendekatan ini tidak hanya berfokus pada peningkatan visual, tetapi juga menilai kontribusinya terhadap hasil klasifikasi secara kuantitatif. Dengan demikian, penelitian ini diharapkan memberikan pemahaman yang lebih mendalam mengenai peran algoritma *Retinex* dalam mendukung sistem visi komputer serta menawarkan rekomendasi praktis bagi pengembangan aplikasi analisis citra di lingkungan bercahaya rendah.

MATERIALS AND METHODS

Penelitian ini menggunakan pendekatan eksperimental untuk mengevaluasi kinerja algoritma *Single Scale Retinex* (SSR) dan *Multi Scale Retinex* (MSR) dalam meningkatkan kualitas visual citra bercahaya rendah serta dampaknya terhadap performa klasifikasi menggunakan *MobileNetV2*. Data yang digunakan berasal dari IP102 Pest Dataset, dengan pemilihan 5 kelas hama tanaman (± 4151 citra). Seluruh citra diproses dalam dua versi,

yaitu citra asli dan citra hasil simulasi low-light melalui penurunan intensitas kecerahan.

Dataset and Preprocessing

Citra dari dataset IP102 diunduh dari Kaggle, kemudian diseleksi dan diproses melalui resizing ke 224×224 piksel, normalisasi, konversi ke RGB, serta transformasi ke array numerik. Simulasi low-light dilakukan dengan mengalikan nilai piksel dengan faktor reduksi tertentu untuk menghasilkan citra berintensitas rendah.

Retinex Image Enhancement

Dua algoritma peningkatan pencahayaan diterapkan pada citra *low-light*, yaitu SSR yang menggunakan satu skala Gaussian untuk memperkirakan iluminasi, serta MSR yang menggabungkan beberapa skala SSR untuk menghasilkan pencahayaan yang lebih stabil dan detail. Hasil citra yang telah ditingkatkan kemudian dievaluasi kualitasnya menggunakan metrik PSNR, SSIM, dan NIQE.

Classification Model

Citra hasil enhancement digunakan sebagai input model CNN *MobileNetV2* dengan pendekatan transfer learning. Model dilatih untuk lima kelas hama menggunakan TensorFlow/Keras pada lingkungan Kaggle GPU. Evaluasi model dilakukan menggunakan *accuracy*, *precision*, *recall*, *F1-score*, dan *log loss*.

RESULTS AND DISCUSSION

Hasil

Dataset yang digunakan berasal dari IP102 Pest Dataset yang diunduh dan disimpan pada lingkungan Kaggle. Untuk menjaga keseimbangan kelas, hanya kelas dengan 600–800 citra yang dipilih. Seluruh citra kemudian dimuat menggunakan OpenCV, di-resize menjadi 224×224 piksel, dan dikonversi dari BGR ke RGB sesuai kebutuhan model CNN.

Tahap Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data citra dari IP102 Pest Dataset yang diunduh dan disimpan pada direktori Kaggle. Untuk menjaga keseimbangan kelas, hanya kelas dengan 600–800 citra yang dipilih. Setiap citra kemudian dimuat menggunakan OpenCV, diubah ukurannya menjadi 224×224 piksel, serta dikonversi dari BGR ke RGB agar sesuai dengan standar model deep learning. Proses ini menghasilkan variabel *images* dan *labels* yang berisi citra serta kelas masing-masing. Dataset

yang telah terstruktur ini kemudian digunakan pada tahap pra-pemrosesan berikutnya, termasuk simulasi pencahayaan rendah dan peningkatan kualitas citra menggunakan algoritma *Retinex*.

Tahap Persiapan Citra

```
[ ]:
IMG_SIZE = (224, 224)
BATCH_SIZE = 32

# Image Data Directory
data_dir = '/kaggle/input/rs0-dataset/classification/train/'

# Filter classes with certain amount of size (800 < n < 800)
filtered_class = [ folder for folder in os.listdir(data_dir) if len(os.listdir(os.path.join(data_dir, folder))) > 800 and
                    len(os.listdir(os.path.join(data_dir, folder))) < 800 and folder != '0' ]

# Reads images and labels
images, labels = [], []
for folder in os.listdir(data_dir):
    if folder in filtered_class and folder != '0':
        class_path = os.path.join(data_dir, folder)
        for img_file in tqdm(os.listdir(class_path), desc='[INFO] Load Images ...'):
            img_path = os.path.join(class_path, img_file)
            image = cv2.imread(img_path)
            image = cv2.resize(image, IMG_SIZE)
            image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

            images.append(image)
            labels.append(folder)

# Label Encoder
le = LabelEncoder()
labels_encoded = le.fit_transform(labels)

df = pd.DataFrame({'images': images, 'labels': labels, 'labels_encoded': labels_encoded})
```

Gambar 1. Persiapan Citra

Tahap implementasi dimulai dengan persiapan dan pra-pemrosesan data citra untuk menghasilkan dataset yang konsisten dan seimbang. Kode Python menetapkan ukuran citra 224×224 piksel dan batch size 32, kemudian memuat citra dari direktori menggunakan OpenCV dengan menyaring kelas yang memiliki 600–800 gambar. Setiap citra di-*resize* dan dikonversi dari BGR ke RGB. Label kelas selanjutnya diencode menjadi nilai numerik menggunakan *LabelEncoder*, sebelum seluruh citra dan label disusun dalam sebuah DataFrame.

Simulasi Citra Gelap

```
[ ]:
low_light_images = []

def simulate_low_light(img: np.ndarray, low: float=0.2, high: float=0.5) -> np.ndarray:
    """
    Simulate a low-light condition by reducing the brightness of the image.

    Args:
        img (np.ndarray): Input image (uint8, range 0-255).
        factor (float): Brightness reduction factor (0 < factor < 1).
            e.g., 0.3 means reduce brightness by 70%.

    Returns:
        np.ndarray: Darkened image.
    """
    factor = np.random.uniform(low, high)
    # Ensure float32 for multiplication, then clip and convert back
    img_dark = np.clip(img.astype(np.float32) * factor, 0, 255).astype(np.uint8)
    return img_dark

for image in tqdm(images, desc='[INFO] Process ...'):
    low_light_images.append(simulate_low_light(image, high=0.25))
```

Setelah data citra dasar siap, langkah selanjutnya adalah menghasilkan citra low-light secara artifisial dengan memanipulasi komponen Value (V) pada ruang warna HSV. Citra RGB dikonversi ke HSV, lalu intensitas komponen V dikurangi menggunakan faktor

gamma 0.2 untuk mensimulasikan kondisi gelap. Komponen V yang telah dimodifikasi digabungkan kembali dengan H dan S, kemudian citra dikonversi lagi ke format RGB untuk digunakan sebagai input pada algoritma *Retinex*.

Penerapan Algoritma *Retinex*

```
def SSR(img, sigma=15):
    img = img.astype(np.float32) + 1.0
    blur = cv2.GaussianBlur(img, (0, 0), sigma)
    retinex = np.log(img) - np.log(blur + 1)
    return cv2.normalize(retinex, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)

def MSR(img: np.ndarray, sigma_list=[15, 80, 250]) -> np.ndarray:
    """
    Apply Multi-Scale Retinex (MSR) to an image, with output normalization.
    """
    img = img.astype(np.float32) + 1.0
    msr = np.zeros_like(img, dtype=np.float32)

    for sigma in sigma_list:
        msr += SSR(img, sigma)
    msr /= len(sigma_list)

    # Normalize each channel individually
    for c in range(msr.shape[2]):
        channel = msr[:, :, c]
        max_val = np.max(channel)
        min_val = np.min(channel)
        msr[:, :, c] = 255 * (channel - min_val) / (max_val - min_val + 1e-6)

    return msr.astype(np.uint8)
```

Gambar 2. Algoritma retinex

Ini merupakan inti eksperimen, di mana algoritma *Retinex* diterapkan pada citra gelap menggunakan fungsi *singleScaleRetinex* dan *multiScaleRetinex*. Implementasi dimulai dengan mengubah citra ke domain logaritmik sesuai prinsip *Retinex*. SSR menghitung reflektansi dengan memperkirakan iluminasi melalui Gaussian blur tunggal (sigma = 100). Sementara itu, MSR menggabungkan tiga SSR dari skala berbeda (sigma 15, 80, 250) yang dijumlahkan secara berbobot (1/3), kemudian dinormalisasi dan dikembalikan ke domain spasial.

Tahap Aarsitektur dan Pemodela

hasil dari tahap ini adalah visualisasi arsitektur model dan kurva kinerja model selama pelatihan, yang menunjukkan tren *loss* dan *accuracy* terhadap *epoch*.



Gambar 3. Arsitektur

Tahap Pengukuran Performa Model

Tahap akhir ini mengukur performa model *MobileNetV2* terhadap tiga skenario citra uji (Asli,

SSR, MSR). Model memprediksi label masing-masing citra, kemudian hasilnya dievaluasi menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* melalui perbandingan prediksi dengan label sebenarnya. Bagian ini membahas interpretasi hasil penelitian dengan menghubungkan temuan ke teori dan studi sebelumnya, berfokus pada efektivitas SSR dan MSR dalam meningkatkan kualitas citra low-light serta dampaknya terhadap performa klasifikasi menggunakan *MobileNetV2*. Hasil Kuesioner

Peningkatan Performa Akurasi Setelah Penerapan Algoritma SSR dan MSR Pembahasan ini menganalisis peningkatan akurasi *MobileNetV2* setelah citra ditingkatkan menggunakan SSR dan MSR. Pada baseline, akurasi sebesar 69.85%. Setelah penerapan SSR, akurasi naik menjadi 75.12%, dan MSR menghasilkan peningkatan tertinggi sebesar 77.68%. Kenaikan ini menunjukkan bahwa *Retinex* mampu memperbaiki kualitas citra *low-light* sehingga fitur lebih mudah diekstraksi oleh model. Temuan tersebut sejalan dengan penelitian sebelumnya yang menyatakan bahwa *Retinex* efektif menangani variasi pencahayaan dan meningkatkan hasil deteksi berbasis CNN, terutama pada citra bercahaya rendah.

Komparasi Metode *Retinex* Terbaik Berdasarkan Akurasi dan *Loss* MSR terbukti paling optimal dengan akurasi 77.68%, lebih tinggi dari SSR (75.12%). SSR hanya memakai satu skala sehingga mudah menghasilkan distorsi, sedangkan MSR menggunakan tiga skala yang mampu menjaga detail dan kontras secara lebih seimbang. Pendekatan multi-skala ini menghasilkan kualitas reflektansi lebih akurat, sehingga meningkatkan performa *MobileNetV2*. Temuan ini konsisten dengan penelitian yang menegaskan efektivitas metode *Retinex* berbasis multi-skala dan dekomposisi iluminasi dalam peningkatan citra.

Analisis Peningkatan *F1-Score* MSR Dibandingkan SSR *F1-Score* meningkat dari 74.01% (SSR) menjadi 76.95% (MSR), menunjukkan bahwa MSR lebih efektif menyeimbangkan *Precision* dan *Recall*. SSR yang hanya memakai satu skala lebih rentan kehilangan detail, sedangkan MSR dengan kombinasi multi-skala menghasilkan citra yang lebih stabil dan informatif, sehingga meningkatkan kinerja klasifikasi.

CONCLUSION

Penelitian ini menunjukkan bahwa algoritma *Retinex* efektif meningkatkan performa klasifikasi citra gelap pada

MobileNetV2, dengan akurasi meningkat dari 69.85% menjadi 75.12% pada SSR dan mencapai hasil terbaik 77.68% pada MSR, yang juga menghasilkan *F1-Score* tertinggi 76.95% berkat kemampuannya mempertahankan detail multi-skala. Untuk pengembangan lebih lanjut, disarankan menguji varian *Retinex* yang lebih canggih seperti MSRCR atau model berbasis deep learning, melakukan optimasi hyperparameter serta membandingkan arsitektur lain seperti EfficientNet atau ResNet, dan memvalidasi metode menggunakan dataset low-light alami agar hasil penelitian lebih robust dan aplikatif di kondisi nyata.

DAFTAR PUSTAKA

- [1] Z. Li, Y. Yang, and Y. Wang, "A survey of image preprocessing methods for deep learning," *J. Imaging*, vol. 7, no. 5, p. 78, 2021, doi: 10.3390/jimaging7050078.
- [2] K. G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognit.*, vol. 61, pp. 650–662, 2021, doi: 10.1016/j.patcog.2021.107674.
- [3] L. Ma, Y. Zhang, and J. Wu, "Comprehensive evaluation of *Retinex*-based algorithms for low-light image enhancement in real-world scenarios," *Signal Process. Image Commun.*, vol. 105, p. 116715, 2022, doi: 10.1016/j.image.2022.116715.
- [4] Z. Liu, J. Zhang, and Z. Tang, "Benchmarking low-light image enhancement methods: Synthetic and real-world data," *J. Vis. Commun. Image Represent.*, vol. 78, p. 103102, 2021, doi: 10.1016/j.jvcir.2021.103102.
- [5] A. Rahman, M. Islam, and M. Hasan, "A comparative study on *Retinex*-based enhancement methods for low-light image," *ICT Express*, vol. 8, no. 1, pp. 36–42, 2022, doi: 10.1016/j.icte.2021.05.004.
- [6] Y. Liu, H. Zhang, and L. Zhao, "A deep learning-based method for pest detection using the IP102 dataset," *Comput. Electron. Agric.*, vol. 184, p. 106064, 2021, doi: 10.1016/j.compag.2021.106064.