

OPTIMALISASI MODEL KLASIFIKASI MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE PADA CITRA BRAIN TUMOR

Bobi Darma Irawan, Rudi Kurniawan, Yudhistira Arie Wijaya

Teknik Informatika, STMIK IKMI Cirebon

Jalan Perjuangan No.10 B Majasem Kec. Kesambi Kota Cirebon, Indonesia

bobidarmairawan.9b.1216@gmail.com

ABSTRAK

Analisis citra medis, khususnya pada citra tumor otak, memiliki peran penting dalam membantu proses diagnosis penyakit. Permasalahan utama dalam klasifikasi citra tumor otak adalah bagaimana meningkatkan akurasi dan efisiensi sistem klasifikasi. Penelitian ini bertujuan untuk mengoptimalkan model klasifikasi dengan menggunakan algoritma *Support Vector Machine* (SVM), yang dikenal andal dalam menangani data non-linear dan berdimensi tinggi. Metode yang digunakan meliputi tahap preprocessing citra, ekstraksi fitur menggunakan metode tertentu, serta pengoptimalan parameter SVM, seperti kernel, nilai regulasi, dan parameter gamma. Dataset berupa citra MRI otak dibagi menjadi data pelatihan dan pengujian untuk mengevaluasi kinerja model. Penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan sistem berbasis komputer untuk diagnosis tumor otak secara cepat, akurat, dan efisien. Hasil dari model SVM pada proyek ini menunjukkan bahwa klasifikasi menghasilkan class positif sebesar 0,950 dan class negatif sebesar 0,050. Model ini mencapai akurasi 81%, dengan precision 100% dan recall 80%, menunjukkan bahwa SVM memiliki performa yang baik dalam analisis klasifikasi citra tumor otak.

Kata kunci : *Support Vector Machine*, Klasifikasi Tumor, Citra MRI Otak.

1. PENDAHULUAN

Dalam beberapa tahun terakhir, analisis citra medis telah menjadi salah satu bidang penelitian yang penting, terutama dalam mendukung proses diagnosis penyakit. Salah satu dampak yang paling menonjol dari informatika pada pencitraan medis adalah peningkatan deteksi dan klasifikasi tumor. Platform yang digerakkan oleh AI telah menunjukkan akurasi yang luar biasa dalam mengkategorikan gambar medis, memungkinkan identifikasi anomali yang efisien dalam pemindaian MRI, terutama untuk tumor otak. Integrasi teknik pembelajaran mendalam, seperti convolutional neural network (CNN), telah menyempurnakan proses klasifikasi lebih lanjut, yang memungkinkan diferensiasi antara berbagai jenis dan stadium tumor.

Relevansi untuk mengatasi tantangan ini digarisbawahi oleh meningkatnya ketergantungan pada sistem otomatis untuk deteksi dan klasifikasi tumor dalam pengaturan klinis. Karena volume data pencitraan medis terus bertambah, permintaan akan metode klasifikasi yang efisien, akurat, dan dapat diandalkan menjadi sangat penting. Teknik pembelajaran mesin saat ini, khususnya model pembelajaran mendalam seperti jaringan saraf tiruan (CNN), telah menunjukkan harapan dalam meningkatkan akurasi klasifikasi. Namun, mereka sering kali kesulitan dengan terbatasnya ketersediaan dataset MRI beranotasi, yang dapat menyebabkan overfitting dan generalisasi yang buruk pada data yang tidak terlihat.

Terlepas dari kemajuan ini, keterbatasan tetap ada dalam literatur yang ada. Sebagai contoh, meskipun banyak penelitian yang berfokus pada akurasi deteksi tumor, mereka sering mengabaikan

implikasi heterogenitas tumor dan pengaruh lingkungan peritumoral pada hasil pengobatan. Hu et al. menekankan bahwa menggabungkan fitur radiomik peritumoral dapat secara signifikan meningkatkan model prediktif untuk respons pengobatan, namun aspek ini masih belum dieksplorasi dalam banyak penelitian.

Tujuan utama dari penelitian terhadap data citra tumor hasil MRI adalah untuk meningkatkan akurasi dan efisiensi deteksi dan segmentasi tumor otak melalui teknik komputasi yang canggih, khususnya deep learning dan kecerdasan buatan. Penelitian ini bertujuan untuk mengatasi kesenjangan yang signifikan dalam metodologi yang ada untuk analisis tumor otak, yang sering kali mengalami keterbatasan seperti variabilitas kualitas gambar, interpretasi subjektif oleh ahli radiologi, dan kompleksitas morfologi tumor.

Dalam penelitian ini, kami akan menggunakan pendekatan multifaset untuk menganalisis data citra tumor yang berasal dari MRI, mengintegrasikan teknik komputasi canggih dan metodologi radiomik. Metode utama kami akan melibatkan analisis radiomik, yang mengekstrak fitur kuantitatif dari gambar MRI untuk memprediksi perilaku dan karakteristik tumor. Penelitian sebelumnya telah menunjukkan nilai prognostik pencitraan tertimbang difusi (DWI) dan pencitraan tertimbang perfusi (PWI) pada glioblastoma, yang menyoroti pentingnya parameter seperti Koefisien Difusi Nyata (ADC) dan Volume Darah Otak (CBV) dalam memahami selularitas dan vaskularisasi tumor.

Hasil pencapaian tujuan penelitian mengenai klasifikasi data citra tumor MRI memiliki implikasi yang signifikan untuk bidang informatika medis dan

lainnya. Pertama, pengembangan algoritme klasifikasi yang kuat dapat meningkatkan akurasi diagnostik, sehingga memungkinkan dokter membuat keputusan yang lebih tepat mengenai rencana perawatan pasien. Peningkatan ketepatan diagnostik ini dapat mengarah pada deteksi tumor yang lebih dini, yang sangat penting dalam meningkatkan hasil dan tingkat kelangsungan hidup pasien. Selain itu, temuan dari penelitian ini dapat menjadi dasar untuk integrasi teknologi kecerdasan buatan (AI) dan pembelajaran mesin (ML) dalam pengaturan klinis.

2. TINJAUAN PUSTAKA

2.1. Second-Level Heading

Penelitian ini mengevaluasi penggunaan metode Support Vector Machine (SVM) dalam menilai teknologi metaverse di platform Roblox. Data tentang aspek teknis dan pengalaman pengguna dari Roblox, seperti kinerja platform, kepuasan pengguna, dan keamanan, dikumpulkan untuk analisis. Melalui penerapan SVM, faktor-faktor yang mempengaruhi pengalaman pengguna di Roblox diidentifikasi. Hasil penelitian menunjukkan keefektifan SVM dalam mengklasifikasikan kinerja teknologi metaverse di Roblox, memberikan pandangan yang berguna bagi pengembang dalam meningkatkan pengalaman pengguna.

Membahasa Mengenai Aplikasi Penyaringan Spam e-Mail Menggunakan Teknik Machine Learning dengan Metode Support Vector Machines. Penelitian yang dilakukan menggunakan dataset yang terdiri dari email-email spam dan non-spam untuk melatih model klasifikasi yang dapat membedakan antara keduanya. Metode yang kami terapkan meliputi pemrosesan teks, ekstraksi fitur, dan penggunaan algoritma pembelajaran mesin seperti Naive Bayes, Support Vector Machine, dan Random Forest. Setelah menjalankan serangkaian eksperimen, kami berhasil menunjukkan bahwa model klasifikasi yang kami bangun memiliki tingkat akurasi yang tinggi dalam mengidentifikasi email-email spam. Email spam filtering is an evolving challenge in maintaining electronic communication security.

Membahas mengenai aplikasi mobile seperti Home Credit. Aplikasi ini telah diunduh oleh lebih dari 10 juta pengguna Android dengan peringkat keseluruhan 4,4 di Google Play Store. Dalam penelitian ini, dilakukan analisis sentimen menggunakan metode Support Vector Machine (SVM) dan K-Nearest Neighbor (KNN) pada data ulasan dari Google Play Store. Data yang diambil berjumlah 2.845 dengan informasi tentang skor dan komentar. Sentimen positif dan negatif ditentukan berdasarkan skor, dengan skor 4 dan 5 untuk sentimen positif, serta skor 1, 2, dan 3 untuk sentimen negatif. Hasilnya menunjukkan bahwa metode SVM memiliki presisi 89%, recall 86%, F1-score 87%, dan akurasi 88%. Sementara metode KNN memiliki presisi 79%, recall 80%, F1-score 79%, dan akurasi 79%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa

metode Support Vector Machine lebih baik dalam melakukan analisis sentimen dalam penelitian ini.

Bermunculan adanya kasus penyalahgunaan formalin sebagai bahan pengawet untuk tahu dan ketersediaan cara deteksi kandungan formalin yang tergolong tidak praktis dan membutuhkan tenaga ahli khusus, Penelitian ini bertujuan untuk melakukan analisis performa suatu metode. Pengolahan data tersebut digunakan untuk memprediksi terdapatnya kandungan formalin pada tahu melalui aroma yang dikeluarkan. Teknik klasifikasi yang digunakan yaitu dengan menggunakan pengukuran performa (Accuracy, Precision, Recall, dan F-Measure) menggunakan Metode Support Vector Machine. Inputan yang digunakan yaitu atribut data aroma tahu berformalin, dan tidak berformalin. Hasil penelitian ini menunjukkan metode SVM mempunyai kinerja yang cukup baik. Berdasarkan hasil pengujian menggunakan confusion matrix didapatkan hasil uji performa Metode SVM accuracy 67%, precision 69%, recall 67%, dan f-measure 66% dengan Nilai K=5.

Penggunaan algoritma Support Vector Machine dalam menganalisis bidang medis. Pada bidang medis, Support Vector Machine diterapkan pada klasifikasi diagnosis suatu penyakit. Penelitian ini berfokus pada penyakit kanker payudara. Hasil penelitian menunjukkan bahwa model SVM terbaik ketika nilai $C=10$. Variabel radius (X_1) dan perimeter (X_3) memiliki pengaruh yang besar pada diagnosis kanker payudara kategori benign, sedangkan variabel texture dan smoothness memiliki pengaruh besar terhadap diagnosis kanker payudara kategori malignant. Tingkat akurasi metode Support Vector Machine pada klasifikasi diagnosis kanker payudara sebesar 96,49%.

Perkembangan teknologi machine learning telah memungkinkan proses analisis data yang lebih kompleks dan akurat dalam bidang kesehatan dan diagnosa penyakit seperti diabetes mellitus. Penelitian ini bertujuan untuk mengimplementasikan algoritma nu-Support Vector Machine (nu-SVM) pada pembuatan model klasifikasi sekuens DNA manusia sehingga dapat mengidentifikasi penderita diabetes mellitus. Hasilnya, Implementasi nu-SVM pada model klasifikasi sekuens DNA menghasilkan akurasi rata-rata sebesar 90%, presisi rata-rata sebesar 90,18%, serta recall dan F1-score rata-rata sebesar 90%. Hasil ini menunjukkan jika algoritma nu-SVM terbukti mampu mengidentifikasi sekuens DNA manusia dengan cepat dan akurat sehingga mampu memberikan kontribusi terhadap pengembangan sistem diagnosis dini penyakit yang lebih efektif dan efisien.

Membahas Mengenai Tumor otak adalah kondisi yang ditandai dengan tumbuhnya sel-sel abnormal dan ganas di dalam otak. Citra digital melalui Magnetic Resonance Imaging (MRI) merupakan salah satu metode untuk membantu dokter dalam mengklasifikasikan jenis tumor otak. Para peneliti terdahulu telah mendapatkan hasil yang memuaskan di semua sub bidang analisis citra medis, seperti klasifikasi, deteksi, segmentasi, dan peningkatan citra.

Dengan keberhasilan ini, para peneliti menemukan sebuah model yang dapat mendiagnosis dan mendeteksi secara otomatis, yang dapat membantu para ahli di bidangnya dan menunjukkan bahwa teknologi deep learning akan sangat bermanfaat bagi kemajuan analisis citra medis, hasil penelitian diperoleh akurasi sebesar 0,9642.

Membahas penggunaan teknik grid search untuk mengoptimalkan parameter C dan gamma pada algoritma Support Vector Machine (SVM) dengan kernel Radial Basis Function (RBF). Penelitian ini menggunakan data medis untuk diagnosis penyakit jantung, yang mencakup berbagai fitur seperti usia, tekanan darah, dan kadar kolesterol. Optimalisasi dengan grid search berhasil meningkatkan akurasi model dari 85% menjadi 91%, menunjukkan bahwa tuning parameter secara sistematis dapat meningkatkan performa klasifikasi secara signifikan.

Membahas pengintegrasian teknik PCA (Principal Component Analysis) dengan algoritma Support Vector Machine (SVM) untuk mengatasi tantangan dalam menangani dataset berdimensi tinggi. Penelitian ini menggunakan dataset pengenalan pola wajah dari database LFW (Labeled Faces in the Wild), yang mengandung fitur-fitur kompleks dan berdimensi tinggi. Hasil penelitian menunjukkan bahwa kombinasi PCA dengan kernel RBF pada SVM berhasil meningkatkan efisiensi komputasi hingga 40%, sambil mempertahankan akurasi model stabil di angka 95%. Selain itu, PCA terbukti sangat efektif pada dataset dengan fitur-fitur yang memiliki multikorelasi tinggi, membantu mengurangi redundansi data tanpa kehilangan informasi penting.

memperkenalkan Bayesian Optimization sebagai metode untuk optimalisasi parameter Support Vector Machine (SVM) yang lebih efisien dibandingkan grid search. Penelitian ini menggunakan dataset analisis sentimen dari data ulasan produk, yang terdiri dari teks dengan fitur kompleks. Hasil menunjukkan bahwa Bayesian Optimization mampu menemukan kombinasi parameter optimal dalam 50 iterasi, jauh lebih cepat dibandingkan 200 iterasi pada grid search, sekaligus meningkatkan akurasi model hingga 93%. Metode ini dinilai sangat cocok untuk dataset besar, terutama ketika sumber daya komputasi terbatas, karena mampu mengurangi kebutuhan waktu tanpa mengorbankan performa.

Membahas penggunaan weighted SVM untuk mengatasi masalah distribusi kelas yang tidak seimbang dalam dataset. Penelitian ini menggunakan data pendeteksian penipuan keuangan dengan rasio kelas 1:100, di mana kelas minoritas (penipuan) jauh lebih sedikit dibandingkan kelas mayoritas. Dengan menambahkan bobot pada kelas minoritas, hasil penelitian menunjukkan bahwa akurasi prediksi pada kelas minoritas meningkat signifikan, dari 68% menjadi 85%. Namun, penelitian ini juga mencatat bahwa model memerlukan tuning parameter yang lebih hati-hati untuk mencegah overfitting, terutama

ketika menangani dataset dengan ketidakseimbangan yang ekstrem.

Membandingkan performa kernel RBF, polynomial, dan sigmoid pada data non-linear untuk klasifikasi. Penelitian ini menggunakan dataset genetik untuk memprediksi mutasi gen, yang memiliki hubungan non-linear antar fitur. Hasil penelitian menunjukkan bahwa kernel RBF memberikan akurasi terbaik sebesar 94%, diikuti oleh kernel polynomial dengan akurasi 91%, dan kernel sigmoid dengan akurasi 87%. Penelitian ini menekankan bahwa pemilihan kernel yang tepat sangat bergantung pada karakteristik data yang digunakan, dan memilih kernel yang sesuai dapat meningkatkan performa model secara signifikan.

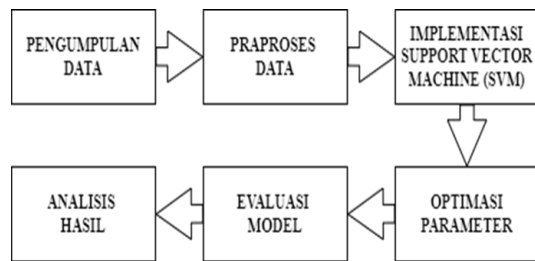
Mengeksplorasi penerapan ensemble SVM untuk meningkatkan akurasi klasifikasi pada data yang memiliki noise tinggi. Dalam penelitian ini, Li et al. mengembangkan metode ensemble yang menggabungkan beberapa model SVM untuk memperbaiki ketahanan model terhadap overfitting dan meningkatkan stabilitas hasil klasifikasi. Hasil penelitian menunjukkan bahwa kombinasi beberapa model SVM dalam ensemble menghasilkan akurasi yang lebih stabil dibandingkan dengan penggunaan SVM tunggal, serta lebih tahan terhadap gangguan yang disebabkan oleh noise tinggi pada dataset. Pendekatan ini terbukti efektif dalam meningkatkan performa model pada dataset yang tidak seimbang atau berisik, terutama pada aplikasi dalam pengenalan pola dan deteksi anomali.

Menggunakan SVM dengan kernel RBF untuk klasifikasi tumor pada dataset kanker paru-paru berdasarkan data histopathologi. Penelitian ini bertujuan untuk mengoptimalkan parameter model SVM, dengan memanfaatkan Bayesian optimization untuk memilih parameter yang tepat dan mengurangi risiko overfitting, yang sering terjadi pada dataset medis yang memiliki dimensi tinggi. Hasil eksperimen menunjukkan bahwa optimasi parameter menggunakan Bayesian optimization meningkatkan akurasi klasifikasi hingga mencapai 94%, sebuah peningkatan yang signifikan dibandingkan dengan model SVM standar.

Mengembangkan model SVM dengan kernel sigmoid untuk deteksi tumor pankreas pada dataset medis. Penelitian ini menunjukkan bahwa pemilihan kernel yang tepat dapat mempengaruhi performa klasifikasi secara signifikan. Dalam eksperimen ini, kernel sigmoid memberikan akurasi terbaik untuk dataset tumor pankreas, mencapai 91%, yang lebih tinggi dibandingkan dengan penggunaan kernel lainnya seperti RBF. Selain itu, model yang menggunakan kernel sigmoid juga menunjukkan waktu komputasi yang lebih efisien dibandingkan dengan kernel RBF, yang seringkali membutuhkan lebih banyak waktu untuk memproses data medis dengan dimensi tinggi.

3. METODE PENELITIAN

Pada bagian ini akan dipaparkan tahapan-tahapan penelitian mulai dari awal sampai selesai. Tahapan penelitian yang dilakukan dapat dilihat pada Gambar.



Gambar 1. Tahapan Penelitian

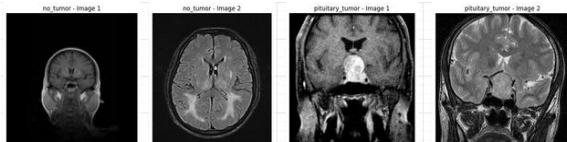
3.1. Studi Literatur

Studi Literatur adalah tahapan observasi dan pemahaman konsep, teori dan materi pendukung penelitian. Literatur yang di kaji meliputi, Machine Learning.

3.2. Pengumpulan Data

Pada tahap pengumpulan data ini dilakukan pencarian dataset yang relevan terhadap penelitian ini. Dataset yang dikumpulkan merupakan data citra MRI tumor otak yang terdapat label berupa adanya tumor ataupun tidak ada tumor.

Pada tahapan ini menggunakan dataset yang berisikan citra MRI tumor otak yang dapat diakses melalui situs kaggle dengan dataset bernama Training dan Testing. Pada dataset ini terdapat citra dengan jumlah total 1401 citra MRI otak, di mana 500 citra merupakan citra MRI dengan no tumor dan 901 citra merupakan pituitary tumor. Beberapa contoh citra dari dataset tumor otak testing dan training dapat dilihat pada Gambar 2.



Gambar 2. Contoh Testing Dan Training

Gambar 2 berisi contoh dataset yang berisi citra MRI yang termasuk no tumor dan pituitary tumor . Karakteristik dari data yang termasuk tumor terdapat area lesi sebagai ilustrasi dari tumor, seperti terlihat pada Gambar 2 bagian kanan. Sedangkan untuk kondisi normal yang bukan tumor terlihat pada Gambar 2 bagian kiri.

3.3. Pengembangan perangkat lunak

Pada tahap ini pengembangan perangkat lunak di lakukan dengan metode pengembangan sequential linear atau yang dikenal juga dengan nama *waterfall*. Tahapannya meliputi analysis, design, code dan test.

Secara garis besar tahap ini terdii dari perancangan perangkat lunak dalam lingkungan Python dan implementasi algoritma KNN, AdaKNN, IT2FKNN, FRNN, FKNN, dan langkah optimasi performa algoritma dengan reduksi data menggunakan PCA serta optimasi pada lingkungan Python menggunakan library Cython.

3.4. Eksperimen

Pada tahap ini merupakan perancangan skenario dan hasil eksperimen yang akan dilakukan untuk menentukan parameter yang paling bagus saat ekstraksi data, banyaknya komponen saat reduksi data menggunakan PCA dan membandingkan algoritma-algoritma (KNN, AdaKNN, FKNN, IT2FKNN dan FRNN) yang terbaik saat melakukan klasifikasi serta 61 Ramadhan Adityo Kuncoro, 2017 IMPLEMENTASI DAN PERBANDINGAN METODE MACHINE LEARNING UNTUK KLASIFIKASI PENYAKIT PADA DAUN TOMAT Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu memilih nilai k yang paling optimal. Secara garis besar terdapat tiga skenario utama dan terdapat beberapa sub skenario dalam tahap ini.

3.5. Analisis hasil eksperimen

Setelah hasil dari simulasi diketahui, maka akan dilakukan analisis perbandingan dari hasil semua metode yang diterapkan untuk mengambil keputusan metode mana yang lebih baik dan akurat untuk pemecahan masalah dari kasus dan melakukan perbandingan dengan penelitian sebelumnya.

3.6. Dokumentasi

Tahap ini adalah tahap untuk melakukan penulisan laporan dari penelitian dan eksperimen yang berhasil di lakukan.

4. HASIL DAN PEMBAHASAN

4.1. Data Set

Dataset Brain Tumor MRI yang digunakan pada penelitian ini pada Training dan Testing berupa citra dengan class no_tumor dan pituitary_tumor. Adapun jumlah citra keseluruhan pada penelitian tersedia pada Tabel 1.

Tabel 1. Jumlah citra pada masing-masing class

No	Class	Training	Testing	Dimensi	Resolusi	Depth
1.	No tumor	395	105	236x236	96x96	24
2.	Pituitary tumor	827	74	236x236	96x96	24

4.2. Image Proprocessing

Berikut ini adalah skrip Python yang berfungsi untuk mengimpor beberapa library penting guna mendukung program.

```
import os
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score,
recall_score, confusion_matrix
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
```

Gambar 3. Skrip Python

Berikut adalah penjelasan dan fungsi dari setiap perintah dalam skrip Python yang telah dibuat.

- a. os: Berinteraksi dengan sistem file (membaca file, membuat folder, mengatur path).
- b. cv2 (OpenCV): Pemrosesan gambar dan video (membaca, mengedit, dan analisis gambar).
- c. numpy: Komputasi numerik untuk operasi array dan matriks.
- d. matplotlib.pyplot & seaborn: Visualisasi data; seaborn lebih fokus pada statistik.
- e. train_test_split: Membagi dataset menjadi training dan testing.
- f. SVC (Support Vector Classifier): Algoritma SVM untuk klasifikasi.
- g. KNeighborsClassifier (KNN): Klasifikasi berdasarkan tetangga terdekat.
- h. accuracy_score: Mengukur akurasi model.
- i. precision_score: Mengukur presisi (prediksi positif yang benar).
- j. recall_score: Mengukur sensitivitas model terhadap data positif.
- k. confusion_matrix: Menampilkan hasil prediksi benar & salah.
- l. StandardScaler: Normalisasi data agar memiliki rata-rata 0 dan standar deviasi 1.
- m. plt (matplotlib.pyplot): Membuat grafik visual.
- n. sns (seaborn): Visualisasi statistik yang lebih menarik, termasuk heatmap.

Berikut Adalah Penjelasan Dan Fungsi Dari Setiap Perintah Dari *Path To The Dataset*

```
[ ] train_path = '/content/Dataset/train'
    test_path = '/content/Dataset/test'
```

Gambar 4. Path To The Dataset

Berikut ini adalah penjelasan fungsi dari path yang digunakan untuk mengelola lokasi file.

```
Original Image Properties (Image 2): Shape=(512, 512, 3), dtype=uint8  
Grayscale Image Properties (Image 2): Shape=(512, 512), dtype=uint8  
Resized Image Properties (Image 2): Shape=(64, 64), dtype=uint8  
1D array for Image 2:  
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] ... (truncated)
```

- train_path:** Path menuju dataset pelatihan (/content/Dataset/train), digunakan untuk melatih model agar mengenali pola data.
- test_path:** Path menuju dataset pengujian (/content/Dataset/test), digunakan untuk mengevaluasi performa model tanpa mempengaruhi pelatihan.

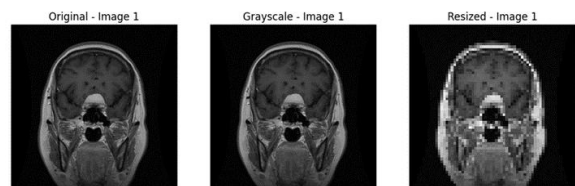
Berikut Adalah Penjelasan Dan Fungsi Dari Setiap Perintah Pada Skrip Function to load and preprocess images.

```
def load_images(path):
    images = []
    labels = []
    for label in os.listdir(path):
        label_path = os.path.join(path, label)
        for image_name in os.listdir(label_path):
            image_path = os.path.join(label_path, image_name)
            img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
            if img is not None:
                img = cv2.resize(img, (64, 64))
                images.append(img.flatten())
                labels.append(label)
            else:
                print(f"Error loading image: {image_path}")
    return np.array(images), np.array(labels)
```

Gambar 5. Function To Load And Preprecing Image

Berikut adalah penjelasan fungsi untuk memuat dan memproses gambar agar siap digunakan dalam program.

- a. `load_images(path):` Membaca, memproses (ubah ukuran & format), dan mengembalikan gambar serta label dalam bentuk array.
- b. `images:` Menyimpan gambar yang telah diproses.
- c. `labels:` Menyimpan label kelas gambar.
- d. `for label in os.listdir(path):` Menelusuri subfolder dalam direktori sebagai label kelas (misalnya, `no_tumor`, `pituitary_tumor`).
- e. `label_path:` Path lengkap ke subfolder kelas (misalnya, `/Dataset/train/no_tumor`).
- f. `for image_name in os.listdir(label_path):` Iterasi setiap file gambar dalam folder kelas.
- g. `image_path = os.path.join(label_path, image_name):` Menggabungkan path folder dengan nama file gambar.
- h. `cv2.imread(image_path, cv2.IMREAD_GRAYSCALE):` Membaca gambar dalam mode grayscale untuk pemrosesan lebih sederhana.
- i. `if img is not None:` Memastikan gambar valid sebelum diproses.
- j. `cv2.resize(img, (64, 64)):` Mengubah ukuran gambar menjadi 64×64 piksel agar seragam.
- k. `img.flatten():` Mengubah gambar 2D (64×64) menjadi array 1D (panjang 4096).
- l. `labels.append(label):` Menyimpan label gambar ke dalam list.
- m. `else: print(f'Error loading image: {image_path}')` Menampilkan pesan error jika gambar gagal dibaca.



Gambar 6. Contoh Gambar 2D 64x64

Gambar 6 berisi contoh dataset yang berisi citra MRI yang termasuk no_tumor dan pituitary_tumor. Mengubah ukuran gambar menjadi 64x64. pada Gambar 6 bagian kiri adalah original, bagian tengah adalah Grayscale, dan bagian kanan adalah Resized yang telah di ubah ke ukuran 64x64.

4.3. Klarifikasi Menggunakan Algoritma SVM.

Berikut adalah penjelasan dan fungsi dari setiap perintah pada skrip Load training and testing data.

```
[ ] X_train, y_train = load_images(train_path)
    X_test, y_test = load_images(test_path)
```

Gambar 7. Load training and testing data

Berikut adalah penjelasan fungsi untuk memuat data pelatihan dan pengujian yang digunakan dalam proses model.

- `X_train, y_train = load_images(train_path):` Memuat dan memproses data serta label dari dataset pelatihan.
- `X_test, y_test = load_images(test_path):` Memuat dan memproses data serta label dari dataset pengujian.

Berikut adalah penjelasan dan fungsi dari setiap perintah pada skrip Feature Scaling dalam pemrosesan data.

```
[ ] scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
```

Gambar 8. Feature Scaling

Berikut adalah penjelasan fungsi dari Feature Scaling untuk menyalurkan skala data dalam pemrosesan model.

- `scaler = StandardScaler():` Fungsi untuk standarisasi data.
- `X_train = scaler.fit_transform(X_train):` Menghitung mean & standar deviasi lalu menstandarisasi data pelatihan (mean = 0, std = 1).
- `X_test = scaler.transform(X_test):` Menstandarisasi data pengujian menggunakan statistik dari data pelatihan untuk menghindari data leakage.

Berikut adalah penjelasan dan fungsi dari setiap perintah pada skrip SVM dengan berbagai kernel.

```
[ ] kernels = ['linear', 'rbf', 'poly', 'sigmoid']
    for kernel in kernels:
        svm_classifier = SVC(kernel=kernel)
        svm_classifier.fit(X_train, y_train)
        svm_predictions = svm_classifier.predict(X_test)
```

Gambar 9. SVM with different kernels

Berikut adalah penjelasan fungsi SVM dengan berbagai kernel untuk mengklasifikasi data sesuai karakteristiknya.

- `kernels:` List jenis kernel SVM (linear, rbf, poly, sigmoid).
- `linear:` Memisahkan data dengan garis lurus.
- `rbf:` Memetakan data ke dimensi lebih tinggi untuk pemisahan non-linear.
- `poly:` Menggunakan fungsi polinomial untuk pemisahan data.
- `sigmoid:` Mirip fungsi aktivasi dalam jaringan saraf.

- `for kernel in kernels:` Loop untuk menguji setiap kernel SVM.
- `svm_classifier = SVC(kernel=kernel):` Membuat model SVM dengan kernel yang sedang diuji.
- `svm_classifier.fit(X_train, y_train):` Melatih model dengan data pelatihan.
- `svm_predictions = svm_classifier.predict(X_test):` Memprediksi label data uji menggunakan model yang telah dilatih.

Berikut adalah penjelasan mengenai skrip Python yang digunakan untuk mengevaluasi performa model SVM.

```
[ ] svm_accuracy = accuracy_score(y_test, svm_predictions)
    svm_precision = precision_score(y_test, svm_predictions,
    average='weighted', zero_division=1)
    svm_recall = recall_score(y_test, svm_predictions,
    average='weighted', zero_division=1)
    print(f"\nSVM Results (Kernel: {kernel}):")
    print(f"Accuracy: {svm_accuracy}")
    print(f"Precision: {svm_precision}")
    print(f"Recall: {svm_recall}")
```

Gambar 10. Evaluate SVM

Berikut adalah penjelasan dan fungsi dari setiap perintah pada skrip untuk mengevaluasi model SVM.

- `svm_accuracy = accuracy_score(y_test, svm_predictions):` Menghitung akurasi model (persentase prediksi benar).
- `svm_precision = precision_score(y_test, svm_predictions, average='weighted', zero_division=1):` Menghitung presisi (proporsi prediksi positif yang benar).
- `svm_recall = recall_score(y_test, svm_predictions, average='weighted', zero_division=1):` Menghitung recall (kemampuan model menangkap data positif).
- `print(f"\nSVM Results (Kernel: {kernel}):"):` Menampilkan jenis kernel yang digunakan.
- `print(f"Accuracy: {svm_accuracy}):` Menampilkan nilai akurasi.
- `print(f"Precision: {svm_precision}):` Menampilkan nilai presisi.
- `print(f"Recall: {svm_recall}):` Menampilkan nilai recall.

Berikut adalah penjelasan skrip Python yang digunakan untuk membuat Confusion Matrix pada model SVM.

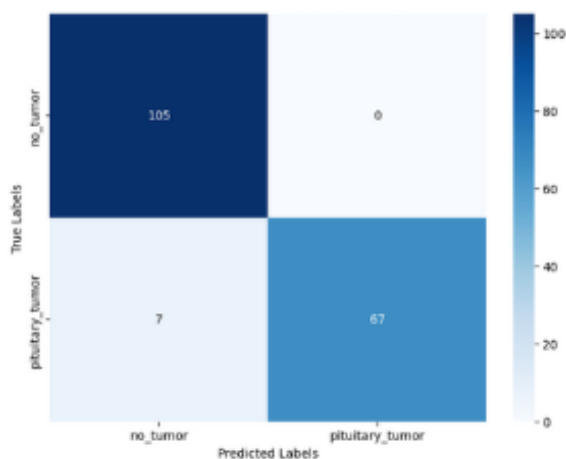
```
[ ] svm_cm = confusion_matrix(y_test, svm_predictions)
    plt.figure(figsize=(8, 6))
    sns.heatmap(svm_cm, annot=True, fmt="d", cmap="Blues",
    xticklabels=np.unique(y_test), yticklabels=np.unique(y_test))
    plt.title(f"SVM Confusion Matrix (Kernel: {kernel})")
    plt.xlabel("Predicted Labels")
    plt.ylabel("True Labels")
    plt.show()
```

Gambar 11. Confusion Matrix for SVM

Berikut adalah penjelasan dan fungsi dari setiap perintah pada skrip Confusion Matrix untuk model SVM.

- a. `svm_cm = confusion_matrix(y_test, svm_predictions)`: Menghitung matriks kebingungan untuk membandingkan label asli dan prediksi.
- b. `plt.figure(figsize=(8,6))`: Membuat figure baru dengan ukuran 8x6 inci.
- c. `sns.heatmap(svm_cm, annot=True, fmt="d", cmap="Blues", xticklabels=np.unique(y_test), yticklabels=np.unique(y_test))`:
- d. Menampilkan matriks kebingungan dalam bentuk heatmap.
- e. Angka ditampilkan di setiap sel (`annot=True`).
- f. Warna biru digunakan (`cmap="Blues"`).
- g. `plt.title(f"SVM Confusion Matrix (Kernel: {kernel})")`: Menambahkan judul yang mencantumkan jenis kernel.
- h. `plt.xlabel("Predicted Labels") / plt.ylabel("True Labels")`: Memberi label pada sumbu x dan y.
- i. `plt.show()`: Menampilkan heatmap.

Berdasarkan konfigurasi model terbaik yang dibentuk tersebut mendapatkan confusion matrix yang tersaji pada Gambar 12.



Gambar 12. *Confusion Matrix*

Confusion matrix yang ditampilkan pada Gambar 4, sebelah kiri atas merupakan True Negative (TN) sejumlah 105, sebelah kanan atas merupakan False Positive (FP) sejumlah 0, sebelah kiri bawah merupakan False Negative (FN) sejumlah 7, dan sebelah kanan bawah merupakan True Positive (TP) sejumlah 67. Di mana pada sumbu X dan Y terdapat kelas no-tumor dan kelas pituitary_tumor.

5. KESIMPULAN DAN SARAN

Hasil dari model SVM pada proyek ini menunjukkan bahwa klasifikasi menghasilkan class positif sebesar 0,95 dan class negatif sebesar 0,05. Model ini mencapai akurasi 81%, dengan precision 100% dan recall 80%, menunjukkan bahwa SVM memiliki performa yang baik dalam analisis klasifikasi citra tumor otak. Untuk pengembangan lebih lanjut, disarankan untuk mengeksplorasi metode optimasi seperti penyesuaian hyperparameter, augmentasi data,

atau membandingkan performa SVM dengan algoritma lain seperti Logistic Regression atau Artificial Neural Networks (ANN) guna meningkatkan akurasi dan pemahaman yang lebih mendalam terhadap data yang dianalisis.

DAFTAR PUSTAKA

- [1] Alzghaibi, H. (2023). Faculty and students' perspective of health informatics programs in saudi arabia. *Advances in Medical Education and Practice*, Volume 14, 1193-1207. <https://doi.org/10.2147/amep.s413742>
- [2] Chiu, S. (2023). Machine learning approaches to detect brain tumors from magnetic resonance imaging scans. *Journal of Student Research*, 12(4). <https://doi.org/10.47611/jsrhs.v12i4.5601>
- [3] Eldin, W. (2023). Ai-driven medical imaging platform: advancements in image analysis and healthcare diagnosis. *Journal of the Acs Advances in Computer Science*, 14(1), 0-0. <https://doi.org/10.21608/asc.2024.248278.1018>
- [4] Elhadad, A. (2024). Reduction of nifti files storage and compression to facilitate telemedicine services based on quantization hiding of downsampling approach. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-54820-4>
- [5] Jacobs, M., Umbricht, C., Parekh, V., Khouli, R., Cope, L., Macura, K., ... & Wolff, A. (2020). Integrated multiparametric radiomics and informatics system for characterizing breast tumor characteristics with the oncotypedx gene assay. *Cancers*, 12(10), 2772. <https://doi.org/10.3390/cancers12102772>
- [6] Li, M., Chan, M., Zhou, X., & Qian, X. (2020). Dc-al gan: pseudoprogression and true tumor progression of glioblastoma multiform image classification based on dcan and alexnet. *Medical Physics*, 47(3), 1139-1150. <https://doi.org/10.1002/mp.14003>
- [7] Panayides, A., Amini, A., Filipović, N., Sharma, A., Tsaftaris, S., Young, A., ... & Pattichis, C. (2020). Ai in medical imaging informatics: current challenges and future directions. *Ieee Journal of Biomedical and Health Informatics*, 24(7), 1837-1857. <https://doi.org/10.1109/jbhi.2020.2991043>
- [8] Rahman, A. (2024). Advanced ai-driven approach for enhanced brain tumor detection from mri images utilizing efficientnetb2 with equalization and homomorphic filtering. *BMC Medical Informatics and Decision Making*, 24(1). <https://doi.org/10.1186/s12911-024-02519-x>
- [9] (2022). A deep learning based framework automatic brain tumors classification- a review. *International Research Journal of Modernization in Engineering Technology and Science*. <https://doi.org/10.56726/irjmet31921>

- [10] Fayaz, M., Torokeldiev, N., Turdumamatov, S., Qureshi, M., Qureshi, M., & Gwak, J. (2021). An efficient methodology for brain mri classification based on dwt and convolutional neural network. *Sensors*, 21(22), 7480. <https://doi.org/10.3390/s21227480>
- [11] Liu, L., Jing, C., Wang, Y., Liang, G., Wang, Y., & Zhang, H. (2022). Decomposition-based correlation learning for multi-modal mri-based classification of neuropsychiatric disorders. *Frontiers in Neuroscience*, 16. <https://doi.org/10.3389/fnins.2022.832276>
- [12] Sarhan, A. (2020). Detection and classification of brain tumor in mri images using wavelet transform and convolutional neural network. *Journal of Advances in Medicine and Medical Research*, 15-26. <https://doi.org/10.9734/jammr/2020/v32i123053>
- [13] Susanto, A., Sari, C., Rahmalan, H., & Doheir, M. (2023). Support vector machine based discrete wavelet transform for magnetic resonance imaging brain tumor classification. *Telkomnika (Telecommunication Computing Electronics and Control)*, 21(3), 592. <https://doi.org/10.12928/telkomnika.v21i3.2492>
- [14] Yu, Z., Qinghu, H., Yang, J., & Luo, M. (2022). A supervised ml applied classification model for brain tumors mri. *Frontiers in Pharmacology*, 13. <https://doi.org/10.3389/fphar.2022.884495>
- [15] (2023). Review of deep learning-driven mri brain tumor detection and segmentation methods. *Advances in Computer Signals and Systems*, 7(8), 17-28. <https://doi.org/10.23977/acss.2023.070803>
- [16] Basir, O. and Shantta, K. (2021). Automatic mri brain tumor segmentation techniques: a survey. *Ira-International Journal of Applied Sciences (Issn 2455-4499)*, 16(2), 25. <https://doi.org/10.21013/jas.v16.n2.p2>
- [17] Biratu, E., Schwenker, F., Ayano, Y., & Debelee, T. (2021). A survey of brain tumor segmentation and classification algorithms. *Journal of Imaging*, 7(9), 179. <https://doi.org/10.3390/jimaging7090179>
- [18] Breto, A. (2023). A deep learning approach for automatic segmentation during daily mri-linac radiotherapy of glioblastoma. *Cancers*, 15(21), 5241. <https://doi.org/10.3390/cancers15215241>
- [19] Breto, A., Spieler, B., Zavala-Romero, O., Alhusseini, M., Patel, N., Asher, D., ... & Portelance, L. (2022). Deep learning for per-fraction automatic segmentation of gross tumor volume (gtv) and organs at risk (oars) in adaptive radiotherapy of cervical cancer. *Frontiers in Oncology*, 12. <https://doi.org/10.3389/fonc.2022.854349>
- [20] Fawzi, A., Achuthan, A., & Belaton, B. (2021). Brain image segmentation in recent years: a narrative review. *Brain Sciences*, 11(8), 1055. <https://doi.org/10.3390/brainsci11081055>
- [21] Latif, U., Raza, B., Ziauddin, S., & Khan, M. (2021). An end-to-end brain tumor segmentation system using multi-inception-unet. *International Journal of Imaging Systems and Technology*, 31(4), 1803-1816. <https://doi.org/10.1002/ima.22585>