

OPTIMIZATION OF MACHINE LEARNING MODEL ACCURACY FOR BRAIN TUMOR CLASSIFICATION WITH PRINCIPAL COMPONENT ANALYSIS

Indra Maulana¹, Amril Mutoi Siregar^{*2}, Rahmat³, Ahmad Fauzi⁴

^{1,2,3,4}Informatic Departement, Faculty Of Computer Science, Universitas Buana Perjuangan Karawang, Indonesia
Email: ¹if20.indramaulana@mhs.ubpkarawang.ac.id, ²amrilmutoi@ubpkarawang.ac.id,
³rahmat@ubpkarawang.ac.id, ⁴ahmadfauzi@ubpkarawang.ac.id

(Article received: May 07, 2024; Revision: May 30, 2024; published: June 11, 2024)

Abstract

The main issue in brain tumor classification is the accuracy and speed of diagnosis through medical imaging. This study aims to improve the accuracy of machine learning models for brain tumor classification by using Principal Component Analysis (PCA) for dimensionality reduction. The research methods include image preprocessing, feature scaling, PCA application, and the implementation of machine learning algorithms such as Logistic Regression, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Naive Bayes. The dataset consists of 3,264 images divided into training and testing sets. The results show that the use of PCA has varying impacts on different algorithms. PCA increases the accuracy of the SVM algorithm from 81% to 83% and KNN from 68% to 71%, but decreases the accuracy of Logistic Regression from 77% to 69% and Naive Bayes from 49% to 42%. Evaluation is performed using the Confusion Matrix and AUC-ROC to measure model performance. In conclusion, selecting the appropriate algorithm and preprocessing method is crucial in medical image classification, and the use of PCA should be considered based on the characteristics of the data and the algorithms used. This study also encourages the exploration of alternative dimensionality reduction methods for medical image analysis.

Keywords: Brain tumor classification, Component Analysis, KNN, Machine learning, Medical image, Naive Bayes, Principal, Random Forest, SVM.

OPTIMASI AKURASI MODEL PEMBELAJARAN MESIN UNTUK KLASIFIKASI TUMOR OTAK DENGAN ANALISIS KOMPONEN UTAMA

Abstrak

Masalah utama dalam klasifikasi tumor otak adalah ketepatan dan kecepatan diagnosis menggunakan citra medis. Penelitian ini bertujuan meningkatkan akurasi model pembelajaran mesin untuk klasifikasi tumor otak dengan menggunakan Principal Component Analysis (PCA) untuk reduksi dimensi. Metode penelitian meliputi pre-processing gambar, scaling fitur, penerapan PCA, dan implementasi algoritma pembelajaran mesin seperti Logistic Regression, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Naive Bayes. Dataset terdiri dari 3264 gambar yang dibagi menjadi set pelatihan dan pengujian. Hasil menunjukkan bahwa penggunaan PCA memiliki dampak berbeda pada setiap algoritma. PCA meningkatkan akurasi SVM dari 81% menjadi 83% dan KNN dari 68% menjadi 71%, tetapi menurunkan akurasi Logistic Regression dari 77% menjadi 69% dan Naive Bayes dari 49% menjadi 42%. Evaluasi dilakukan menggunakan Confusion Matrix dan AUC-ROC untuk mengukur kinerja model. Kesimpulannya, pemilihan algoritma dan metode preprocessing yang tepat sangat penting dalam klasifikasi citra medis, dan penggunaan PCA harus dipertimbangkan berdasarkan karakteristik data dan algoritma yang digunakan. Penelitian ini juga mendorong eksplorasi metode reduksi dimensi alternatif untuk analisis citra medis.

Kata kunci: Citra medis, Klasifikasi tumor otak, KNN, Naive Bayes, Pembelajaran mesin, Principal Component Analysis, Random Forest, SVM.

1. PENDAHULUAN

Tumor adalah massa jaringan ekstra seluler yang mengarah pada pembentukan tumor. Mungkin

ada tumor jinak atau ganas, meskipun tumor ganas dapat menyebabkan kanker, tumor jinak tidak [1].

Dalam Otak atau bagian lain dari sistem saraf pusat (SSP) seperti sumsum tulang belakang atau

saraf kranial, tumor dapat mengalami pertumbuhan . Fungsi Kesadaran, gerakan, persepsi sensorik, kognisi, bicara, memori dan sebagian aktivitas tubuh adalah semua diawasi oleh fungsi kerja otak. Tumor dapat mengganggu kemampuan otak untuk melakukan tugas tersebut [2]

Sebagian besar tumor sistem saraf pusat (SSP) adalah tumor otak yang merupakan 85% hingga 90% dari semua tumor SSP. Peluang menemukan tumor jenis ini dalam kehidupan seseorang kurang dari 1 persen. Sekitar 308.102 orang di dunia menerima diagnosis tumor primer di otak atau sumsum tulang belakang pada 2020 [3].

Tanda-tanda tumor otak sering kali diabaikan. Deteksi tumor otak dapat dilakukan sebagian dengan menggunakan pendekatan anatomi pada pencitraan medis. Sebagai contoh, pertimbangkan *Computed Tomography scan* (CT scan) dan *Magnetic Resonance Imaging* (MRI). Jaringan lunak otak tidak dapat dideteksi dengan CT scan. CT scan cocok untuk melihat struktur tulang. MRI lebih baik dalam menunjukkan perbedaan antara jaringan lunak dan keras otak karena memberikan informasi citra yang lebih dalam. Dokter dapat menggunakan hasil MRI untuk mendapatkan informasi penting untuk evaluasi dan diagnosis.[4]

Metode yang diusulkan bertujuan untuk mengatasi keterbatasan teknik konvensional, seperti konsumsi waktu dan tingkat kesalahan, dengan menyediakan klasifikasi tumor otak yang lebih kuat dan akurat. Metode ini menangani masalah kritis dalam mendiagnosis tumor otak secara akurat melalui teknik pencitraan, menyoroiti kompleksitas yang melekat dan potensi kesalahan dalam metode tradisional.

Segi kuantitas yang diteliti dari penelitian ini adalah jumlah komponen utama yang dipertahankan dalam PCA, ukuran gambar yang digunakan untuk pelatihan model, serta akurasi, sensitivitas, dan spesifisitas hasil klasifikasi dari berbagai algoritma pembelajaran mesin yang digunakan.

Penelitian oleh [5] membahas klasifikasi tumor otak menggunakan Support Vector Machines (SVM) dengan koefisien wavelet haar dan db4 sebagai vektor fitur, ekstraksi fitur menggunakan Level-3 Discrete Wavelet Transform (DWT), reduksi data dengan PCA, serta algoritma evolusioner seperti Genetic Algorithms (GA) untuk segmentasi tumor otak, mencapai akurasi 99,38% menggunakan kernel GRB. [6] menggunakan transformasi wavelet dan analisis tekstur, PCA untuk reduksi dimensi, dan algoritma seperti SVM, CNN, serta Random Forests untuk klasifikasi, mencapai akurasi 99,38% dengan kernel GRB. [7] menggunakan segmentasi superpixel, ekstraksi fitur dengan PCA, dan clustering menggunakan K-means berbasis template untuk deteksi tumor otak, dengan akurasi 99,38%. [8] menggunakan arsitektur EfficientNet dan model UNet, mencapai akurasi 99,5% untuk klasifikasi dan 96% untuk segmentasi. [9] menggunakan GLCM dan

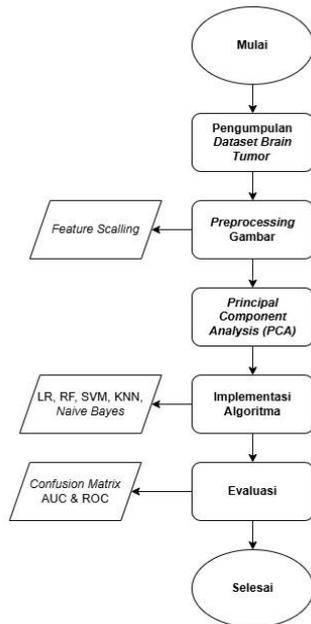
SVM, menunjukkan akurasi klasifikasi 96,875%. [10] menggunakan DWT dan PCA, mencapai akurasi 96,83%. [11] menggunakan fitur entropi, morfologis, SIFT, dan tekstur serta Naive Bayes, mencapai akurasi 100%. Penelitian ini bertujuan meningkatkan akurasi model pembelajaran mesin dalam klasifikasi tumor otak menggunakan PCA untuk reduksi dimensi dan berbagai algoritma pembelajaran mesin seperti Logistic Regression, Random Forest, SVM, KNN, dan Naive Bayes.

Penelitian ini bertujuan untuk meningkatkan akurasi model pembelajaran mesin dalam klasifikasi tumor otak menggunakan citra MRI. Penelitian ini memiliki keunikan dengan menggabungkan beberapa teknik terbaik dari penelitian sebelumnya serta menerapkan proses pre-processing dan pengurangan dimensi yang lebih terperinci dan terstruktur. Dalam penelitian ini, digunakan kombinasi PCA untuk pengurangan dimensi dan berbagai algoritma pembelajaran mesin seperti Logistic Regression, Random Forest, SVM, KNN, dan Naive Bayes untuk klasifikasi.

Selain itu, penelitian ini memanfaatkan teknik standarisasi data menggunakan StandardScaler dan mengubah ukuran gambar menjadi 200x200 piksel untuk memastikan konsistensi dan efisiensi dalam pelatihan model[12], [13]. Penggunaan ukuran gambar yang lebih kecil ini bertujuan untuk mengurangi beban komputasi sambil tetap mempertahankan detail penting dalam gambar[14]. Dengan demikian, penelitian ini diharapkan dapat memberikan wawasan tentang metode yang paling efektif untuk klasifikasi tumor otak serta mengeksplorasi pengaruh teknik preprocessing dan pemilihan algoritma terhadap hasil klasifikasi[15]. Tidak seperti penelitian sebelumnya yang hanya berfokus pada satu atau beberapa aspek tertentu, penelitian ini menawarkan pendekatan yang lebih komprehensif dengan mengintegrasikan teknik preprocessing yang lebih canggih dan analisis reduksi dimensi yang lebih detail, sehingga dapat memberikan kontribusi yang signifikan terhadap peningkatan akurasi dan efisiensi dalam klasifikasi tumor otak[16].

2. METODE PENELITIAN

Proses penelitian dimulai dengan mengkaji literatur. Tujuan dari kajian literatur ini adalah untuk menemukan dasar teoritis yang relevan dan mencari referensi ilmiah yang mendukung penelitian. Selama penelitian, langkah-langkah atau tahapan dapat dilihat pada Gambar 1. Gambar 1 menggambarkan alur penelitian untuk klasifikasi tumor otak, yang mencakup pengumpulan *dataset*, *Pre-Processing* gambar, *feature scaling*, penerapan *Principal Component Analysis* (PCA), implementasi berbagai algoritma pembelajaran mesin, dan evaluasi menggunakan *Confusion Matrix*, AUC dan ROC.



Gambar 1 Flowchart Tahapan Penelitian.

2.1. Dataset

Dataset penelitian ini diperoleh dari kaggle, sebuah situs web yang tersedia untuk umum [17]. Dataset ini terdiri dari 3264 gambar yang menggambarkan berbagai jenis tumor otak, termasuk glioma_tumor (926 gambar), meningioma_tumor (937 gambar), tidak ada tumor (500 gambar), dan hipofisis_tumor (901 gambar). Dataset ini telah dipisahkan menjadi dua kategori untuk tujuan pelatihan dan pengujian. Set pelatihan mencakup 2.870 gambar, sedangkan set pengujian mencakup 394 gambar. Penelitian ini menggunakan lima teknik pembelajaran mesin: *Logistics Regression* (LR), *Random Forest* (RF), *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Naive Bayes* [18].

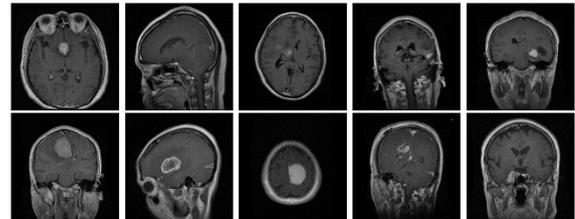
2.2. Pre-Processing

Sebelum pelatihan dan evaluasi, data menjalani fase pre-processing. Tahap ini melibatkan perubahan ukuran dan transformasi gambar menjadi vektor yang kemudian diskalakan untuk proses pelatihan.

2.2.1. Data Transformation

Langkah awal dalam transformasi data untuk klasifikasi tumor otak adalah menetapkan jalur ke dataset pelatihan dan pengujian serta menyusun daftar kategori yang mewakili berbagai jenis tumor otak. Kategori ini diambil dari subdirektori dalam jalur pelatihan. Setelah itu, dibuat kamus untuk memetakan setiap jenis tumor ke label integer, yang akan memudahkan pengolahan dan pelatihan model. Langkah berikutnya adalah memilih secara acak dan menampilkan beberapa gambar dari kategori tumor untuk memverifikasi bahwa data telah dibaca dengan

benar, sebagaimana ditunjukkan pada Gambar 2 yang menampilkan gambar secara acak dari setiap kategori tumor untuk memberikan gambaran umum tentang kumpulan data. Proses ini memastikan bahwa data diorganisasi dan dilabeli dengan benar sebelum digunakan dalam pelatihan model klasifikasi, sehingga memudahkan model untuk mengenali dan mengklasifikasikan gambar berdasarkan jenis tumornya.



Gambar 2 Gambar acak dari setiap kategori tumor

2.2.2. Resize Data

Proses mengubah ukuran gambar dilakukan dengan menggunakan fungsi `cv2.resize` dari OpenCV, di mana setiap gambar dibaca dalam mode grayscale menggunakan `cv2.imread` dan kemudian diubah ukurannya menjadi 200x200 piksel [19]. Mengubah ukuran gambar menjadi konsisten sangat penting untuk memastikan bahwa semua gambar dalam dataset memiliki dimensi yang sama, yang diperlukan untuk pelatihan model yang efisien dan konsisten [20]. Ukuran 200x200 dipilih untuk mengurangi dimensi gambar, yang mengurangi jumlah fitur yang harus diproses oleh model, sehingga meningkatkan efisiensi dan kecepatan pelatihan. Selain itu, ukuran ini cukup besar untuk mempertahankan detail penting dalam gambar, namun cukup kecil untuk meminimalkan beban komputasi [21].

2.2.3. Data Reshape and data Split

Langkah selanjutnya melibatkan pembagian data dan perubahan bentuk data. Pertama, data gambar yang telah diubah menjadi vektor satu dimensi dibagi menjadi set pelatihan dan pengujian menggunakan metode `train_test_split`, dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian. Pembagian ini memastikan bahwa model dapat dilatih pada sebagian data dan diuji pada data yang terpisah, sehingga menghindari overfitting dan memungkinkan evaluasi kinerja model secara objektif [22], [23]. Setelah pembagian, data pelatihan dan pengujian diubah bentuknya kembali untuk memastikan konsistensi bentuk data yang digunakan dalam model. Meskipun tampak redundan, perubahan bentuk ini memberikan verifikasi tambahan bahwa data memiliki struktur yang tepat untuk diproses oleh algoritma pembelajaran mesin. Langkah-langkah ini penting dalam praproses data untuk memastikan data siap digunakan dalam pelatihan dan pengujian model dengan distribusi yang

baik dan bentuk yang konsisten[24], [25]. Untuk melihat hasil data yang telah kita proses reshape bisa kita lihat keterangan seperti pada gambar 3 dibawah.

(3264, 200, 200)(3264, 40000)

Gambar 3 Keterangan data sebelum reshape (kiri) dan sesudah reshape (kanan)

2.2.4. Feature Scalling

Penskalaan fitur, juga disebut sebagai standarisasi, merupakan tahap penting dalam pre-processing data[26]. Tujuannya adalah untuk menstandarkan data dalam rentang tertentu, sehingga memungkinkan perhitungan langsung dalam algoritma. Selain itu, penskalaan fitur merupakan hal yang wajib dilakukan untuk eksperimen yang menggunakan Keras, Scikit-learn, dan deep learning. Dataset yang diimplementasikan dalam eksperimen ini mencakup variabel yang mengadopsi skala yang berbeda. Oleh karena itu, dataset mengalami penskalaan fitur untuk mengubah vektor fitur ke dalam format yang lebih sesuai. Dalam studi ini, Min-Maxscaler digunakan untuk mengonversi nilai fitur ke dalam kisaran tertentu. Langkah selanjutnya adalah penskalaan fitur, yang berarti mengubah nilai piksel gambar dari rentang [0, 255] menjadi [0, 1] dengan membagi setiap nilai piksel dengan 255. Ini penting untuk memastikan bahwa semua fitur memiliki skala yang sama, sehingga algoritma pembelajaran mesin dapat bekerja lebih efisien dan menemukan pola dengan lebih baik. Normalisasi ini juga mencegah nilai piksel yang besar mendominasi proses pelatihan, yang bisa menyebabkan model tidak bekerja dengan baik. Setelah penskalaan, kita memeriksa nilai maksimum dan minimum untuk memastikan semua nilai berada dalam rentang yang diharapkan. Dengan melakukan penskalaan fitur, data menjadi lebih seragam dan siap untuk digunakan dalam pelatihan model, yang membantu meningkatkan akurasi dan kinerja model pembelajaran mesin [27].

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

- x' adalah nilai asli dari sebuah fitur
- x adalah nilai asli dari fitur
- x_{min} adalah nilai minimum dari fitur tersebut dalam dataset.
- x_{max} adalah nilai maksimum dari fitur tersebut dalam dataset.

2.4. Principal Component analysis (PCA)

Principal Component Analysis (PCA) adalah teknik statistik yang menyederhanakan kumpulan data dengan tetap mempertahankan kovarian data [28]. Teknik ini memungkinkan visualisasi hasil dalam bentuk diagram pencar dengan sedikit kehilangan informasi [29]. Teknik ini banyak digunakan dalam genetika populasi dan bidang terkait

dengan paket-paket yang banyak dikutip seperti EIGENSOFT [30] dan PLINK [31] yang umum digunakan. Hasil PCA menginformasikan desain studi, memungkinkan identifikasi dan karakterisasi individu dan populasi, dan memfasilitasi penarikan kesimpulan historis dan etnobiologi mengenai asal-usul evolusi penyebaran dan keterkaitan ([32]).

Pada penelitian ini, beberapa variabel penting digunakan untuk memproses dan mereduksi dimensi data. Scaler adalah instance StandardScaler yang menstandarkan data pelatihan (X_train_scaled) dan data uji (X_test_scaled), memastikan setiap fitur memiliki mean 0 dan standar deviasi. n_components menetapkan jumlah komponen utama yang dipertahankan, diatur ke 150 dalam contoh ini. PCA adalah instance PCA yang digunakan untuk mereduksi dimensi data, menghasilkan X_train_pca dan X_test_pca, yaitu data pelatihan dan uji yang telah direduksi dimensinya [28]. explained_variance_ratio menyimpan rasio varians yang dijelaskan oleh setiap komponen utama, sementara cumulative_variance menghitung jumlah kumulatif dari rasio varians tersebut [29]. Visualisasi dari explained_variance_ratio dan cumulative_variance membantu memahami berapa banyak varians total yang dijelaskan oleh sejumlah komponen utama. Variabel-variabel ini memungkinkan pemrosesan data yang efektif dan pengurangan dimensi yang efisien, yang penting untuk analisis lebih lanjut atau penerapan model pembelajaran mesin.

2.5. Logistic Regression (LR)

Menurut [33] Model *Logistic Regression* adalah kerangka kerja serbaguna dan fleksibel yang dapat memenuhi kebutuhan banyak analisis. Namun, penting untuk dicatat bahwa model ini mungkin tidak cocok untuk semua situasi. Dalam bidang teknik dan sains, banyak masalah yang melibatkan variabel respon dan variabel prediktor yang berhubungan melalui fungsi *nonlinier* yang diketahui. Dalam penelitian ini, pertama-tama kami mengurangi dimensi menggunakan PCA dan kemudian menjalankan regresi linier untuk memprediksi data uji. Kami menghitung dan mencetak hasil akurasi, matriks kebingungan, dan laporan klasifikasi.

Rumus untuk Regresi Logistik Fungsi Sigmoid

$$h\sigma(x) = \frac{1}{1 + e^{-nz}} \quad (2)$$

- $h\sigma(x)$ adalah fungsi logistik yang mengambil nilai antara 0 dan 1

Berikut adalah *Pseudocode* untuk tahapan yang dilakukan oleh Logistic Regression:

```
BEGIN
INITIALIZE names AS LIST
INITIALIZE accuracies AS LIST
INITIALIZE finalResults AS LIST
```

```

INITIALIZE X_train_pca, y_train, X_test_pca,
y_test

SET lg TO LogisticRegression WITH
PARAMETER C = 0.1
CALL lg.fit WITH ARGUMENTS X_train_pca,
y_train

SET y_pred TO lg.predict WITH ARGUMENT
X_test_pca

SET accuracy TO CALL accuracy_score WITH
ARGUMENTS y_test, y_pred
SET CM TO CALL confusion_matrix WITH
ARGUMENTS y_test, y_pred

IF accuracy >= 0.5 THEN
SET ClassificationReport TO CALL
classification_report WITH ARGUMENTS y_test,
y_pred

SET name TO "Logistic Regression"
CALL names.append WITH ARGUMENT
name
CALL accuracies.append WITH
ARGUMENT accuracy
CALL finalResults.append WITH
ARGUMENT (name, accuracy)

VAR accuracyOutput = "Accuracy: " +
(accuracy * 100) + "%"
VAR classificationReportOutput =
"Classification Report is: " +
ClassificationReport

PRINT accuracyOutput
PRINT classificationReportOutput
ELSE

VAR belowThresholdMessage = "Model
accuracy is below acceptable threshold."
PRINT belowThresholdMessage
ENDIF
END

```

2.6. Random Forest (RF)

Salah satu teknik yang digunakan untuk regresi dan klasifikasi adalah pengklasifikasi *Random Forest* (RF). Ini dapat dilihat sebagai berasal dari pohon keputusan tunggal atau dari kumpulan pohon keputusan. Yang perlu diperhatikan adalah kemampuannya untuk menghasilkan prediksi kategorikal dengan probabilitas keluaran yang dapat diubah-ubah dan berbagai nilai potensial. *Overfitting* harus dihindari dengan cara apa pun, terutama ketika berhadapan dengan set data yang kecil. Kemampuan algoritma *Random Forest* untuk mengidentifikasi data dengan karakteristik yang hilang adalah salah satu keunggulannya. Penggunaan utama metode pengklasifikasi *Random Forest* adalah dalam

manajemen data sampel besar dan klasifikasi data. Regresi tidak cocok untuk itu. Untuk mempersiapkan data pelatihan dan pengujian (untuk model), memasukkan data churn pelanggan telekomunikasi, menjalankan visualisasi data, dan meramalkan *Random Forest*, algoritma ini melewati banyak fase [34]

Berikut adalah *Pseudocode* untuk tahapan yang dilakukan oleh Algoritma *Random Forest*:

```

BEGIN
INITIALIZE names AS LIST
INITIALIZE accuracies AS LIST
INITIALIZE finalResults AS LIST
INITIALIZE X_train_pca, y_train, X_test_pca,
y_test

SET rf TO RandomForestClassifier WITH
PARAMETERS n_estimators = 10, random_state
= 42
CALL rf.fit WITH ARGUMENTS X_train_pca,
y_train

SET y_pred TO CALL rf.predict WITH
ARGUMENT X_test_pca

SET accuracy TO CALL accuracy_score WITH
ARGUMENTS y_test, y_pred
SET CM_rf TO CALL confusion_matrix WITH
ARGUMENTS y_test, y_pred

IF accuracy >= 0.5 THEN
SET ClassificationReport TO CALL
classification_report WITH ARGUMENTS y_test,
y_pred

SET name TO "Random Forest"
CALL names.append WITH ARGUMENT name
CALL accuracies.append WITH ARGUMENT
accuracy
CALL finalResults.append WITH ARGUMENT
(name, accuracy)

SET accuracyOutput TO "Accuracy: " +
(accuracy * 100) + "%"
SET classificationReportOutput TO
"Classification Report is: " +
ClassificationReport

PRINT accuracyOutput
PRINT classificationReportOutput
ELSE
SET belowThresholdMessage TO "Model
accuracy is below acceptable threshold."
PRINT belowThresholdMessage
ENDIF
END

```

2.7. Support Vector Machine (SVM)

SVM adalah pengklasifikasi margin maksimum yang didefinisikan sebagai masalah optimasi seperti yang diberikan di bawah ini. Dengan vektor fitur x_i , vektor bobot w , dan label kelas y , SVM dapat dengan mudah diperluas ke masalah multi-kelas. Dalam praktiknya, pendekatan umum untuk klasifikasi multi-kelas adalah metode satu-lawan-semua dan metode satu-lawan-satu. Untuk tugas klasifikasi c -kelas, pendekatan pertama membutuhkan c pengklasifikasi SVM biner. Persyaratannya adalah $c(c-1)/2$ pengklasifikasi biner, ketika pendekatan yang terakhir diikuti. Penelitian yang diusulkan menggunakan model pengklasifikasi 3-kelas dengan tiga pengklasifikasi biner SVM, mengikuti pendekatan satu-lawan-semua [35]

Rumus untuk prediksi SVM.

$$f(x) = \text{sign} \left(\sum_{i=1}^n \alpha_i y_i K(x_i, x) + b \right) \quad (3)$$

- $f(x)$: Fungsi keputusan SVM yang digunakan untuk menentukan kelas dari data input x
- α_i : Koefisien Lagrange yang diperoleh dari solusi masalah optimasi SVM. Koefisien ini non-zero hanya untuk support vectors.
- y_i : Label kelas dari support vector x_i yang bernilai +1 atau -1 dalam kasus klasifikasi biner.
- $K(x_i, x)$: Fungsi kernel yang mengukur kesamaan antara dua vektor x_i dan x Fungsi kernel ini memungkinkan SVM untuk bekerja dalam ruang fitur berdimensi tinggi tanpa harus secara eksplisit menghitung koordinat dalam ruang tersebut.
- b : Bias dari fungsi keputusan.
- sign Fungsi tanda yang menentukan apakah hasil dari ekspresi di dalamnya positif atau negatif, yang akhirnya menentukan kelas dari x [36]

Berikut adalah *Pseudocode* untuk tahapan yang dilakukan oleh *Support Vector Machine* (SVM):

```
BEGIN
INITIALIZE names AS LIST
INITIALIZE accuracies AS LIST
INITIALIZE finalResults AS LIST
INITIALIZE X_train_pca, y_train, X_test_pca,
y_test

SET svm TO SVC WITH PARAMETER kernel =
"rbf"
CALL svm.fit WITH ARGUMENTS X_train_pca,
y_train

SET y_pred TO CALL svm.predict WITH
ARGUMENT X_test_pca

SET accuracy TO CALL accuracy_score WITH
ARGUMENTS y_test, y_pred
SET CM_svm TO CALL confusion_matrix WITH
ARGUMENTS y_test, y_pred
```

```
IF accuracy >= 0.5 THEN
SET ClassificationReport TO CALL
classification_report WITH ARGUMENTS y_test,
y_pred

SET name TO "SVM"
CALL names.append WITH ARGUMENT name
CALL accuracies.append WITH ARGUMENT
accuracy
CALL finalResults.append WITH ARGUMENT
(name, accuracy)

SET accuracyOutput TO "Accuracy: " +
(accuracy * 100) + "%"
SET classificationReportOutput TO
"Classification Report is: " + ClassificationReport

PRINT accuracyOutput
PRINT classificationReportOutput
ELSE
SET belowThresholdMessage TO "Model
accuracy is below acceptable threshold."
PRINT belowThresholdMessage
ENDIF
END
```

2.8. K-Nearest Neighbors (KNN)

K-Nearest Neighbor (KNN) adalah algoritma pembelajaran yang diawasi di mana hasil dari contoh yang baru diklasifikasikan didasarkan pada mayoritas kategori K -tetangga terdekat. Tujuan dari algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel dari data pelatihan. Algoritma *KNearest Neighbor* menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari nilai *instance* baru. Pengklasifikasi tidak menggunakan model apa pun untuk dicocokkan dan hanya didasarkan pada memori. Algoritma metode KNN bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan KNN [37].

Berikut adalah *Pseudocode* untuk tahapan yang dilakukan oleh Algoritma *K-Nearest Neighbors*:

```
BEGIN
INITIALIZE names AS LIST
INITIALIZE accuracies AS LIST
INITIALIZE finalResults AS LIST
INITIALIZE X_train_pca, y_train, X_test_pca,
y_test

SET knn TO KNeighborsClassifier WITH
PARAMETER n_neighbors = 10
CALL knn.fit WITH ARGUMENTS X_train_pca,
y_train

SET y_pred TO CALL knn.predict WITH
ARGUMENT X_test_pca
```

```

SET accuracy TO CALL accuracy_score WITH
ARGUMENTS y_test, y_pred
SET CM_knn TO CALL confusion_matrix WITH
ARGUMENTS y_test, y_pred

IF accuracy >= 0.5 THEN
    SET ClassificationReport TO CALL
    classification_report WITH ARGUMENTS y_test,
    y_pred

    SET name TO "KNN"
    CALL names.append WITH ARGUMENT name
    CALL accuracies.append WITH ARGUMENT
    accuracy
    CALL finalResults.append WITH ARGUMENT
    (name, accuracy)

    SET accuracyOutput TO "Accuracy: " +
    (accuracy * 100) + "%"
    SET classificationReportOutput TO
    "Classification Report is: " +
    ClassificationReport

    PRINT accuracyOutput
    PRINT classificationReportOutput
ELSE
    SET belowThresholdMessage TO "Model
    accuracy is below acceptable threshold."
    PRINT belowThresholdMessage
ENDIF
END
    
```

2.9. Naïve Bayes

Klasifikasi Naïve Bayes adalah klasifikasi pembelajaran mesin yang diawasi, berdasarkan pendekatan probabilistik yang menggunakan teorema probabilitas Bayes. Algoritma Naïve Bayes disebut “naïf” karena algoritma ini membuat asumsi bahwa kemunculan fitur-fitur tidak bergantung satu sama lain. Itulah alasan utama mengapa kami menggunakan algoritma ini untuk mendeteksi tumor otak dari lokasi yang berbeda dengan jenis fitur yang berbeda. Matriks fitur yang diekstraksi akan dilatih dalam pengklasifikasi Naïve Bayes sehingga dapat memprediksi gambar uji apakah itu normal atau tumor [38].

Berikut adalah *Pseudocode* untuk tahapan yang dilakukan oleh Algoritma *Naive Bayes*:

```

BEGIN
INITIALIZE names AS LIST
INITIALIZE accuracies AS LIST
INITIALIZE finalResults AS LIST
INITIALIZE X_train_pca, y_train, X_test_pca
y_test

SET nv TO GaussianNB
CALL nv.fit WITH ARGUMENTS X_train_pca,
y_train
    
```

```

SET y_pred TO CALL nv.predict WITH
ARGUMENT X_test_pca

SET accuracy TO CALL accuracy_score WITH
ARGUMENTS y_test, y_pred
SET CM_nv TO CALL confusion_matrix WITH
ARGUMENTS y_test, y_pred

IF accuracy >= 0.5 THEN
    SET ClassificationReport TO CALL
    classification_report WITH ARGUMENTS y_test,
    y_pred

    SET name TO "NaiveBayes"
    CALL names.append WITH ARGUMENT name
    CALL accuracies.append WITH ARGUMENT
    accuracy
    CALL finalResults.append WITH ARGUMENT
    (name, accuracy)

    SET accuracyOutput TO "Accuracy: " +
    (accuracy * 100) + "%"
    SET classificationReportOutput TO
    "Classification Report is: " +
    ClassificationReport

    PRINT accuracyOutput
    PRINT classificationReportOutput
ELSE
    SET belowThresholdMessage TO "Model
    accuracy is below acceptable threshold."
    PRINT belowThresholdMessage
ENDIF
END
    
```

2.10. Confusion Matrix

Confusion Matrix mengevaluasi kinerja metode klasifikasi dan mencakup empat istilah: true positive (TP), true negative (TN), false positive (FP), dan false negative (FN) dari hasil klasifikasi. TN mewakili nilai negatif yang diidentifikasi dengan benar, sedangkan FP mewakili nilai negatif yang salah diidentifikasi sebagai positif [39]. Model ini digunakan untuk menghitung nilai akurasi, presisi, *recall*, dan F1-score. Presisi mencerminkan proporsi nilai positif yang diprediksi dengan benar terhadap total nilai positif yang diprediksi, sedangkan akurasi mengukur keseluruhan keseluruhan kinerja model [40]. Berikut adalah contoh tabel *Confusion matrix*:

Tabel 1 Contoh Tabel *Confusion Matrix*

	Prediksi Positif	Prediksi Negatif
Aktual Positif	TP	FN
Aktual Negatif	FP	TN

Berikut adalah rumus untuk Evaluasi Model pada confusion matrix:

Evaluasi Model

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi yang benar}}{\text{Jumlah Total Data}} \tag{4}$$

Confusion Matrix

$$\begin{pmatrix} \text{True Negative (TN)} & \text{False Positive (FP)} \\ \text{False Negative (FN)} & \text{True Positive (TP)} \end{pmatrix} \quad (5)$$

2.11. Area Under Curve (AUC) & Receiver Operating Characteristic (ROC)

Dalam metode penelitian ini, kami menggunakan kurva ROC (Receiver Operating Characteristic) dan AUC (Area Under the Curve) untuk mengevaluasi kinerja model klasifikasi biner. Kurva ROC adalah grafik yang menggambarkan kinerja model dengan memplot True Positive Rate (TPR) terhadap False Positive Rate (FPR) pada berbagai threshold. TPR, yang juga dikenal sebagai sensitivitas, adalah rasio antara jumlah prediksi benar positif dengan total jumlah aktual positif, sementara FPR adalah rasio antara jumlah prediksi salah positif dengan total jumlah aktual [41]

AUC adalah area di bawah kurva ROC dan digunakan untuk memberikan ukuran kinerja model klasifikasi. Nilai AUC berkisar antara 0 hingga 1, di mana nilai 0,5 menunjukkan bahwa model tidak lebih baik dari penebakan acak, sementara nilai mendekati 1 menunjukkan bahwa model sangat baik dalam membedakan antara kelas positif dan negatif [42] Rumus sederhana untuk menghitung AUC adalah sebagai berikut:

$$f_{baud} = \frac{2^{SMOD}}{64} \times f_{osc} \quad (6)$$

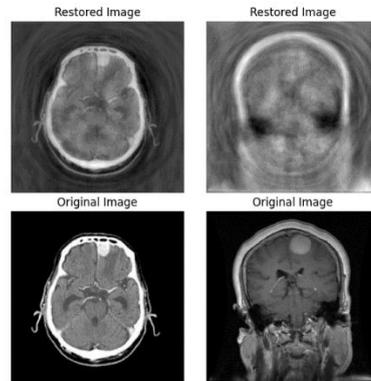
- TPR (True Positive Rate) adalah rasio antara jumlah prediksi benar positif dengan total jumlah aktual positif (juga dikenal sebagai sensitivitas atau recall).
- FPR (False Positive Rate) adalah rasio antara jumlah prediksi salah positif dengan total jumlah aktual negatif.
- n adalah jumlah titik pada kurva ROC yang dihasilkan dari berbagai threshold.

Di mana n adalah jumlah titik pada kurva ROC, TPR adalah True Positive Rate, dan FPR adalah False Positive Rate [43].

3. HASIL DAN PEMBAHASAN

Setelah melalui tahap *pre-processing* hingga ke tahap penggunaan PCA penelitian ini dilanjutkan dengan mengimplementasikan berbagai algoritma pembelajaran mesin. Algoritma-algoritma ini diuji pada set pelatihan dan dievaluasi pada set pengujian. Hasil dari eksperimen ini tidak hanya akan memberikan wawasan tentang kinerja relatif masing-masing algoritma dalam konteks klasifikasi tumor otak berdasarkan citra MRI, tetapi juga akan menampilkan hasil akurasi masing-masing algoritma. Selain itu, penelitian ini juga akan menampilkan gambar confusion matrix untuk memberikan pemahaman visual yang lebih mendalam mengenai kemampuan masing-masing algoritma dalam

mengklasifikasikan jenis tumor otak. Keseluruhan analisis ini bertujuan untuk memberikan informasi yang komprehensif dan terperinci mengenai keberhasilan algoritma *Machine Learning* dalam menangani tugas klasifikasi citra medis dalam konteks spesifik ini.



Gambar 4 Perbandingan gambar original dan sesudah PCA

Gambar asli sebelum PCA dan setelah PCA sangat jelas perbedaannya, seperti yang ditunjukkan pada gambar 3. Bagian atas gambar menunjukkan gambar asli yang di rekonstruksi kembali ke dimensi aslinya setelah PCA, dan bagian bawah menunjukkan gambar asli sebelum mereka mengalami reduksi dimensi melalui PCA. Perhatikan bagaimana representasi visual mereka telah di ubah oleh proses PCA.

3.1. Logistic Regression

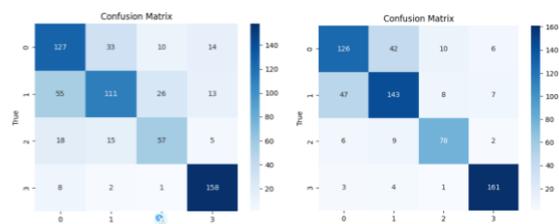
Algoritma *Logistic Regression* mengalami penurunan performa ketika diuji dengan menggunakan PCA, seperti yang ditunjukkan pada Tabel 2 di bawah ini.

Table 2 Perbandingan akurasi algoritma *Logistic Regression* dengan dan tanpa PCA

Method	Accuracy	Precision	Recall	F1-Score
LR	69%	69%	69%	69%
LR-NPA	77%	78%	78%	78%

Hasil di atas menunjukkan bahwa penggunaan PCA mengakibatkan penurunan akurasi algoritma Regresi Logistik, terlihat dari perbandingan antara LR (dengan PCA) dan LR-NPA (tanpa PCA).

Perbedaan hasil pengujian dapat dilihat pada confusion matrix seperti yang ditunjukkan pada Gambar 5 di bawah ini:



Gambar 5 Confusion Matrix Algoritma *Logistic Regression* dengan PCA (kiri) dan tanpa PCA (kanan)

3.2. Random Forest

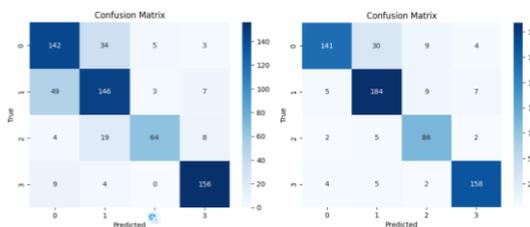
Performa algoritma *Random Forest* mengalami penurunan akurasi, seperti yang terlihat pada Tabel 3 di bawah ini:

Tabel 3 Perbandingan akurasi algoritma *Random Forest* dengan dan tanpa PCA

Method	Accuracy	Precision	Recall	F1-Score
RF	78%	78%	78%	78%
RF-NPA	87%	88%	87%	87%

Hasil yang disajikan menunjukkan bahwa penggunaan PCA dapat mengurangi akurasi algoritma *Random Forest*, yang dibuktikan dengan adanya perbedaan antara RF (dengan PCA) dan RF-NPA (tanpa PCA).

Perbedaan hasil pengujian dapat dilihat pada *confusion matrix* seperti yang ditunjukkan pada Gambar 6 di bawah ini:



Gambar 6 *Confusion Matrix* Algoritma *Random Forest* dengan PCA (kiri) dan tanpa PCA (kanan)

3.3. Support Vector Machine (SVM)

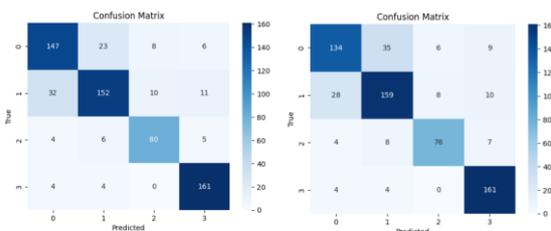
Hasil pengujian menunjukkan bahwa algoritma SVM mengalami peningkatan akurasi ketika menggunakan PCA, seperti yang tercatat pada Tabel 4 di bawah ini:

Tabel 4 Perbandingan akurasi algoritma SVM dengan dan tanpa PCA

Method	Accuracy	Precision	Recall	F1-Score
SVM	82%	83%	83%	83%
SVM-NPA	81%	81%	81%	81%

Hasil di atas menunjukkan bahwa penggunaan PCA meningkatkan akurasi dan metrik evaluasi lainnya dalam algoritma SVM.

Perbedaan hasil pengujian dapat diamati pada matriks kebingungan, seperti yang digambarkan pada Gambar 7 di bawah ini:



Gambar 7 *Confusion Matrix* dari algoritma SVM dengan PCA (kiri) dan tanpa PCA (kanan)

3.4. K-Nearest Neighbors (KNN)

Hasil pengujian menunjukkan bahwa algoritma KNN mengalami peningkatan nilai akurasi dengan

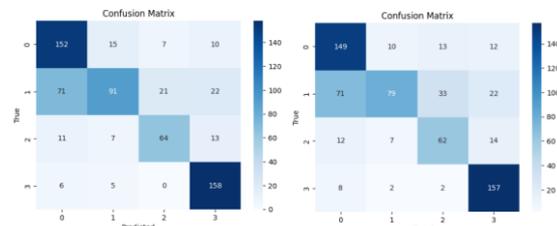
menggunakan PCA, seperti yang tertera pada Tabel 5 di bawah ini:

Tabel 5 Perbandingan akurasi algoritma KNN dengan dan tanpa PCA

Method	Accuracy	Precision	Recall	F1-Score
KNN	71%	72%	72%	72%
KNN-NPA	68%	71%	68%	67%

Hasil di atas menunjukkan bahwa penggunaan PCA meningkatkan akurasi dan metrik evaluasi lainnya dari algoritma KNN.

Perbedaan hasil pengujian dapat dilihat pada *confusion matrix* seperti yang ditunjukkan pada Gambar 8 di bawah ini:



Gambar 8 *Confusion Matrix* Algoritma KNN dengan PCA (kiri) dan tanpa PCA (kanan)

3.5. Naïve Bayes

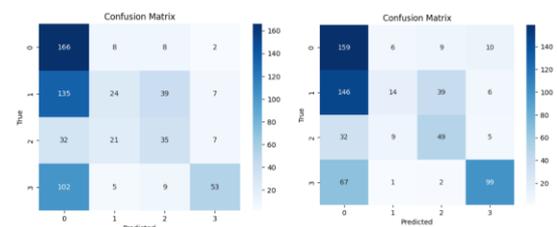
Algoritma *Naïve Bayes* mengalami penurunan performa ketika diuji dengan menggunakan PCA, seperti yang ditunjukkan pada Tabel 6 di bawah ini:

Tabel 6 Perbandingan akurasi algoritma Naïve Bayes dengan dan tanpa PCA

Method	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	42%	49%	43%	38%
Naïve Bayes-NPA	49%	54%	49%	44%

Hasil di atas menunjukkan bahwa penggunaan PCA mengakibatkan penurunan akurasi algoritma Regresi Logistik, terlihat dari perbandingan antara LR (dengan PCA) dan LR-NPA (tanpa PCA).

Perbedaan hasil pengujian dapat dilihat pada matriks kebingungan seperti yang ditunjukkan pada Gambar 9 di bawah ini:

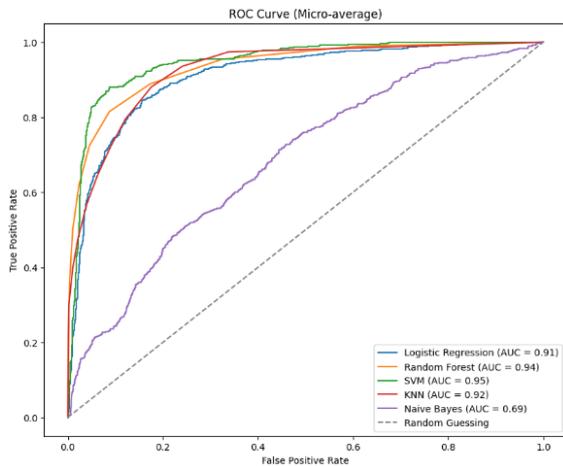


Gambar 9 *Confusion Matrix* Algoritma Naïve Bayes dengan PCA (kiri) dan tanpa PCA (kanan)

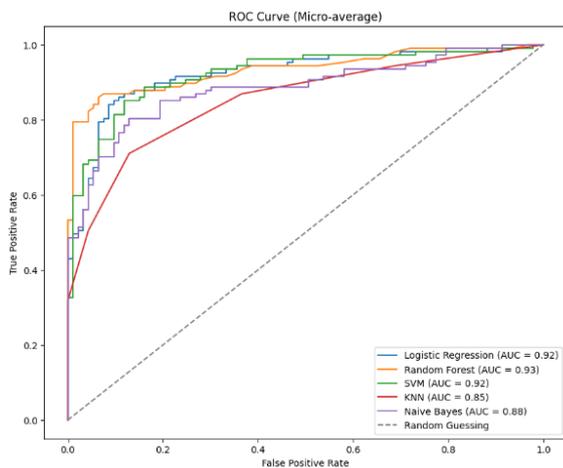
3.5. Pengukuran Kinerja dengan Area Under the Curve (AUC) & Receiver Operating Characteristic (ROC)

Setelah hasil evaluasi *Confusion Matrix* diketahui, dilanjutkan dengan evaluasi AUC-ROC

seperti yang diilustrasikan pada Gambar 10. *Multiclass AUC-ROC* dengan PCA dan Gambar 11 *Multiclass AUC-ROC* Non-PCA.



Gambar 10 Multiclass AUC-ROC dengan PCA



Gambar 11 Multiclass AUC-ROC Non-PCA

Dalam penelitian ini, kami membandingkan kinerja model klasifikasi dengan dan tanpa menggunakan Principal Component Analysis (PCA) berdasarkan Area Under the Curve (AUC) dari Receiver Operating Characteristic (ROC). Hasil menunjukkan bahwa penggunaan PCA secara signifikan meningkatkan performa Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN), dengan peningkatan AUC masing-masing sebesar 0.03 dan 0.07. Namun, model Naive Bayes menunjukkan peningkatan performa yang lebih baik tanpa PCA, dengan AUC meningkat sebesar 0.19. Logistic Regression dan Random Forest menunjukkan variasi kecil dalam performa dengan dan tanpa PCA. Secara keseluruhan, penelitian ini menegaskan bahwa dampak PCA pada kinerja model bervariasi tergantung pada karakteristik data dan algoritma yang digunakan, dan penerapan PCA harus dipertimbangkan secara kontekstual sesuai dengan kebutuhan spesifik model dan data.

4. DISKUSI

4.1. Penelitian Sebelumnya

Penelitian sebelumnya telah menunjukkan berbagai metode yang digunakan untuk klasifikasi tumor otak. [1] menggunakan CNN untuk klasifikasi gambar tumor otak dengan hasil yang menunjukkan akurasi tinggi. Studi oleh [5] menggunakan kombinasi teknik Deep Wavelet Transform (DWT) dan PCA bersama dengan jaringan syaraf tiruan (ANN), yang meningkatkan akurasi klasifikasi tumor otak dari citra MRI. Selain itu, penelitian oleh [7] menggabungkan superpixels, PCA, dan clustering K-means untuk mendeteksi tumor otak dari gambar MRI, mencapai akurasi 94%.

Studi lain yang dilakukan oleh [35] menerapkan CNN dan SVM untuk kategori otomatisasi tumor otak dari MRI, menunjukkan peningkatan signifikan dalam performa klasifikasi. [9] menggunakan teknik preprocessing, ekstraksi fitur dengan PCA, dan clustering berbasis template untuk klasifikasi tumor otak, mencapai akurasi 96.16%. [10] memanfaatkan DWT dan PCA untuk ekstraksi fitur dan menggunakan Kernel Support Vector Machine (KSVM) untuk klasifikasi, mencapai akurasi 96.25%.

Selain itu, [11] menggunakan kombinasi prapemrosesan gambar, ekstraksi fitur dengan GLCM dan CNN, serta optimasi dengan algoritma paus dan serigala abu-abu untuk mencapai akurasi deteksi tumor otak sebesar 98.9%. Studi-studi ini menunjukkan bahwa kombinasi teknik prapemrosesan yang canggih dan pemilihan algoritma yang tepat sangat penting dalam meningkatkan akurasi klasifikasi tumor otak.

4.2. Interpretasi Hasil

Dalam penelitian ini, penggunaan PCA menunjukkan hasil yang bervariasi tergantung pada algoritma yang digunakan. Logistic Regression dan Naive Bayes mengalami penurunan akurasi ketika menggunakan PCA, yang menunjukkan bahwa reduksi dimensi melalui PCA tidak efektif untuk algoritma ini. Sebaliknya, algoritma SVM dan KNN mengalami peningkatan akurasi dengan penggunaan PCA, menunjukkan bahwa PCA dapat membantu meningkatkan performa klasifikasi untuk algoritma tersebut. Random Forest menunjukkan penurunan akurasi dengan penggunaan PCA, yang mungkin disebabkan oleh sifat inherent dari algoritma yang sudah robust terhadap fitur yang berlebih.

Penelitian ini menggarisbawahi pentingnya pemilihan algoritma yang tepat dan metode prapemrosesan dalam klasifikasi citra medis. Dampak PCA pada performa model bervariasi tergantung pada karakteristik data dan algoritma yang digunakan. Oleh karena itu, penggunaan PCA harus dipertimbangkan secara kontekstual sesuai dengan kebutuhan spesifik model dan data yang digunakan. Hasil penelitian ini mendorong eksplorasi lebih lanjut terhadap metode pengurangan dimensi alternatif dan

pemilihan algoritma yang optimal untuk meningkatkan akurasi klasifikasi tumor otak.

5. SARAN

Berdasarkan hasil penelitian ini, terdapat beberapa saran yang dapat diberikan untuk penelitian lebih lanjut dan implementasi praktis. Penelitian di masa depan disarankan untuk mengeksplorasi metode pengurangan dimensi selain PCA, seperti t-SNE atau LDA, serta menggunakan dataset yang lebih besar dan beragam untuk meningkatkan generalisasi model. Selain itu, menggabungkan beberapa algoritma pembelajaran mesin dalam pendekatan ensemble dan melakukan optimasi hyperparameter yang lebih mendalam menggunakan teknik seperti Grid Search atau Random Search dapat meningkatkan kinerja model. Mengintegrasikan teknik deep learning seperti Convolutional Neural Networks (CNNs) dengan algoritma klasik, serta menggunakan metode validasi silang yang lebih ekstensif seperti k-fold cross-validation, juga disarankan untuk memastikan model bekerja dengan baik pada data yang tidak terlihat selama pelatihan. Kolaborasi dengan ahli radiologi dan profesional medis lainnya dapat memberikan wawasan yang lebih mendalam tentang fitur-fitur penting dalam klasifikasi tumor otak, sehingga model yang dikembangkan lebih relevan dalam konteks klinis. Terakhir, pengembangan aplikasi klinis berbasis web atau mobile yang mengimplementasikan model klasifikasi tumor otak ini untuk digunakan oleh dokter dan tenaga medis dalam mendiagnosis tumor otak secara real-time dapat meningkatkan deteksi dini dan pengobatan pasien. Dengan memperhatikan saran-saran ini, diharapkan penelitian di masa depan dapat terus meningkatkan akurasi dan efektivitas model pembelajaran mesin dalam klasifikasi tumor otak serta mengaplikasikannya dalam lingkungan klinis untuk manfaat yang lebih besar.

6. KESIMPULAN

Penelitian ini bertujuan untuk mengoptimalkan akurasi model pembelajaran mesin dalam klasifikasi tumor otak menggunakan citra MRI dengan memanfaatkan Analisis Komponen Utama (PCA) untuk reduksi dimensi. Beberapa algoritma pembelajaran mesin seperti Logistic Regression, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), dan Naïve Bayes telah diimplementasikan dan dievaluasi kinerjanya baik dengan maupun tanpa PCA. Hasil penelitian menunjukkan bahwa penggunaan PCA memiliki dampak yang bervariasi tergantung pada algoritma yang digunakan. Logistic Regression dan Naïve Bayes mengalami penurunan akurasi masing-masing dari 77% menjadi 69% dan dari 49% menjadi 42% dengan penggunaan PCA, menunjukkan bahwa PCA tidak efektif untuk algoritma ini. Random Forest juga mengalami penurunan akurasi dari 87% menjadi 78%

dengan PCA. Sebaliknya, PCA meningkatkan akurasi SVM dari 81% menjadi 83% dan KNN dari 68% menjadi 71%, menunjukkan efek positif dari reduksi dimensi. Evaluasi menggunakan Confusion Matrix dan kurva ROC-AUC juga menunjukkan bahwa dampak PCA bervariasi tergantung pada algoritma yang digunakan. Kesimpulannya, pemilihan algoritma dan metode preprocessing yang tepat sangat penting dalam klasifikasi tumor otak menggunakan citra MRI. PCA dapat membantu meningkatkan akurasi beberapa algoritma, namun tidak untuk semua, sehingga penggunaannya harus dipertimbangkan berdasarkan karakteristik data dan algoritma yang digunakan. Penelitian ini juga mendorong eksplorasi metode pengurangan dimensi alternatif untuk analisis citra medis.

DAFTAR PUSTAKA

- [1] M. Gupta, S. K. Sharma, and G. C. Sampada, "Classification of Brain Tumor Images Using CNN," *Comput Intell Neurosci*, vol. 2023, pp. 1–6, Oct. 2023, doi: 10.1155/2023/2002855.
- [2] American Brain Tumor Association, "Brain tumor Education," <https://www.abta.org/>.
- [3] American Society of Clinical Oncology (ASCO), "Brain Tumor: Statistics," <https://www.cancer.net/>.
- [4] R. Andre, B. Wahyu, and R. Purbaningtyas, "KLASIFIKASI TUMOR OTAK MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR EFFICIENTNET-B3," 2021. [Online]. Available: <https://jurnal.umj.ac.id/index.php/just-it/index>
- [5] B. Panda and C. S. Panda, "A Review on Brain Tumor Classification Methodologies," *Int J Sci Res Sci Technol*, pp. 346–359, Dec. 2019, doi: 10.32628/IJSRST20717.
- [6] Dr. P. and M. P. , R. and R. Ramamurthy, "Evaluation and Classification of the Brain Tumor MRI using Machine Learning Technique," *Control Engineering and Applied Informatics*, vol. 21, pp. 12–21, Dec. 2019.
- [7] M. K. Islam, M. S. Ali, M. S. Miah, M. M. Rahman, M. S. Alam, and M. A. Hossain, "Brain tumor detection in MR image using superpixels, principal component analysis and template based K-means clustering algorithm," *Machine Learning with Applications*, vol. 5, p. 100044, Sep. 2021, doi: 10.1016/j.mlwa.2021.100044.
- [8] G. Cinarer and B. G. Emiroglu, "Classification of Brain Tumors by Machine Learning Algorithms," in *2019 3rd International Symposium on Multidisciplinary Studies and*

- Innovative Technologies (ISMSIT)*, IEEE, Oct. 2019, pp. 1–4. doi: 10.1109/ISMSIT.2019.8932878.
- [9] M. Pareek, C. K. Jha, and S. Mukherjee, “Brain Tumor Classification from MRI Images and Calculation of Tumor Area,” 2020, pp. 73–83. doi: 10.1007/978-981-15-0751-9_7.
- [10] D. M. Toufiq, A. M. Sagheer, and H. Veisi, “Brain tumor identification with a hybrid feature extraction method based on discrete wavelet transform and principle component analysis,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2588–2597, Oct. 2021, doi: 10.11591/eei.v10i5.3013.
- [11] L. Hussain, S. Saeed, I. A. Awan, A. Idris, M. S. A. Nadeem, and Q.-A. Chaudhry, “Detecting Brain Tumor using Machines Learning Techniques Based on Different Features Extracting Strategies,” *Current Medical Imaging Formerly Current Medical Imaging Reviews*, vol. 15, no. 6, pp. 595–606, Jul. 2019, doi: 10.2174/1573405614666180718123533.
- [12] S. Mohsen, A. M. Ali, E.-S. M. El-Rabaie, A. ElKaseer, S. G. Scholz, and A. M. A. Hassan, “Brain Tumor Classification Using Hybrid Single Image Super-Resolution Technique With ResNext101_32× 8d and VGG19 Pre-Trained Models,” *IEEE Access*, vol. 11, pp. 55582–55595, 2023, doi: 10.1109/ACCESS.2023.3281529.
- [13] S. K. Panda and R. C. Barik, “MR Brain 2D image Tumor and Cyst Classification Approach: an Empirical Analogy,” in *2023 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*, IEEE, Feb. 2023, pp. 1–6. doi: 10.1109/SCEECS57921.2023.10063064.
- [14] F. S. Chowdhury, T. Noor, Md. S. Islam, and M. K. Alam, “Brain Tumor Classification Using Watershed Segmentation with ANN Classifier,” in *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, IEEE, Feb. 2023, pp. 1–5. doi: 10.1109/ECCE57851.2023.10101528.
- [15] G. L. E. Maquen-Niño *et al.*, “Brain Tumor Classification Deep Learning Model Using Neural Networks,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 09, pp. 81–92, Jul. 2023, doi: 10.3991/ijoe.v19i09.38819.
- [16] K. V. Kumar, M. Baid, and K. Menon, “Brain Tumor Classification using Transfer Learning on Augmented Data and Visual Explanation using Grad-CAM,” in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, May 2023, pp. 965–971. doi: 10.1109/ICICCS56967.2023.10142464.
- [17] Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, and Swati Kanchan, “Brain Tumor Classification (MRI),” <https://www.kaggle.com/dsv/1183165>.
- [18] N. Basuni and Amril Mutoi Siregar, “Comparison of the Accuracy of Drug User Classification Models Using Machine Learning Methods,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1348–1353, Dec. 2023, doi: 10.29207/resti.v7i6.5401.
- [19] A. R. Alexey Spizhevoy, *OpenCV 3 Computer Vision with Python Cookbook: Leverage the power of OpenCV 3 and Python to build computer vision applications*, 1st ed. Birmingham : Packt Publishing, 2019.
- [20] S. A. B. R. Kalakuntla, “Detection of Brain Tumor in Magnetic Resonance Imaging (MRI) Images using Fuzzy C-Means and Thresholding,” *SUNY Polytechnic Institute*, Aug. 2020.
- [21] Krishnendu Kar, *Mastering Computer Vision with TensorFlow 2.x: Build advanced computer vision applications using machine learning and deep learning techniques.*, 1st ed. Birmingham-Mumbai : Packt Publishing, 20AD. Accessed: May 26, 2024. [Online]. Available: <https://www.oreilly.com/library/view/mastering-computer-vision/9781838827069/>
- [22] Abid Ali Awan, “Recurrent Neural Network Tutorial (RNN),” <https://www.datacamp.com/>. Accessed: May 27, 2024. [Online]. Available: <https://www.datacamp.com/tutorial/tutorial-for-recurrent-neural-network>
- [23] Jason Brownlee, “Train-Test Split for Evaluating Machine Learning Algorithms,” <https://machinelearningmastery.com/>. Accessed: May 27, 2024. [Online]. Available: <https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/>
- [24] Geeks for Geeks Team, “How To Do Train Test Split Using Sklearn In Python,” <https://www.geeksforgeeks.org/>. Accessed: May 27, 2024. [Online]. Available: <https://www.geeksforgeeks.org/how-to-do-train-test-split-using-sklearn-in-python/>
- [25] Ryan Wells, “Handwriting Image Classification with Python Sklearn,” <https://wellsr.com/>. Accessed: May 27, 2024.

- [Online]. Available: <https://wellsr.com/python/image-classification-with-python-sklearn-handwriting/>
- [26] A. Ambarwari, Q. Jafar Adrian, and Y. Herdiyeni, "Analysis of the Effect of Data Scaling on the Performance of the Machine Learning Algorithm for Plant Identification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 1, pp. 117–122, Feb. 2020, doi: 10.29207/resti.v4i1.1517.
- [27] D. U. Ozsahin, M. Taiwo Mustapha, A. S. Mubarak, Z. Said Ameen, and B. Uzun, "Impact of feature scaling on machine learning models for the diagnosis of diabetes," in *2022 International Conference on Artificial Intelligence in Everything (AIE)*, IEEE, Aug. 2022, pp. 87–94. doi: 10.1109/AIE57029.2022.00024.
- [28] J. Lever, M. Krzywinski, and N. Altman, "Principal component analysis," *Nat Methods*, vol. 14, no. 7, pp. 641–642, Jul. 2019, doi: 10.1038/nmeth.4346.
- [29] R. D. Cook, "Principal Components, Sufficient Dimension Reduction, and Envelopes," *Annu Rev Stat Appl*, vol. 5, no. 1, pp. 533–559, Mar. 2019, doi: 10.1146/annurev-statistics-031017-100257.
- [30] K. J. Galinsky *et al.*, "Fast Principal-Component Analysis Reveals Convergent Evolution of ADH1B in Europe and East Asia," *The American Journal of Human Genetics*, vol. 98, no. 3, pp. 456–472, Mar. 2018, doi: 10.1016/j.ajhg.2015.12.022.
- [31] C. C. Chang, C. C. Chow, L. C. Tellier, S. Vattikuti, S. M. Purcell, and J. J. Lee, "Second-generation PLINK: rising to the challenge of larger and richer datasets," *Gigascience*, vol. 4, no. 1, p. 7, Dec. 2018, doi: 10.1186/s13742-015-0047-8.
- [32] I. Mathieson *et al.*, "The genomic history of southeastern Europe," *Nature*, vol. 555, no. 7695, pp. 197–203, Mar. 2019, doi: 10.1038/nature25778.
- [33] A. I. Khuri, "Introduction to Linear Regression Analysis, Fifth Edition by Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining," *International Statistical Review*, vol. 81, no. 2, pp. 318–319, Aug. 2013, doi: 10.1111/insr.12020_10.
- [34] M. Mia, A. F. N. Masruriyah, and A. R. Pratama, "The Utilization of Decision Tree Algorithm In Order to Predict Heart Disease," *JURNAL SISFOTEK GLOBAL*, vol. 12, no. 2, p. 138, Sep. 2022, doi: 10.38101/sisfotek.v12i2.551.
- [35] S. Deepak and P. M. Ameer, "Automated Categorization of Brain Tumor from MRI Using CNN features and SVM," *J Ambient Intell Humaniz Comput*, vol. 12, no. 8, pp. 8357–8369, Aug. 2021, doi: 10.1007/s12652-020-02568-w.
- [36] A. M. S. S. F. Koirunnisa Koirunnisa, "Optimized Machine Learning Performance with Feature Selection for Breast Cancer Disease Classification," *JITEKI*, vol. 9, pp. 1131–1143, Nov. 2023.
- [37] R. H. Ramdlon, E. Martiana Kusumaningtyas, and T. Karlita, "Brain Tumor Classification Using MRI Images with K-Nearest Neighbor Method," in *2019 International Electronics Symposium (IES)*, IEEE, Sep. 2019, pp. 660–667. doi: 10.1109/ELECSYM.2019.8901560.
- [38] L. Setiyani, H. H. Handayani, and W. A. Geraldine, "Comparison of the Performance of the SQL Injection Detection Model Using CNN, Logistic Regression, Random Forest, Naive Bayes, and Decision Tree," in *2023 1st International Conference on Advanced Engineering and Technologies (ICONNIC)*, IEEE, Oct. 2023, pp. 109–113. doi: 10.1109/ICONNIC59854.2023.10467843.
- [39] H. Yun, "Prediction model of algal blooms using logistic regression and confusion matrix," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, p. 2407, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2407-2413.
- [40] H. Sholikhin, D. Wahiddin, and K. A. Baihaqi, "PENERAPAN ALGORITMA BACKWARD CHAINING UNTUK MENDIAGNOSA PENYAKIT DAN HAMA TANAMAN PADI," vol. III, no. 1, p. 22, 2022.
- [41] Ralph Stern, "Interpretation of the Area Under the ROC Curve for Risk Prediction Models," *Cornel University*, Feb. 2021.
- [42] ع. ركاب, "الصوت الروائي النسائي الجزائري ركاب, وهندسة الأمكنة: قراءة في نماذج." □□□□ □□□□, p. 213, 2019, doi: 10.37167/1677-000-012-020.
- [43] M. Dalili Shoaie and M. Dastani, "The Role of Social Media During the COVID-19 Crisis: A Narrative Review," *Health Technology Assessment in Action*, Mar. 2021, doi: 10.18502/htaa.v4i1.5865.