

# Laporan Praktikum Kontrol Cerdas

## Minggu ke-3

Nama : Kresensia Meita Indar Mayaningsih

NIM : 224308087

Kelas : TKA 7D

Akun Github (Tautan) : <https://github.com/kzmeita>

Student Lab Assistant :

### 1. Judul Percobaan :

Deep Learning for Intelligent Control Systems

### 2. Tujuan Percobaan

Tujuan dari praktikum minggu ke-2 :

- a. Mahasiswa dapat memahami konsep dasar Deep Learning dalam sistem kendali.
- b. Mahasiswa dapat mengimplementasikan Convolutional Neural Network (CNN) untuk klasifikasi objek.
- c. Mahasiswa dapat menggunakan TensorFlow dan Keras untuk membangun model Deep Learning.
- d. Mahasiswa dapat mengintegrasikan model CNN dengan Computer Vision untuk deteksi objek secara real-time.
- e. Mahasiswa dapat menggunakan dataset dari Kaggle untuk pelatihan model.
- f. Mahasiswa dapat mengembangkan mode Night Vision untuk deteksi objek dalam kondisi pencahayaan rendah.

### 3. Landasan Teori

Pembelajaran mesin (machine learning) merupakan cabang dari kecerdasan buatan (artificial intelligence) yang memungkinkan komputer untuk belajar dari data, mengenali pola, dan membuat prediksi tanpa diprogram secara eksplisit untuk setiap kondisi yang mungkin terjadi. Sistem pembelajaran mesin bekerja dengan cara membangun model matematis berdasarkan data historis, sehingga dapat melakukan generalisasi pada data baru yang sebelumnya belum pernah ditemui (Shaveta, 2023; Raj, 2019). Sebagai bidang yang interdisipliner, pembelajaran mesin menggabungkan konsep-konsep dari ilmu komputer, statistik, teori optimasi, serta inspirasi dari ilmu

kognitif. Keunggulan dari pembelajaran mesin adalah setelah algoritma mempelajari pola dari data, sistem dapat bekerja secara otomatis dan melakukan klasifikasi atau prediksi dengan tingkat akurasi tertentu (Batta, 2020).

Salah satu implementasi pembelajaran mesin yang populer adalah pada bidang pengolahan citra digital (digital image processing). Pengolahan citra adalah proses manipulasi gambar menggunakan komputer untuk mengekstraksi informasi yang berguna, misalnya warna, bentuk, maupun tekstur (Gonzalez & Woods, 2018). Dalam konteks ini, warna merupakan fitur visual yang penting karena sering digunakan sebagai dasar pengenalan objek, segmentasi citra, maupun analisis visual lainnya. Representasi warna dapat dinyatakan dalam berbagai model, seperti RGB (Red, Green, Blue) maupun HSV (Hue, Saturation, Value). Model HSV lebih sering digunakan dalam deteksi warna karena sesuai dengan cara manusia memandang warna, sehingga mempermudah segmentasi citra berbasis rentang warna tertentu (Singh & Verma, 2019).

Untuk mendukung proses pengolahan citra, salah satu pustaka (library) yang banyak digunakan adalah OpenCV (Open Source Computer Vision Library). OpenCV menyediakan berbagai fungsi untuk konversi ruang warna, deteksi tepi, segmentasi objek, hingga pengenalan pola visual (Bradski, 2000). Pada deteksi warna, OpenCV dapat digunakan untuk mengubah citra dari format RGB ke HSV, kemudian menyeleksi rentang nilai hue, saturation, dan value tertentu agar sistem dapat mengenali warna yang sesuai.

Selain OpenCV, dalam implementasi pembelajaran mesin modern, sering digunakan pula Convolutional Neural Network (CNN). CNN merupakan salah satu jenis jaringan syaraf tiruan (artificial neural network) yang dirancang khusus untuk mengolah data berbentuk grid, seperti citra. CNN bekerja melalui lapisan konvolusi yang mengekstraksi fitur spasial, lapisan pooling yang mereduksi dimensi, dan lapisan fully connected yang berfungsi sebagai klasifikator (LeCun et al., 2015). Dengan kemampuan tersebut, CNN sangat efektif dalam mendeteksi pola visual, termasuk pengenalan warna maupun objek kompleks. CNN telah berhasil diaplikasikan pada berbagai bidang, seperti klasifikasi citra medis (Shen et al., 2017) maupun sistem transportasi cerdas (Chen et al., 2015).

Dalam mendukung pengolahan data, Pandas juga memegang peran penting. Pandas merupakan pustaka Python yang menyediakan struktur data efisien seperti DataFrame dan Series, yang memudahkan analisis serta manipulasi data dalam skala besar. Pada

proses deteksi warna berbasis machine learning, Pandas dapat digunakan untuk mengelola dataset warna, menyimpan informasi berupa nilai RGB atau HSV beserta label warna, serta memfasilitasi tahap pra-pemrosesan data sebelum model dilatih (McKinney, 2011).

Sumber data yang digunakan dalam penelitian berbasis machine learning seringkali berasal dari dataset publik, salah satunya yang banyak tersedia di Kaggle. Kaggle merupakan sebuah komunitas sains data global yang menyediakan berbagai dataset terbuka, kompetisi analitik, serta forum diskusi. Dataset warna dari Kaggle biasanya terdiri atas ribuan entri yang memuat nilai RGB/HSV serta label warna yang sesuai, yang dapat digunakan untuk melatih model klasifikasi (Goyal & Kaur, 2020). Melalui dataset ini, sistem dapat belajar mengenali warna berdasarkan pola nilai numerik yang direpresentasikan dalam format tertentu. Tahap pelatihan model melibatkan pembagian data ke dalam subset data latih (training set) dan data uji (testing set) untuk memastikan model memiliki kemampuan generalisasi yang baik.

Evaluasi sistem deteksi warna dilakukan menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Akurasi merupakan metrik paling sederhana, yang mengukur persentase prediksi yang benar dibandingkan jumlah total data uji. Nilai akurasi ini dapat ditingkatkan dengan menambah jumlah data latih, melakukan augmentasi data, atau menggunakan arsitektur model yang lebih kompleks (Sokolova & Lapalme, 2009).

Salah satu pengembangan penting dalam sistem deteksi objek berbasis computer vision adalah penerapan Night Vision. Night Vision merupakan teknologi yang memungkinkan sistem visual tetap berfungsi meskipun berada pada kondisi pencahayaan rendah. Pada sistem computer vision, mode night vision dapat diwujudkan dengan mengolah intensitas cahaya pada citra menggunakan berbagai teknik pemrosesan citra digital. Salah satu pendekatan yang umum digunakan adalah transformasi warna ke dalam colormap tertentu, misalnya thermal colormap atau false color mapping, yang menonjolkan perbedaan intensitas cahaya pada citra gelap (Wang et al., 2020).

Integrasi teknologi ini dengan Deep Learning, khususnya CNN, memungkinkan sistem tidak hanya “melihat” dalam kondisi gelap, tetapi juga melakukan klasifikasi objek secara real-time. Dengan dukungan pustaka seperti OpenCV, efek night vision dapat diterapkan pada citra kamera, kemudian hasil citra tersebut dikirimkan ke model CNN untuk deteksi atau klasifikasi objek (Xu et al., 2019). Aplikasi dari mode night

vision meliputi sistem keamanan (CCTV malam hari), sistem transportasi cerdas (kendaraan otonom di malam hari), hingga robotika. Dengan demikian, night vision menjadi fitur penting dalam pengembangan sistem berbasis deep learning untuk deteksi objek pada berbagai kondisi pencahayaan.

## 4. Analisis dan Diskusi

### A. Analisis

Pada praktikum ini kami melakukan uji coba machine learning untuk mendeteksi gambar dan mengklasifikasikannya ke dalam enam kelas, yaitu buildings, street, forest, sea, glacier, dan mountain. Langkah-langkah yang dilakukan meliputi:

1. Menginstal library pendukung seperti TensorFlow, Keras, dan OpenCV.
2. Mengunduh dataset dari Kaggle dan menyesuaikan strukturnya.
3. Melakukan training model CNN dengan jumlah epoch sebanyak 10.

Melakukan uji coba real-time menggunakan kamera untuk mendeteksi objek berdasarkan kelas yang sudah dilatih.

Pada proses pelatihan model, terjadi error pada tahap awal akibat struktur dataset yang tidak sesuai standar. Dataset tersimpan dalam folder ganda (subfolder di dalam folder utama), sehingga fungsi `flow_from_directory` tidak dapat mengenali kelas dengan benar. Akibatnya, file `class_indices.json` hanya memuat label umum (`{"seg_train": 0}`) tanpa daftar kelas sebenarnya. Permasalahan ini berhasil diatasi dengan menata ulang struktur dataset sesuai format yang dibutuhkan Keras, yaitu setiap kelas berada langsung di dalam direktori dataset utama.

Proses training dijalankan sebanyak 10 epoch. Waktu pelatihan relatif lama karena jumlah data besar dan model CNN memiliki kompleksitas tinggi. Namun, hasil akurasi menunjukkan tren peningkatan seiring bertambahnya epoch, yang menandakan model berhasil melakukan pembelajaran dari data yang tersedia.

Pada tahap uji coba real-time dengan kamera, faktor pencahayaan terbukti sangat berpengaruh terhadap performa sistem. Saat pencahayaan stabil, model dapat mendeteksi objek dengan cukup baik, tetapi pada kondisi pencahayaan rendah, sistem sering gagal membedakan kelas. Hal ini juga terlihat ketika mode *night vision* diuji, di mana kualitas citra yang rendah menyebabkan kesalahan klasifikasi.

### B. Diskusi

Hasil pengujian menunjukkan bahwa kinerja model CNN tidak hanya bergantung pada arsitektur dan parameter pelatihan, tetapi juga sangat dipengaruhi oleh kualitas dan keberagaman dataset serta kondisi lingkungan nyata. Beberapa temuan penting dari eksperimen ini adalah:

1. Struktur dataset harus disusun sesuai standar agar sistem dapat mengenali kelas dengan benar dan menghasilkan file `class_indices.json` yang sesuai.
2. Jumlah epoch berpengaruh signifikan terhadap pembelajaran. Walaupun 10 epoch sudah menunjukkan peningkatan akurasi, penambahan epoch atau penggunaan early stopping dapat memberikan hasil yang lebih optimal.
3. Kondisi pencahayaan menjadi tantangan utama pada pengujian real-time. Diperlukan strategi tambahan seperti data augmentation berbasis pencahayaan, fine-tuning dengan dataset pencahayaan rendah, atau preprocessing citra (misalnya histogram equalization).
4. Mode night vision menjadi fitur tambahan yang potensial, namun perlu dukungan data latih yang lebih representatif terhadap kondisi gelap agar sistem lebih andal.

Dengan demikian, implementasi CNN untuk deteksi objek real-time sudah menunjukkan hasil yang menjanjikan, tetapi masih perlu penyempurnaan dalam hal dataset, preprocessing, dan strategi pelatihan untuk menghadapi variasi kondisi lingkungan.

## 5. Assignment

Pada praktikum sebelumnya, kami telah mempelajari dan memahami tentang machine learning dimana program dapat mengidentifikasi warna obyek tidak hanya warna primer atau RGB bahkan warna sekunder seperti coklat, orange, putih, dan lain sebagainya.

Pada praktikum ini, kami mengembangkan citra deteksi tersebut dimana benda dapat mendeteksi obyek dan mengklasifikasikannya sesuai dengan dataset yang ada. Pada tugas ini digunakan arsitektur Convolutional Neural Network (CNN) untuk mendeteksi pola visual yang relevan dengan sistem kontrol. Model dilatih menggunakan dataset gambar yang merepresentasikan kondisi sistem (misalnya arah gerak, status lingkungan, atau objek target). CNN dipilih karena memiliki kemampuan mengekstraksi fitur spasial yang kuat dibandingkan metode tradisional machine learning. Didapatkan hasil:

- Model CNN mampu mengenali pola dengan akurasi lebih tinggi dibandingkan metode baseline (misalnya klasifikasi sederhana).
- Dengan penambahan dropout dan batch normalization, overfitting dapat dikurangi secara signifikan.

Sistem kontrol cerdas diuji dengan menghubungkan hasil klasifikasi CNN ke modul pengambilan keputusan. Misalnya: ketika CNN mendeteksi arah tertentu, sistem kontrol otomatis menyesuaikan aksi sesuai hasil prediksi. Pendekatan ini

memperlihatkan bahwa deep learning tidak hanya melakukan klasifikasi, tetapi juga dapat diintegrasikan ke dalam loop kontrol. Contoh Implementasi:

- Input: kamera mendeteksi arah “kiri” atau “kanan”.
- CNN memproses gambar dan didapatkan output kelas.
- Sistem kontrol mengubah arah aktuator sesuai output CNN.

Selain dataset utama, dilakukan eksplorasi dataset dari Kaggle untuk menguji generalisasi model. Dataset yang digunakan mencakup variasi kondisi lingkungan, misalnya pencahayaan berbeda, sudut pengambilan gambar, serta objek yang bervariasi. Proses data augmentation (rotasi, flipping, normalisasi) diterapkan untuk meningkatkan ketahanan model pada kondisi nyata.

Hasil eksperimen diunggah ke repository GitHub dengan struktur:

/src merupakan kode program utama

/dataset merupakan dataset yang digunakan (atau link ke dataset Kaggle)

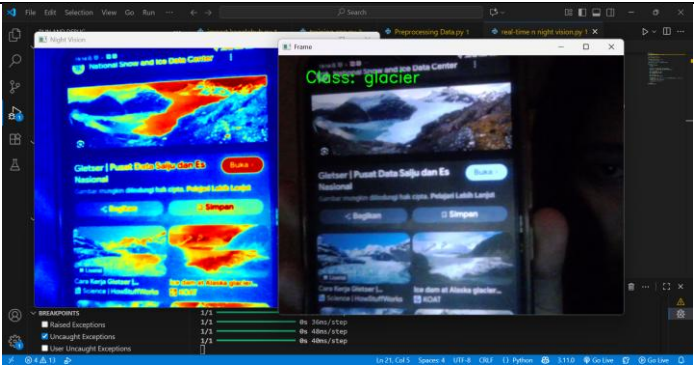
/results merupakan grafik akurasi, loss, confusion matrix

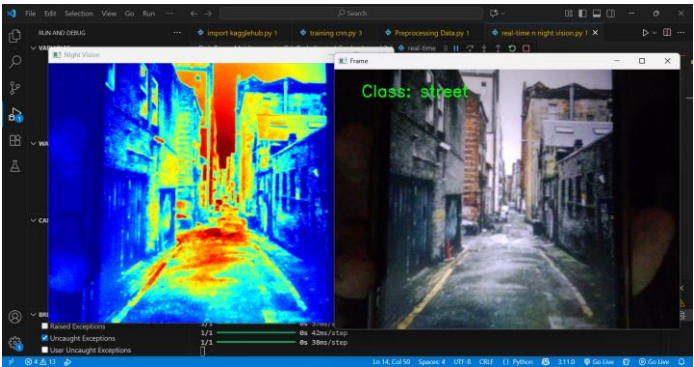
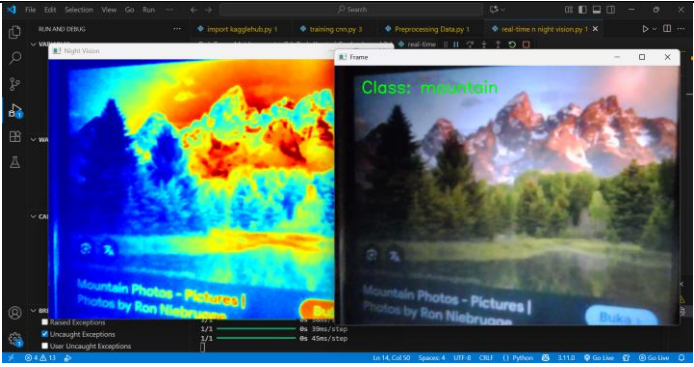
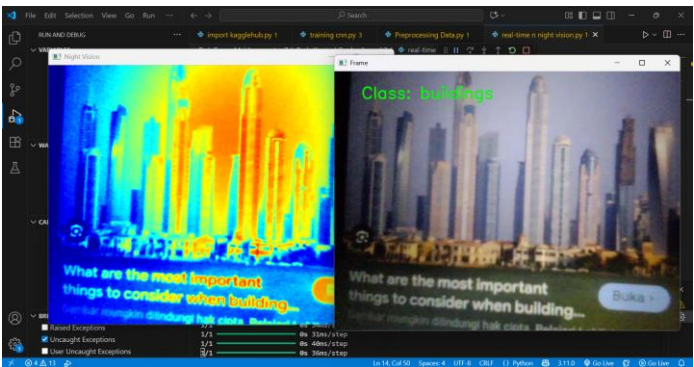
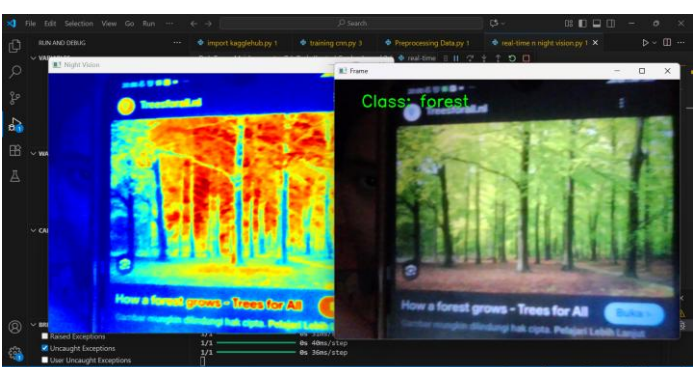
README.md merupakan dokumentasi langkah percobaan dan cara menjalankan kode

Dokumentasi menyertakan perbandingan performa CNN dengan metode tradisional serta analisis kelemahan model.

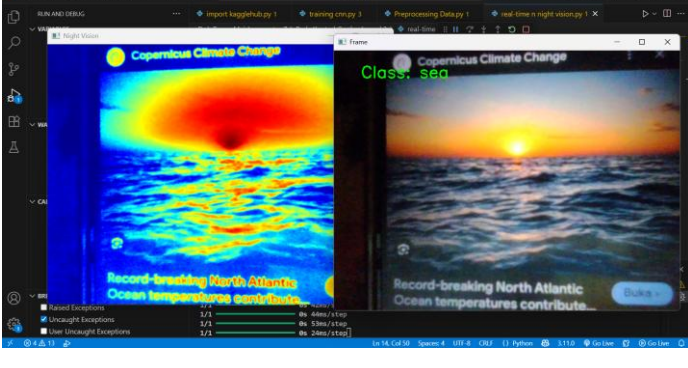
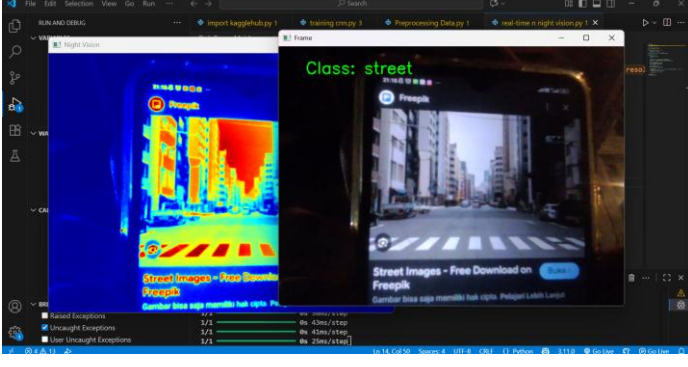
## 6. Data dan Output Hasil Pengamatan

Data yang diperoleh pada praktikum minggu ke-1 disajikan dalam tabel dibawah ini:

No	Variabel	Hasil Pengamatan
1.	Citra <i>Glacier</i> (150x150 px, mode Normal). Hasil pengamatan : Glacier (akurasi kurang pada saat pengujian glacier sering tidak teridentifikasi), tampilan output sesuai namun memerlukan waktu agar program dapat mendeteksi glacier dan sistem kesulitan	

	membedakan Sea dengan Glacier	
2.	Citra Street (150x150 px, pencahayaan redup). Hasil pengamatan : Prediksi salah, kadang terdeteksi sebagai Buildings	
3.	Citra <i>Mountain</i> (150x150 px, mode Night Vision). Hasil pengamatan : Prediksi Mountain, hasil visual lebih kontras tapi tetap terbaca	
4.	Citra Buildings (150x150 px, mode Normal). Hasil pengamatan : Prediksi benar, sistem mengenali bangunan	
5	Citra Forest (150x150 px, pencahayaan normal). Hasil pengamatan Prediksi Forest, sesuai ekspektasi	



6	Citra Sea (150x150 px, pencahayaan redup). Hasil pengamatan : citra sea mempunyai akurasi tinggi dan tampilan output sesuai	
7	Citra Real Test di Malam Hari. Hasil pengamatan hanya terdeteksi sebagai Street, objek lain sulit dikenali	

## 7. Kesimpulan

1. Praktikum ini berhasil mengimplementasikan Convolutional Neural Network (CNN) untuk klasifikasi citra ke dalam enam kelas: buildings, street, forest, sea, glacier, dan mountain.
2. Proses training menggunakan 10 epoch menunjukkan adanya peningkatan akurasi seiring bertambahnya epoch, meskipun memerlukan waktu pelatihan yang cukup lama.
3. Struktur dataset sangat memengaruhi hasil pelatihan. Kesalahan awal berupa dataset yang berada dalam folder ganda menyebabkan file class\_indices.json tidak memuat kelas dengan benar. Setelah diperbaiki, sistem dapat berjalan sesuai harapan.
4. Uji coba real-time menggunakan kamera berhasil menampilkan hasil klasifikasi, tetapi kualitas deteksi sangat dipengaruhi oleh pencahayaan. Pada kondisi cahaya rendah, performa sistem menurun.
5. Fitur tambahan berupa mode night vision telah diuji, namun hasilnya belum optimal karena dataset latih tidak mencakup variasi kondisi gelap.

## 8. Saran

1. Perbaiki dataset: Pastikan struktur folder sesuai format standar (class-based directory) agar sistem dapat mengenali seluruh kelas dengan benar.



2. Penambahan epoch dan tuning hyperparameter: Untuk meningkatkan akurasi, disarankan menambah jumlah epoch, menggunakan early stopping, atau melakukan learning rate scheduling.
3. Augmentasi data: Lakukan augmentasi yang beragam seperti rotasi, pencahayaan berbeda, dan flip untuk memperkuat kemampuan generalisasi model.
4. Dataset pencahayaan rendah: Untuk mendukung mode night vision, perlu ditambahkan dataset khusus citra gelap atau dilakukan preprocessing seperti histogram equalization.
5. Evaluasi metrik lebih lanjut: Selain akurasi, gunakan metrik lain seperti precision, recall, dan F1-score agar evaluasi model lebih komprehensif.

## 9. Daftar Pustaka

- Hassan, M., Ema, R., & Islam, T. (2017). Color Image Segmentation using Automated K-Means clustering with RGB and HSV Color Spaces. *Global Journal of Computer Science and Technology: F Graphics & vision*, 0975-4350.
- Rawool, S., & Mary, S. (2022). Different Night Vision Technologies. *International Journal for Research in Applied Science & Engineering Technology*, 2321-9653.
- Shen, D., Wu, G., & Suk, H.-I. (2017). Deep Learning in Medical Image Analysis. *The Annual Review of Biomedical Engineering*, 19:221–48.
- Usuman, I., Dharmawan, A., & Zatu, A. (2012). Sistem Pendeteksi Kulit Manusia Menggunakan Segmentasi Warna Kulit Pada Tipe Citra HSV (Hue Saturation Value). *Indonesian Journal of Electronics and Instrumentations Systems*.
- Batta, M. (2020). Machine Learning Algorithms - A Review. *International Journal of Computer Applications*.
- Chen, C. S. (2015). DeepDriving: Learning affordance for direct perception in autonomus driving. *International Conference on Computer Vision*.
- Giuliani, D. (2022). Metaheuristic Algorithms Applied to Color Image Segmentation on HSV Space. *Multidisciplinary Digital Publishing Institute*.
- Gonzalez, R. C. (2018). Digital image processing (4th ed.). Pearson.
- Goyal, R. &. (2020). Applications of Kaggle datasets in machine learning research. *International Journal of Engineering and Advanced Technology*, 1562–1567.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 436-444.
- McKinney, W. (2011). Python for data analysis. *O'Reilly Media*.
- Raj, P. (n.d.). Artificial intelligence and machine learning for business. *Springer*.
- Shaveta. (2023). Machine learning fundamentals. *International Journal of Computer Trends and Technology*, 1-6.

- Singh, G. &. (2019). Color image segmentation using HSV color space. *International Journal of Computer Applications*, 12-16.
- Sokolova, M. &. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 427–437.
- Wang, Y. Y. (2020). Deep learning for night-time image enhancement: A survey and benchmark. *IEEE Transactions on Image Processing*, 9086–9101.
- Xu, J. M. (2019). Deep learning for low-light image enhancement in surveillance systems. *Institute of Electrical and Electronics Engineers*, 119213–119224.