

Model Ekstraksi Fitur Berbasis Tekstur untuk Identifikasi Keaslian Objek

Florentina Tatrini Kurniati
Institut Teknologi dan Bisnis STIKOM Bali
e-mail: florent@stikom-bali.ac.id

Diajukan: 19 Maret 2024; Direvisi: 19 April 2024; Diterima: 22 April 2024

Abstrak

Identifikasi keaslian objek merupakan aspek penting dalam berbagai sektor, termasuk keamanan dan perdagangan, guna mencegah kerugian finansial dan reputasi akibat pemalsuan. Penelitian ini mengembangkan model klasifikasi berbasis tekstur dengan menggunakan metode Decision Tree dan Logistic Regression untuk membedakan antara objek asli dan palsu. Model ini memanfaatkan Gray Level Co-occurrence Matrix (GLCM) untuk ekstraksi fitur tekstur, yang kemudian diklasifikasi menggunakan kedua metode tersebut. Hasil evaluasi menunjukkan bahwa Decision Tree memiliki kinerja yang lebih unggul dibandingkan Logistic Regression, dengan akurasi sebesar 96.37%, recall 97.67%, dan F1 score 94.92%, menjadikannya lebih efektif dalam mendeteksi keaslian objek. Sementara itu, Logistic Regression mencapai presisi lebih tinggi, yaitu 98.15%, namun mengalami penurunan performa dalam recall dan F1 score. Berdasarkan hal tersebut Decision Tree menunjukkan keseimbangan yang lebih baik antara berbagai metrik evaluasi dan lebih cocok untuk identifikasi keaslian objek.

Kata kunci: Identifikasi keaslian, Decision tree, Logistic regression, Ekstraksi tekstur, Klasifikasi objek.

Abstract

Object authenticity identification is a crucial aspect in various sectors, including security and commerce, to prevent financial and reputational losses due to counterfeiting. This research developed a texture-based classification model using Decision Tree and Logistic Regression methods to differentiate between genuine and counterfeit objects. The model utilizes the Gray Level Co-occurrence Matrix (GLCM) for texture feature extraction, which is then classified using these methods. The evaluation results show that the Decision Tree outperforms Logistic Regression, with an accuracy of 96.37%, recall of 97.67%, and F1 score of 94.92%, making it more effective in detecting object authenticity. Meanwhile, Logistic Regression achieved higher precision, at 98.15%, but experienced a decline in performance in terms of recall and F1 score. Based on these findings, the Decision Tree demonstrates a better balance across various evaluation metrics and is more suitable for object authenticity identification.

Keywords: Authenticity identification, Decision tree, Logistic regression, Texture extraction, Object classification.

1. Pendahuluan

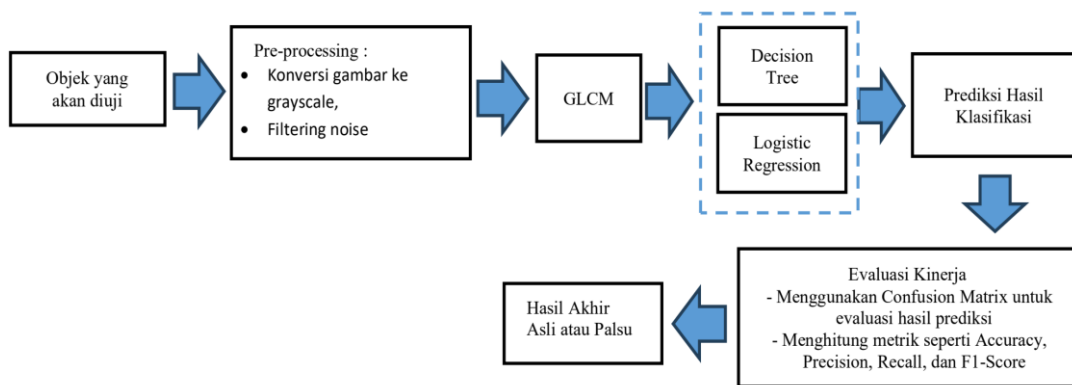
Identifikasi keaslian objek telah menjadi perhatian penting dalam berbagai sektor, dari keamanan hingga perdagangan[1]. Maraknya produk tiruan dan replikasi yang semakin sulit dibedakan dari aslinya. Kebutuhan akan metode yang andal untuk memastikan keaslian suatu objek semakin dibutuhkan[2]. Pemalsuan bukan hanya menimbulkan kerugian finansial yang besar bagi individu dan perusahaan, tetapi juga dapat merusak reputasi, menurunkan kualitas produk di pasaran. Dalam konteks keamanan, terutama di sektor perbankan dan pemerintahan, identifikasi keaslian objek sangat penting untuk mencegah penipuan[3][4]. Dalam penerbitan paspor, kartu identitas, atau dokumen resmi lainnya. Untuk pencegahan dengan memastikan keaslian dokumen ataupun objek sehingga pencurian identitas dan penggunaan identitas palsu untuk tujuan ilegal dapat diminimalkan[5].

Identifikasi keaslian prosesnya menggunakan detail-detail fitur yang dapat menentukan keaslian sebuah objek[6][7]. Salah satu aspek utama yang diteliti menggunakan pendekatan tekstur sebagai salah satu fitur dasar untuk identifikasi suatu objek asli atau palsu[8]. Fitur dari objek asli dan palsu digunakan dan dimanfaatkan untuk dilatihkan pada machine learning. Untuk memperoleh fitur prosesnya dengan ekstraksi yang berfokus pada tekstur objek. Ekstraksi fitur berbasis tekstur menjadi dasar digunakan untuk

memperoleh pola-pola tekstur yang melekat pada objek asli, yang biasanya memiliki karakteristik unik yang sulit ditiru oleh pemalsu[9]. Dengan menganalisis pola-pola tekstur yang ada dalam objek menggunakan machine learning, model akan mampu membedakan antara objek asli dan palsu. Decision Trees dan Logistic Regression merupakan metode machine learning yang dapat digunakan untuk menganalisis tekstur dengan proses klasifikasi[10][11]. Pengembangan model identifikasi dengan ekstraksi fitur berbasis tekstur untuk mendeteksi keaslian objek membandingkan antara Decision Trees dengan Logistic Regression dapat digunakan untuk membedakan suatu objek asli ataupun palsu.

2. Metode Penelitian

Pada penelitian ini mengembangkan model identifikasi keaslian objek dengan tahapan pre-processing, ekstraksi menggunakan GLCM, fitur yang diperoleh diklasifikasi menggunakan metode *Decision Tree dan Logistic Regression*. Hasil klasifikasi untuk menentukan keaslian objek dievaluasi menggunakan confusion matrix[12]. Model pada penelitian ditunjukkan pada Gambar 1.



Gambar 1. Model Identifikasi Keaslian Objek.

Pada model identifikasi keaslian objek Gambar 1, salah satu bagian utama prosesnya yaitu metode ekstraksi, menggunakan metode Gray Level Co-occurrence Matrix (GLCM). Metode ini digunakan dalam berbagai ekstraksi tekstur untuk analisis citra digital. GLCM dalam analisis tekstur berdasarkan frekuensi kemunculan pasangan piksel dengan intensitas tertentu yang terpisah oleh jarak dan arah tertentu dalam suatu citra. Pasangan piksel berdasarkan arah yaitu horizontal, vertikal, dan diagonal dengan sudut 0°, 45°, 90°, 135° serta jarak tertentu antara piksel[13]. Matriks pasangan piksel merupakan fitur. Berdasarkan fitur tersebut dianalisis statistik, untuk contrast Persamaan (1) mengukur perbedaan intensitas antara piksel dan tetangganya. Energy pada Persamaan (2) menunjukkan tingkat keseragaman tekstur. Homogeneity ditunjukkan pada Persamaan (3) menunjukkan seberapa dekat elemen-elemen dalam GLCM ke diagonal utama. Sedangkan correlation Persamaan (4) menunjukkan hubungan antara piksel-piksel dengan intensitas serupa[14].

$$Contrast = \sum_{i,j=0}^{N-1} P_{ij}(i - j)^2 \tag{1}$$

$$Energy = \sum_{i,j=0}^{N-1} (P_{ij})^2 \tag{2}$$

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1 + (i - j)^2} \tag{3}$$

$$Correlation = \sum_{i,j=0}^{N-1} \frac{P_{ij}(i - \mu)(j - \mu)}{\sigma^2} \tag{4}$$

Fitur yang diperoleh diklasifikasi menggunakan Decision Trees dan Logistic Regression. Metode Decision Trees merupakan algoritma machine learning yang handal dalam membangun aturan klasifikasi berdasarkan fitur-fitur spesifik yang diekstraksi dari tekstur objek. Model ini mampu menangkap pola-pola non-linear yang mungkin terdapat dalam data, dan menghasilkan prediksi. Sedangkan Logistic Regression merupakan algoritma yang sangat efektif dalam memprediksi probabilitas suatu kejadian berdasarkan hubungan linear antara fitur-fitur dan hasil yang diinginkan seperti ditunjukkan pada Gambar 1.

Berdasarkan pseudocode Tabel 1 proses pembentukan Decision Tree dimulai dengan mengevaluasi data pada setiap langkah untuk menentukan kriteria penghentian terpenuhi. Kriteria tersebut berupa homogenitas data, tercapainya kedalaman maksimum yang telah ditetapkan, atau tidak adanya fitur yang tersisa untuk memecah data lebih lanjut. Tahapan berikutnya memilih fitur terbaik untuk memisahkan data. Pemilihan ini didasarkan pada perhitungan impurity, Gini Impurity atau Entropy, mengukur seberapa baik fitur tersebut memisahkan data ke dalam kelas yang berbeda. Fitur yang menghasilkan impurity terendah digunakan sebagai fitur terbaik dan digunakan untuk memisahkan data pada node tersebut. Pemilihan fitur terbaik selanjutnya data dipecah menjadi sub-dataset berdasarkan nilai unik dari fitur tersebut. Untuk setiap sub-dataset yang dihasilkan, algoritma memanggil fungsi pembentukan pohon secara rekursif, membentuk sub-pohon yang terhubung dengan node internal dari fitur terbaik yang dipilih. Pembentukan pohon keputusan tersebut digunakan untuk membuat prediksi pada data baru. Proses prediksi dilakukan dengan menelusuri cabang pohon mulai dari root hingga mencapai node daun, mengikuti cabang yang sesuai dengan nilai fitur dari data *input*. Node daun akan memberikan kelas yang telah diprediksi berdasarkan data pelatihan, dan itulah hasil akhir prediksi.

Tabel 1. Pseudocode Decision Tree.

Algoritma Decision Tree
<ol style="list-style-type: none"> 1. Definisikan Fungsi DecisionTree(data, depth): <ol style="list-style-type: none"> a. Tentukan parameter stopping criteria: <ul style="list-style-type: none"> - Jika semua data termasuk dalam satu kelas (homogen), atau - Jika kedalaman maksimum tercapai, atau - Jika tidak ada fitur yang tersisa untuk dipecah: <ul style="list-style-type: none"> - Kembalikan node daun (leaf) dengan kelas mayoritas dari data b. Tentukan fitur terbaik untuk memisahkan data: <ul style="list-style-type: none"> - Untuk setiap fitur: <ol style="list-style-type: none"> i. Hitung impurity (Gini Impurity atau Entropy) dari pemisahan data berdasarkan fitur tersebut ii. Pilih fitur dengan impurity terendah sebagai fitur terbaik (best_feature) c. Pisahkan data berdasarkan fitur terbaik: <ul style="list-style-type: none"> - Untuk setiap nilai unik dari best_feature: <ol style="list-style-type: none"> i. Buat sub-dataset yang hanya berisi data dengan nilai tersebut ii. Panggil fungsi DecisionTree secara rekursif untuk membentuk sub-pohon (subtree) d. Buat node internal: <ul style="list-style-type: none"> - Simpan best_feature sebagai node - Hubungkan node dengan subtree untuk setiap nilai unik dari best_feature 2. Return node sebagai root dari pohon yang dibangun 3. Prediksi (untuk data baru): <ol style="list-style-type: none"> a. Untuk setiap data baru: <ul style="list-style-type: none"> - Mulai dari root - Ikuti cabang sesuai dengan nilai fitur dalam data - Ulangi sampai mencapai node daun - Kembalikan kelas yang tertera pada node daun sebagai prediksi

Pada pseudocode Logistic Regression ditunjukkan menggunakan pendekatan statistik untuk memprediksi kemungkinan suatu data termasuk dalam kategori tertentu. Proses pelatihan Logistic Regression dimulai dengan menginisialisasi bobot dan bias, dengan nilai nol atau nilai acak kecil. Menentukan parameter-parameter learning rate (α) dan jumlah iterasi yang diperlukan untuk memperbarui bobot dan bias melalui proses optimasi. Pada setiap iterasi, model menghitung output prediksi untuk setiap data *input* menggunakan fungsi sigmoid. Fungsi sigmoid mengonversi kombinasi linear dari fitur *input* dan bobot dengan bias menjadi nilai probabilitas yang berada dalam rentang 0 hingga 1. Hasil prediksi selanjutnya model menghitung kesalahan (loss) menggunakan fungsi loss logaritmik (log-loss).

Untuk meminimalkan kesalahan dengan menghitung gradien dan melakukan perbaikan untuk bobot dan bias. Algoritma gradien descent digunakan untuk melakukan pembaruan dengan mengurangi nilai bobot dan bias berdasarkan gradien yang dihitung dan learning rate yang telah ditentukan. Proses ini berulang hingga jumlah iterasi yang ditentukan tercapai. Model Logistic Regression terlatih digunakan untuk melakukan prediksi pada data baru. Pada proses prediksi, model akan menghitung nilai probabilitas menggunakan bobot dan bias yang telah dipelajari. Untuk probabilitas di atas atau sama dengan 0,5, model

akan mengklasifikasikan data ke dalam kelas positif dan bawah 0,5 data diklasifikasikan ke dalam kelas negatif[15].

Tabel 2. Pseudocode Logistic Regression.

```

1. Inisialisasi:
- Tentukan learning rate (α)
- Tentukan jumlah iterasi (n_iterations)
- Inisialisasi bobot (W) dan bias (b) dengan nilai nol atau kecil secara acak
2. Untuk setiap iterasi hingga n_iterations:
a. Hitung output prediksi (y_pred) menggunakan fungsi sigmoid:
- Untuk setiap data point (x_i):
z = W * x_i + b
y_pred = 1 / (1 + exp(-z))
b. Hitung loss menggunakan fungsi loss logaritmik (log-loss):
loss = - (1/m) * Σ [y_i * log(y_pred) + (1 - y_i) * log(1 - y_pred)]
- Di mana m adalah jumlah data point
c. Hitung gradien untuk bobot dan bias:
- dW = (1/m) * Σ [(y_pred - y_i) * x_i]
- db = (1/m) * Σ [(y_pred - y_i)]
d. Perbarui bobot dan bias menggunakan gradien descent:
- W = W - α * dW
- b = b - α * db
3. Iterasi selesai, model Logistic Regression digunakan untuk prediksi.
4. Prediksi:
- Untuk data baru x_new:
z_new = W * x_new + b
y_pred_new = 1 / (1 + exp(-z_new))
- Tentukan kelas:
Jika y_pred_new >= 0.5, prediksi kelas positif (1)
Jika y_pred_new < 0.5, prediksi kelas negatif (0)
    
```

Hasil dari Decision Tree dan Logistic Regression di evaluasi menggunakan Confusion Matrix. Evaluasi ini memberikan gambaran menyeluruh tentang kinerja model dan area di mana model mungkin memerlukan perbaikan. Berikut ditunjukkan confusion matrik Persamaan 5 sampai dengan persamaan 8.

$$Akurasi = \frac{(TP + TN)}{(TP + FP + FN + TN)} \times 100\% \tag{5}$$

$$Presisi = \frac{TP}{(TP + FP)} \times 100\% \tag{6}$$

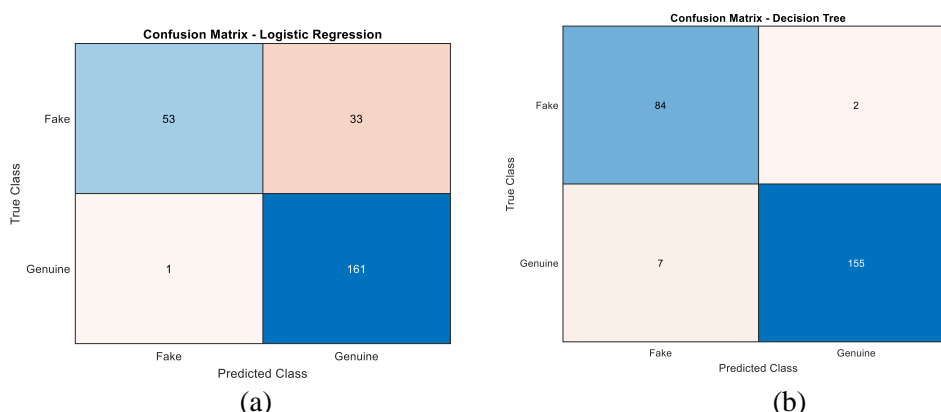
$$Recall = \frac{TP}{(TP + FN)} \times 100\% \tag{7}$$

$$F1 - Score = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{8}$$

3. Hasil dan Pembahasan

Hasil pengujian dan evaluasi dengan confusion matrix ditunjukkan pada Gambar 2(a) untuk model Logistic Regression menunjukkan bahwa dari total 162 data yang asli model tersebut berhasil mengklasifikasikan 161 di antaranya dengan benar, terdapat 1 data yang salah diklasifikasikan sebagai palsu. Sedangkan dari 86 data yang sebenarnya palsu model salah mengklasifikasikan 33 data sebagai asli dengan hanya 53 data yang benar diklasifikasikan sebagai palsu.

Sedangkan untuk model Decision Tree ditunjukkan pada Gambar 2(b) dari 162 data asli, 155 di antaranya diklasifikasikan dengan benar, dan 7 data salah diklasifikasikan sebagai palsu. Untuk 86 data palsu model ini menunjukkan performa yang lebih baik dibandingkan Logistic Regression, dengan hanya 2 data yang salah diklasifikasikan sebagai asli dan 84 data diklasifikasikan dengan benar sebagai palsu.



Gambar 2. Evaluasi Dengan Confusion Matrix (a) Logistic Regression dan (b) Decision Tree.

Perbandingan metode Logistic Regression dan Decision Tree ditunjukkan pada Tabel 3, hasil evaluasi performa model, terlihat bahwa Decision Tree menunjukkan kinerja yang lebih unggul dibandingkan Logistic Regression. Decision Tree mencapai akurasi sebesar 96.37%, yang jauh lebih tinggi dari pada akurasi Logistic Regression yang sebesar 86.29%. Dari segi presisi Logistic Regression memiliki nilai presisi yang sedikit lebih tinggi 98.15% dibandingkan dengan 92.31% Decision Tree. Untuk recall Decision Tree unggul dengan nilai 97.67%, jauh melampaui Logistic Regression yang hanya 61.63%. Hal tersebut menunjukkan bahwa Decision Tree lebih baik dalam mendeteksi kelas positif dengan benar. Sedangkan untuk F1 Score Decision Tree lebih tinggi, sebesar 94.92% dibandingkan dengan 75.71% Logistic Regression, hal ini menunjukkan bahwa Decision Tree memberikan keseimbangan yang lebih baik antara presisi dan recall.

Tabel 3. Perbandingan metode Logistic Regression dan Decision Tree.

Model	Akurasi%	Presisi%	Recall%	F1_Score%
Logistic Regression	86.29	98.15	61.63	75.71
Decision Tree	96.37	92.31	97.67	94.92

Model klasifikasi yang baik harus memiliki keseimbangan antara akurasi, presisi, recall, dan F1 score untuk memastikan bahwa model tidak hanya mengidentifikasi kelas dengan benar tetapi juga mampu mengenali positif dan negatif secara efektif.

4. Kesimpulan

Hasil pengujian yang telah dilakukan Decision Tree lebih unggul dalam identifikasi keaslian objek dengan akurasi 96.37% dan F1 score 94.92%. Meskipun Logistic Regression memiliki presisi lebih tinggi 98.15%, akurasinya lebih rendah 86.29%, membuat Decision Tree lebih efektif untuk aplikasi ini. Meskipun Logistic Regression menunjukkan presisi yang sedikit lebih tinggi, keterbatasannya dalam recall dan F1 score mengindikasikan bahwa metode ini mungkin kurang efisien dalam mengenali semua kasus positif. Dengan demikian, Decision Tree lebih direkomendasikan untuk aplikasi yang membutuhkan identifikasi keaslian objek secara andal. Pada penelitian berikutnya, model ini dapat dikembangkan pada bagian ekstraksi ciri yang tidak hanya menggunakan pendekatan tekstur namun juga menggunakan morfologi. Kombinasi tekstur dan bentuk diharapkan mampu meningkatkan performa model machine learning dalam mengenali gambar pada setiap kelasnya.

Daftar Pustaka

- [1] S. Ranjan, P. Garhwal, A. Bhan, M. Arora, and A. Mehra, "Framework for Image Forgery Detection and Classification Using Machine Learning," *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. ICICCS 2018*, no. Icoei, pp. 1872–1877, 2019, doi: 10.1109/ICCONS.2018.8663168.
- [2] D. M. Ballesteros, Y. Rodriguez, and D. Renza, "A dataset of histograms of original and fake voice recordings (H-Voice)," *Data Br.*, vol. 29, p. 105331, 2020, doi: 10.1016/j.dib.2020.105331.
- [3] F. T. Kurniati and R. R. Huizen, "Verifikasi Dokumen Cetak Menggunakan Metode Edge Detection-Glcm Dan K-Mean Clustering," *Dinamik*, vol. 25, no. 2, pp. 85–93, 2020, doi: 10.35315/dinamik.v25i2.8188.

-
- [4] A. C. Valente *et al.*, “Print defect mapping with semantic segmentation,” *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, pp. 3540–3548, 2020, doi: 10.1109/WACV45572.2020.9093470.
- [5] S. Ranjan, P. Garhwal, A. Bhan, M. Arora, and A. Mehra, “Framework for Image Forgery Detection and Classification Using Machine Learning,” *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. ICICCS 2018*, no. Iciccs, pp. 1872–1877, 2019, doi: 10.1109/ICCONS.2018.8663168.
- [6] G. Mukarambi, H. Gaikwadl, and B. V. Dhandra, “Segmentation and Text extraction from Document Images: Survey,” *2019 Innov. Power Adv. Comput. Technol. i-PACT 2019*, pp. 1–5, 2019, doi: 10.1109/i-PACT44901.2019.8960097.
- [7] R. Ghosh, C. Panda, and P. Kumar, “Handwritten Text Recognition in Bank Cheques,” *2018 Conf. Inf. Commun. Technol. CICT 2018*, 2018, doi: 10.1109/INFOCOMTECH.2018.8722420.
- [8] M. Ali, A. Sabir, and M. Hassan, “Fake audio detection using Hierarchical Representations Learning and Spectrogram Features,” *Int. Conf. Robot. Autom. Ind.*, 2021.
- [9] A. Hamid, M. Bibi, I. Siddiqi, and M. Moetesum, “Historical manuscript dating using textural measures,” *Proc. - 2018 Int. Conf. Front. Inf. Technol. FIT 2018*, pp. 235–240, 2019, doi: 10.1109/FIT.2018.00048.
- [10] J. Kozak, K. Kania, P. Juszczuk, and M. Mitrega, “Swarm intelligence goal-oriented approach to data-driven innovation in customer churn management,” *Int. J. Inf. Manage.*, vol. 60, no. May, 2021, doi: 10.1016/j.ijinfomgt.2021.102357.
- [11] M. Guo *et al.*, “A study of freeway crash risk prediction and interpretation based on risky driving behavior and traffic flow data,” *Accid. Anal. Prev.*, vol. 160, Sep. 2021, doi: 10.1016/j.aap.2021.106328.
- [12] K. Balachandar and R. Jegadeeshwaran, “Friction stir welding tool condition monitoring using vibration signals and Random forest algorithm - A Machine learning approach,” in *Materials Today: Proceedings*, 2021, vol. 46, pp. 1174–1180, doi: 10.1016/j.matpr.2021.02.061.
- [13] F. Utamingrum, S. J. A. Sarosa, C. Karim, F. Gapsari, and R. C. Wihandika, “The combination of gray level co-occurrence matrix and back propagation neural network for classifying stairs descent and floor,” *ICT Express*, vol. 8, no. 1, pp. 151–160, 2022, doi: 10.1016/j.ict.2021.05.010.
- [14] S. Saifullah and R. Drezewski, “Non-Destructive Egg Fertility Detection in Incubation Using SVM Classifier Based on GLCM Parameters,” *Procedia Comput. Sci.*, vol. 207, no. Kes, pp. 3248–3257, 2022, doi: 10.1016/j.procs.2022.09.383.
- [15] X. Zhao, H. Yang, Y. Yao, H. Qi, M. Guo, and Y. Su, “Factors affecting traffic risks on bridge sections of freeways based on partial dependence plots,” *Phys. A Stat. Mech. its Appl.*, vol. 598, Jul. 2022, doi: 10.1016/j.physa.2022.127343.