

IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI DAN DETEKSI TUMOR OTAK

Ganis Sanhaji, Decky Putra Kurnia, dan Vito Dwi Nur Hidayat

Fakultas Teknik, Program Studi Teknik Elektro, Universitas Islam Nusantara, Kota Bandung, Indonesia
Email:ganissanhaji90@gmail.com, deckyputrakurnia2304@gmail.com, vitodwinurhidayat03@gmail.com

Abstrak– Penelitian ini mengimplementasikan algoritma Convolutional Neural Network (CNN) untuk mendeteksi tumor otak melalui citra digital. Tumor otak merupakan salah satu penyakit dengan tingkat kompleksitas tinggi yang memerlukan diagnosis cepat dan akurat. CNN digunakan karena kemampuannya untuk mengekstraksi pola kompleks dari citra tanpa memerlukan rekayasa fitur manual. Dataset yang digunakan mencakup empat kategori utama: glioma, meningioma, pituitary, dan non-tumor. Model dilatih menggunakan teknik transfer learning untuk meningkatkan efisiensi pelatihan dan akurasi prediksi. Evaluasi model menunjukkan hasil yang menjanjikan dengan tingkat akurasi sebesar 88%, menggunakan metrik seperti precision, recall, F1-score, dan Confusion Matrix untuk analisis kinerja. Hasil penelitian ini diharapkan dapat mendukung pengembangan sistem berbasis kecerdasan buatan untuk membantu diagnosis medis secara otomatis, terutama di wilayah dengan akses terbatas terhadap fasilitas diagnostik dan tenaga medis.

Kata Kunci: CNN, Tumor Otak, Diagnosis Otomatis, Confusion Matrix

Abstract– This study implements the Convolutional Neural Network (CNN) algorithm to detect brain tumors through digital images. Brain tumors are highly complex conditions requiring rapid and accurate diagnosis. CNN is employed due to its ability to extract complex patterns from images without the need for manual feature engineering. The dataset comprises four main categories: glioma, meningioma, pituitary, and non-tumor. The model was trained using transfer learning techniques to enhance training efficiency and prediction accuracy. Evaluation results demonstrate promising outcomes with an accuracy rate of 88%, analyzed using metrics such as precision, recall, F1-score, and Confusion Matrix. This research is expected to support the development of artificial intelligence-based systems to assist in automatic medical diagnoses, particularly in areas with limited access to diagnostic facilities and medical professionals.

Keywords: CNN, Brain Tumor, Automatic Diagnosis, Confusion Matrix

1. PENDAHULUAN

Tumor otak merupakan salah satu penyakit dengan tingkat kompleksitas tinggi yang memberikan dampak signifikan terhadap kualitas hidup penderitanya. Penyakit ini terjadi akibat pertumbuhan sel yang tidak terkendali di jaringan otak, yang dapat bersifat jinak maupun ganas. Meskipun tumor jinak biasanya tidak menyebar ke bagian tubuh lain, keduanya memerlukan perhatian khusus karena dapat mengganggu fungsi vital otak dalam mengendalikan aktivitas tubuh manusia. Berdasarkan data dari World Health Organization (WHO), angka kejadian tumor otak terus meningkat setiap tahun, baik di negara maju maupun berkembang, termasuk Indonesia. Namun, rendahnya kesadaran masyarakat dan keterbatasan akses terhadap teknologi diagnostik modern menjadi hambatan besar dalam mendeteksi tumor otak pada tahap awal, sehingga meningkatkan risiko komplikasi serius pada pasien. [1]

Saat ini, metode diagnosis tumor otak masih didominasi oleh teknik pencitraan medis seperti Magnetic Resonance Imaging (MRI) dan Computed Tomography (CT) scan. Teknologi MRI memiliki keunggulan dalam menghasilkan citra otak dengan resolusi tinggi, sementara CT scan memberikan hasil yang lebih cepat dengan memanfaatkan radiasi ion. Kendati demikian, kedua metode ini memiliki sejumlah keterbatasan, seperti biaya tinggi, kebutuhan akan operator berpengalaman, dan waktu yang relatif lama dalam proses interpretasi hasil. Selain itu, proses manual yang dilakukan oleh tenaga medis dalam membaca hasil citra MRI atau CT scan sering kali rentan terhadap kesalahan karena bersifat subjektif. Hal ini menjadi kendala utama dalam menyediakan diagnosis yang cepat dan akurat, terutama di wilayah dengan fasilitas kesehatan yang terbatas. Oleh karena itu, diperlukan solusi yang lebih efisien, otomatis, dan dapat diakses secara luas untuk mendukung deteksi dini tumor otak.[2]

Glioma, meningioma, dan tumor pituitari adalah tiga jenis tumor otak yang paling sering ditemukan. Glioma mencakup sekitar 78% dari total kasus tumor otak ganas dan berkembang di sel-sel glial yang mendukung fungsi neuron. Meningioma, yang menyumbang sekitar 36,8% dari kasus tumor otak histologi, tumbuh di meninges atau lapisan pelindung otak dan sumsum tulang belakang. Sementara itu, tumor pituitari berkembang di kelenjar pituitari yang mengatur fungsi hormon dalam tubuh. Ketiga jenis tumor ini memiliki karakteristik dan lokasi yang berbeda, sehingga memerlukan metode diagnosis yang tepat untuk mengidentifikasinya.[3]

Diagnosis tumor otak secara umum dilakukan melalui teknik MRI atau CT scan. Meskipun teknik ini memiliki keunggulan tertentu, seperti hasil citra beresolusi tinggi dari MRI dan kecepatan hasil dari CT scan, keduanya tetap memiliki keterbatasan. Subjektivitas dalam interpretasi citra, kebutuhan akan operator terlatih, dan keterbatasan akses menjadi kendala utama, terutama di wilayah dengan fasilitas kesehatan yang terbatas. Oleh

karena itu, teknologi berbasis AI, seperti Convolutional Neural Network (CNN), diharapkan dapat menjadi solusi untuk mengatasi tantangan ini.

Kemajuan teknologi kecerdasan buatan (Artificial Intelligence, AI) telah membuka peluang besar di bidang medis, terutama dalam analisis citra medis. Salah satu pendekatan AI yang paling menonjol adalah CNN, yaitu algoritma pembelajaran mendalam yang dirancang untuk mengenali pola kompleks dalam data gambar. CNN telah terbukti sangat efektif dalam berbagai aplikasi medis, seperti deteksi kanker paru-paru, klasifikasi lesi kulit, dan identifikasi penyakit mata seperti glaukoma. Keunggulan utama CNN adalah kemampuannya untuk secara otomatis mempelajari fitur penting dari data citra tanpa memerlukan rekayasa fitur manual. Dengan pendekatan ini, CNN dapat menghasilkan tingkat akurasi yang tinggi dalam klasifikasi dan prediksi, menjadikannya alat yang potensial untuk membantu tenaga medis dalam menganalisis tumor otak. [4]

Implementasi CNN dalam mendeteksi tumor otak menawarkan berbagai manfaat. Sistem berbasis CNN dapat mempercepat proses diagnosis, mengurangi beban kerja tenaga medis, dan meningkatkan efisiensi dalam pengambilan keputusan klinis. Sistem ini juga mampu memberikan hasil yang lebih konsisten dibandingkan metode konvensional, sehingga pasien dapat memperoleh diagnosis awal dengan lebih cepat. Hal ini sangat penting untuk kasus-kasus mendesak, di mana waktu sangat berharga dalam menentukan langkah pengobatan. Dengan adanya teknologi ini, diharapkan kualitas layanan kesehatan dapat ditingkatkan, terutama di daerah dengan keterbatasan fasilitas medis.[5]

Penelitian sebelumnya menunjukkan hasil yang menjanjikan dalam penerapan CNN untuk klasifikasi tumor otak menggunakan dataset MRI. Beberapa studi melaporkan akurasi hingga 95% dalam membedakan jenis tumor seperti glioma, meningioma, pituitary, dan kondisi tanpa tumor. Pendekatan transfer learning sering kali digunakan untuk meningkatkan efisiensi dan akurasi hasil. Dengan memanfaatkan model CNN yang telah dilatih sebelumnya, transfer learning memungkinkan pelatihan pada dataset yang lebih kecil dengan hasil yang tetap optimal. Teknik ini juga mengurangi kebutuhan akan sumber daya komputasi yang besar. Dengan demikian, pendekatan ini diharapkan mampu menghasilkan model yang lebih cepat dan akurat untuk klasifikasi tumor otak.

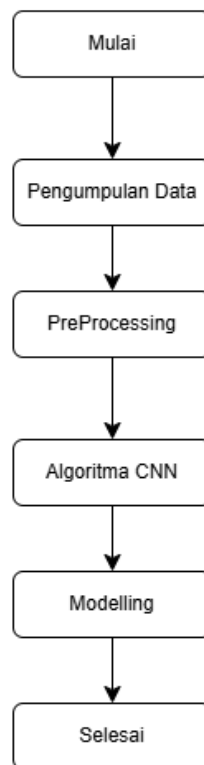
Penelitian ini bertujuan untuk mengembangkan sistem pendeteksi tumor otak berbasis CNN yang mampu menganalisis citra MRI secara otomatis. Model yang dirancang akan dilatih menggunakan dataset citra tumor otak yang telah diklasifikasikan ke dalam beberapa kategori, seperti glioma, meningioma, dan tumor pituitari. Dengan pendekatan ini, diharapkan dapat dihasilkan model dengan tingkat akurasi yang tinggi, sehingga dapat diandalkan sebagai alat bantu diagnosis oleh tenaga medis. Selain itu, penelitian ini juga diharapkan menjadi langkah awal dalam penerapan teknologi kecerdasan buatan untuk berbagai aplikasi medis lainnya, yang pada akhirnya akan memberikan manfaat besar dalam meningkatkan kualitas layanan kesehatan.[6]

Melalui penelitian ini, diharapkan tercipta solusi teknologi yang mampu menjembatani kesenjangan antara kebutuhan medis dan keterbatasan fasilitas kesehatan. Dengan mengintegrasikan algoritma CNN yang canggih dan dataset MRI yang berkualitas, penelitian ini diharapkan dapat memberikan kontribusi signifikan terhadap pengembangan sistem diagnostik berbasis AI. Selain mendukung deteksi dini tumor otak, penelitian ini juga membuka peluang untuk eksplorasi lebih lanjut dalam berbagai aplikasi medis lainnya, sehingga memberikan dampak positif yang lebih luas di masa depan.[7]

2. METODE PENELITIAN

2.1 Tahap Penelitian

Pendekatan ini menjelaskan secara rinci setiap langkah yang diperlukan dalam implementasi algoritma Convolutional Neural Network (CNN) untuk klasifikasi dan deteksi tumor otak. Proses ini melibatkan pengumpulan data berupa citra otak yang relevan dan penting, yang kemudian diproses menjadi informasi yang dapat digunakan untuk mengklasifikasikan jenis tumor. Semua tahapan dalam prosedur ini divisualisasikan melalui diagram alur penelitian yang disajikan pada Gambar 1.



Gambar 1. Flowchart Penelitian

Dari Gambar 3.1 tahapan implementasi algoritma Convolutional Neural Network (CNN) untuk mendeteksi dan mengklasifikasikan tumor otak. Proses dimulai dari tahap pengumpulan data, di mana dataset berupa gambar otak dikumpulkan sebagai sumber utama untuk analisis. Setelah itu, data melalui tahap preprocessing untuk memastikan bahwa gambar berada dalam format dan ukuran yang konsisten, mempermudah pemrosesan lebih lanjut.

Tahap berikutnya adalah penerapan algoritma CNN, di mana jaringan neural mempelajari pola dan fitur pada data untuk melakukan klasifikasi. Hasil dari algoritma ini digunakan dalam tahap pemodelan untuk membuat sistem yang mampu mendeteksi keberadaan tumor berdasarkan input gambar baru. Setelah semua tahap selesai, model akhir dapat digunakan untuk prediksi dan analisis tumor otak secara efisien. Proses ini memastikan pendekatan sistematis dan akurat dalam pengembangan model berbasis CNN.

1. Penghitungan Jumlah Data

```

... Jumlah gambar di direktori Training:
- glioma: 1321 gambar
- meningioma: 1339 gambar
- notumor: 1595 gambar
- pituitary: 1457 gambar

Jumlah gambar di direktori Testing:
- glioma: 300 gambar
- meningioma: 306 gambar
- notumor: 405 gambar
- pituitary: 300 gambar
  
```

Gambar 2. Menghitung jumlah data

Pada tahap awal, kode menentukan direktori dataset dengan menggabungkan path dasar (base_dir) dengan nama folder Training dan Testing. Hal ini dilakukan menggunakan fungsi `os.path.join()`, yang memastikan path dapat dikenali oleh sistem operasi.

Selanjutnya, fungsi `hitung_gambar_per_folder()` dibuat untuk menghitung jumlah file di setiap folder. Fungsi ini memindai seluruh sub-folder di dalam direktori yang diberikan, memeriksa apakah sub-folder tersebut adalah direktori yang valid, lalu menghitung jumlah file dalam folder tersebut. Hasil perhitungan disimpan dalam bentuk dictionary, di mana nama folder digunakan sebagai kunci dan jumlah file sebagai nilainya.

Setelah fungsi tersebut didefinisikan, kode menghitung jumlah file di dalam folder Training dan Testing dengan memanggil fungsi `hitung_gambar_per_folder()` untuk masing-masing direktori. Hasilnya disimpan dalam variabel `gambar_training` dan `gambar_testing`.

Terakhir, hasil perhitungan ditampilkan dengan mencetak jumlah file di setiap folder dalam format yang rapi. Kode menggunakan loop untuk mengakses pasangan nama folder dan jumlah file dalam dictionary hasil perhitungan, lalu menampilkannya di layar.

Kode ini membantu memastikan bahwa dataset terorganisir dengan baik sebelum digunakan untuk pelatihan model, serta memberikan informasi penting terkait distribusi data di setiap kategori. Ini penting untuk mengidentifikasi potensi ketidakseimbangan data yang dapat memengaruhi kinerja model.

2. Load Data

```

x_train = np.array(x_train)
y_train = np.array(y_train)

```

100%	1321/1321	[00:03<00:00, 340.85it/s]
100%	1339/1339	[00:05<00:00, 237.65it/s]
100%	1595/1595	[00:05<00:00, 290.01it/s]
100%	1457/1457	[00:08<00:00, 170.87it/s]
100%	300/300	[00:00<00:00, 322.44it/s]
100%	306/306	[00:01<00:00, 298.22it/s]
100%	405/405	[00:00<00:00, 497.38it/s]
100%	300/300	[00:01<00:00, 292.28it/s]

Gambar 3. Load Data

Proses diawali dengan pembuatan dua list kosong, yaitu `x_train` untuk menyimpan gambar dalam bentuk array dan `y_train` untuk menyimpan label kategori. Variabel labels didefinisikan untuk menetapkan daftar kategori dataset, seperti 'glioma', 'meningioma', 'notumor', dan 'pituitary'. Gambar-gambar dari folder kategori dibaca dalam loop dengan memanfaatkan fungsi `os.listdir()`, sedangkan jalur file digabung menggunakan fungsi `os.path.join()`. Setiap gambar dibuka menggunakan library OpenCV melalui `cv2.imread` dan disesuaikan ukurannya menjadi 256x256 piksel dengan `cv2.resize()`. Gambar yang telah diproses ini kemudian ditambahkan ke list `x_train`, sementara indeks dari kategori yang sesuai dimasukkan ke dalam `y_train` sebagai label.

Langkah serupa dilakukan untuk data yang terdapat di direktori Testing (`val_dir`), sehingga data pengujian juga dipersiapkan dengan cara yang sama. Setelah seluruh data selesai diproses, list `x_train` dan `y_train` dikonversi ke array menggunakan library NumPy (`np.array()`), memastikan formatnya kompatibel untuk digunakan dalam model machine learning. Hasil dari kode ini adalah dataset gambar yang telah diproses, diubah ukurannya, dan dikelompokkan berdasarkan kategorinya masing-masing, sehingga data tersebut siap dimanfaatkan dalam proses pelatihan maupun pengujian model.

2.2. Studi Literature

Machine Learning (ML) adalah salah satu cabang dari kecerdasan buatan (Artificial Intelligence, AI) yang dirancang untuk memungkinkan komputer atau sistem untuk belajar dari data tanpa pemrograman eksplisit. Melalui pendekatan ini, sistem dapat menganalisis data dalam jumlah besar, mengidentifikasi pola, dan membuat prediksi atau keputusan berdasarkan data tersebut. Konsep ML ini menjadi semakin relevan di era modern, terutama karena meningkatnya kebutuhan akan solusi berbasis data di berbagai sektor seperti kesehatan, pendidikan, hingga perubahan iklim.[8]

Machine Learning (ML) ML adalah subbidang AI yang fokus pada pelatihan mesin untuk belajar dari data tanpa memerlukan instruksi eksplisit. Dalam klasifikasi tumor otak, ML digunakan untuk mempelajari pola dari dataset citra MRI dan menerapkan pengetahuan ini untuk mendeteksi dan mengklasifikasikan jenis tumor dengan tingkat akurasi yang tinggi.

Deep Learning adalah cabang dari pembelajaran mesin (Machine Learning) yang menggunakan arsitektur jaringan saraf tiruan (Artificial Neural Networks) dengan lapisan yang lebih dalam dan kompleks. Teknologi ini dirancang untuk meniru cara otak manusia memproses informasi, memungkinkan komputer untuk mempelajari pola dari data yang sangat besar dan heterogen. Dengan menggunakan lapisan non-linear berganda, Deep Learning dapat menyelesaikan masalah yang membutuhkan analisis kompleks, seperti pengenalan suara, pemrosesan gambar, dan pemahaman bahasa alami [9]. Deep Learning (DL) DL adalah pengembangan dari ML yang memanfaatkan jaringan saraf tiruan dengan banyak lapisan (multi-layer neural networks). Metode ini sangat efektif untuk tugas-tugas kompleks seperti analisis citra medis. DL secara otomatis mengekstrak fitur dari data citra tanpa memerlukan rekayasa fitur manual, sehingga sangat cocok untuk deteksi dan klasifikasi tumor otak.

Convolutional Neural Network (CNN) adalah salah satu arsitektur dalam deep learning yang dirancang khusus untuk memproses data berbentuk gambar atau data spasial lainnya. CNN terinspirasi dari cara kerja visual korteks pada otak manusia, yang mampu mengenali pola visual dengan sangat efisien. Dalam konteks analisis citra medis, seperti diagnosis tumor otak dari citra MRI, CNN telah membuktikan efektivitasnya dalam mengenali pola kompleks yang sulit diidentifikasi oleh manusia secara manual [10]. Convolutional Neural Network (CNN) adalah varian dari deep learning yang dirancang khusus untuk analisis citra. CNN bekerja dengan mengekstrak fitur dari citra melalui lapisan konvolusi, pooling, dan fully connected layer. Dalam klasifikasi tumor otak, CNN mampu mengenali pola dan karakteristik spesifik pada citra MRI, sehingga memungkinkan pengelompokan jenis tumor dengan akurasi tinggi. CNN telah digunakan secara luas dalam penelitian medis, dengan hasil yang sangat menjanjikan dalam klasifikasi, segmentasi, dan deteksi tumor otak. [11]

Confusion Matrix adalah alat evaluasi yang sering digunakan untuk mengukur performa model klasifikasi. Matriks ini menyajikan informasi tentang jumlah prediksi yang benar dan salah yang dibuat oleh model, dengan membandingkannya terhadap label aktual. Dalam klasifikasi multilabel, confusion matrix sangat penting untuk menganalisis kemampuan model dalam membedakan setiap kelas [12]. Confusion Matrix adalah metode evaluasi yang digunakan untuk mengukur performa model klasifikasi. Matriks ini mencakup metrik seperti true positive (TP), true negative (TN), false positive (FP), dan false negative (FN), yang membantu dalam menghitung akurasi, presisi, recall, F1-score, dan loss. Dalam konteks klasifikasi tumor otak, confusion matrix digunakan untuk mengevaluasi sejauh mana model dapat mengklasifikasikan jenis tumor secara benar.

Komponen Confusion Matrix

- True Positive (TP): Prediksi benar untuk kelas positif.
- True Negative (TN): Prediksi benar untuk kelas negatif.
- False Positive (FP): Prediksi salah, di mana model memprediksi positif tetapi sebenarnya negatif.
- False Negative (FN): Prediksi salah, di mana model memprediksi negatif tetapi sebenarnya positif.

a. Akurasi

$$Akurasi = \frac{TP}{TP + TN + FP + FN} \times 100\%$$

b. Presisi

$$Presisi = \frac{TP}{TP + FP} \times 100\%$$

c. Recall

$$Recall = \frac{TP}{TP + FN} \times 100\%$$

2.2 Analisis Data

2.2.1. Teknik Pengumpulan Data

Teknik pengumpulan data yang kami lakukan merujuk pada beragam strategi dan metode yang diterapkan untuk menghimpun informasi yang relevan dengan tujuan studi. Kami menggunakan dataset yang diperoleh dari jurnal yang telah kami pilih. Dataset ini mencakup citra MRI otak dari berbagai jenis tumor, yaitu Glioma, Meningioma, No Tumor, dan Pituitary, yang telah diklasifikasikan ke dalam data training sebanyak 5.712 data dan data testing sebanyak 1.311 data. Data tersebut diolah dan disiapkan dalam format yang sesuai untuk dilakukan proses pelatihan model menggunakan algoritma Convolutional Neural Network (CNN). Rincian jumlah data dapat dilihat pada Tabel 1 dan Tabel 2.

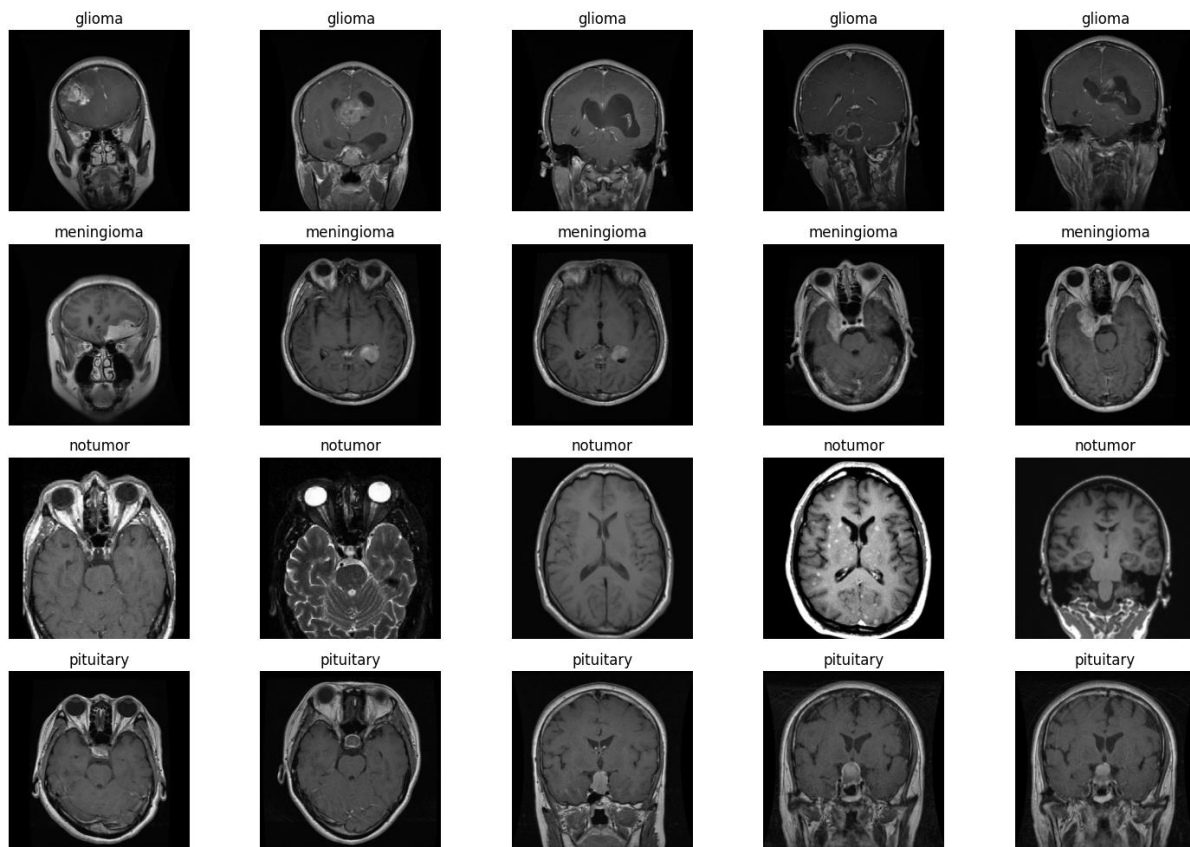
Tabel 1. Jumlah Data Training

Data	Jumlah
Glioma	1.321
Meningioma	1.339
Notumor	1.595
Pituitary	1.457
Total Jumlah	5.712

Tabel 2. Jumlah Data Testing

Data	Jumlah
Glioma	300
Meningioma	306
Notumor	405
Pituitary	300
Total Jumlah	1.311

Proses Pre-Processing dimulai dengan mendefinisikan jumlah sampel gambar yang akan ditampilkan (`num_samples`). Setiap kategori gambar (seperti glioma dan meningioma) divisualisasikan dalam sub-plot menggunakan fungsi `plt.subplot()`. Gambar yang dipilih dari array data ditampilkan menggunakan `plt.imshow()`, sedangkan axis gambar dinonaktifkan dengan `plt.axis('off')` untuk memberikan fokus pada citra saja. Judul pada setiap gambar ditambahkan dengan fungsi `plt.title()`, menampilkan nama kategori gambar.

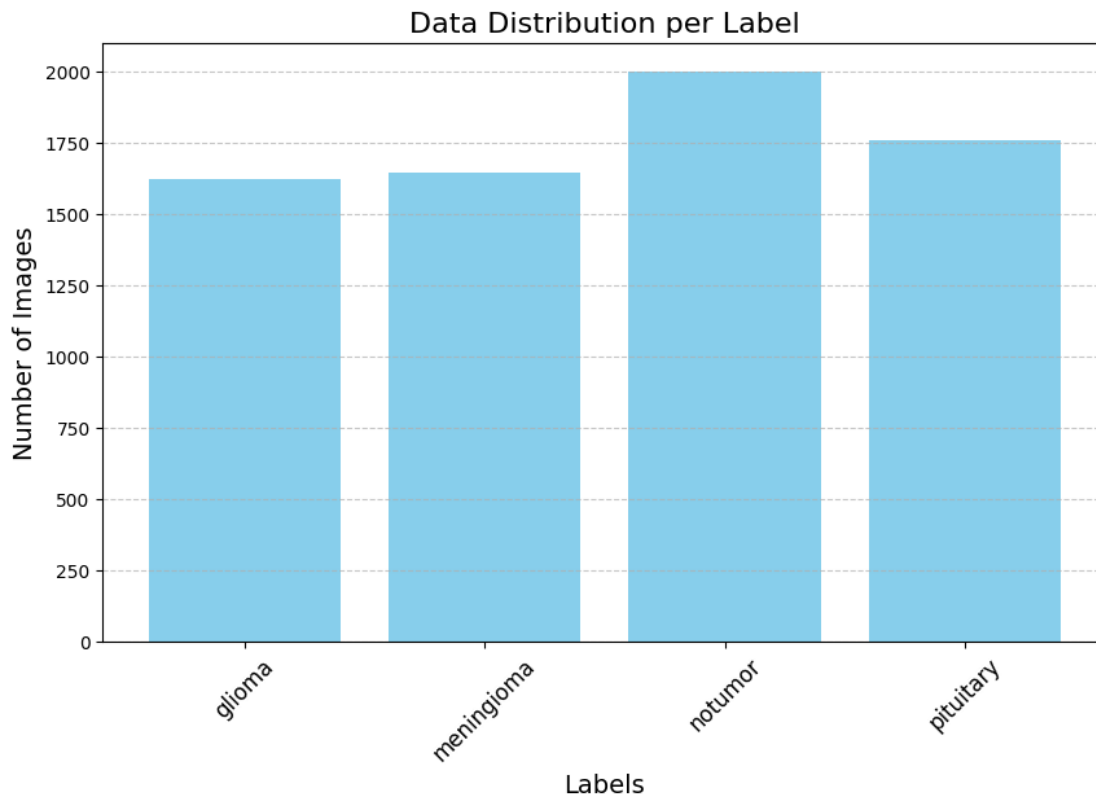


Gambar 4. Pre-Processing

Pengaturan tata letak subplot dilakukan menggunakan `plt.tight_layout()` untuk memastikan gambar tersusun rapi tanpa tumpang tindih. Akhirnya, fungsi `plt.show()` digunakan untuk menampilkan plot tersebut. Hasil yang terlihat adalah beberapa gambar MRI otak yang terbagi berdasarkan kategori, seperti glioma dan meningioma. Gambar ini membantu memvalidasi bahwa data telah diorganisasi dengan benar dan memberikan wawasan awal tentang variasi visual dalam dataset. Visualisasi ini penting untuk memastikan bahwa data sesuai dengan ekspektasi sebelum melanjutkan ke proses pelatihan model.

Kode ini dimulai dengan mengimpor library yang relevan, yaitu `matplotlib.pyplot` untuk membuat grafik dan `Counter` dari `collections` untuk menghitung jumlah elemen dalam dataset berdasarkan label. Fungsi `visualize_data_distribution` didefinisikan untuk menghitung jumlah gambar di setiap kategori dalam dataset, lalu menampilkan hasilnya dalam bentuk diagram batang (`bar chart`). Fungsi ini bekerja dengan menghitung frekuensi label dalam dataset menggunakan `Counter`. Hasil penghitungan tersebut kemudian diurutkan sesuai dengan daftar label yang telah ditentukan sebelumnya. Proses visualisasi dilakukan menggunakan fungsi `plt.bar()` untuk membuat diagram batang, dengan sumbu-x mewakili kategori label seperti glioma, meningioma, notumor, dan pituitary, serta sumbu-y menunjukkan jumlah gambar di setiap kategori.

Grafik diberi judul “Data Distribution per Label” untuk memberikan konteks yang jelas kepada pengguna. Label untuk sumbu-x dan sumbu-y juga ditambahkan untuk memperjelas isi grafik, sementara grid pada sumbu-y digunakan untuk membantu pengguna dalam membaca nilai jumlah gambar dengan lebih mudah. Ukuran grafik diatur menggunakan plt.figure() agar tampil proporsional, dan fungsi plt.show() digunakan untuk menampilkan hasil visualisasi



Gambar 5. Distribusi Data

Pada hasil visualisasi, terlihat bahwa jumlah gambar pada kategori NoTumor memiliki jumlah tertinggi dibandingkan kategori lainnya, sedangkan kategori lainnya seperti glioma dan meningioma memiliki jumlah yang lebih seimbang. Visualisasi ini sangat berguna untuk memahami distribusi data sebelum digunakan dalam pelatihan model. Jika terdapat ketidakseimbangan, pengguna dapat mengambil langkah seperti augmentasi data atau teknik lainnya untuk memastikan performa model tidak bias terhadap kategori tertentu. Kode ini memberikan gambaran yang jelas mengenai kesiapan dataset untuk proses analisis lebih lanjut.

```

print(f"y_train shape: {y_train.shape}")
[33] Python
...
y_train (one-hot encoded):
[[ True False False False]
 [ True False False False]
 [ True False False False]
 ...
 [False False False  True]
 [False False False  True]
 [False False False  True]]
y_train shape: (7023, 4)
    
```

Gambar 6. Hot Encoding

Tahap awal adalah mengonversi label data y_train ke format one-hot encoding menggunakan fungsi pd.get_dummies() dari library Pandas, dan hasilnya disimpan dalam array NumPy. Dalam format ini, setiap label diubah menjadi array biner, di mana setiap elemen array menunjukkan kategori tertentu dengan nilai 1 untuk kategori aktif dan 0 untuk kategori lainnya. Format ini sangat penting dalam model klasifikasi multi-kelas karena model deep learning bekerja lebih baik dengan representasi numerik seperti ini. Hasil dari konversi ditampilkan untuk memverifikasi bahwa data telah diubah dengan benar. Jumlah total data (7023 sampel) juga ditampilkan melalui atribut .shape. Selanjutnya, dataset dibagi menjadi tiga subset: training, testing, dan validation, menggunakan fungsi train_test_split dari library Scikit-learn. Subset training digunakan untuk melatih model, testing untuk mengevaluasi performa model setelah pelatihan, dan validation untuk mengukur kemampuan

generalisasi model selama proses pelatihan. Parameter `test_size` diatur untuk menentukan proporsi data yang dialokasikan ke subset testing dan validation (masing-masing 10%). Parameter `random_state` memastikan bahwa pembagian data dilakukan secara konsisten di setiap kali kode dijalankan. Setelah pembagian data, bentuk (shape) dari setiap subset data ditampilkan untuk memastikan bahwa data telah terpisah dengan benar. Misalnya, subset training memiliki bentuk (5688, 256, 256, 3) untuk fitur dan (5688, 4) untuk label, yang menunjukkan jumlah data, dimensi gambar (256x256 piksel dengan 3 saluran warna), dan jumlah kategori (4 kelas). Hal serupa dilakukan untuk subset testing dan validation.

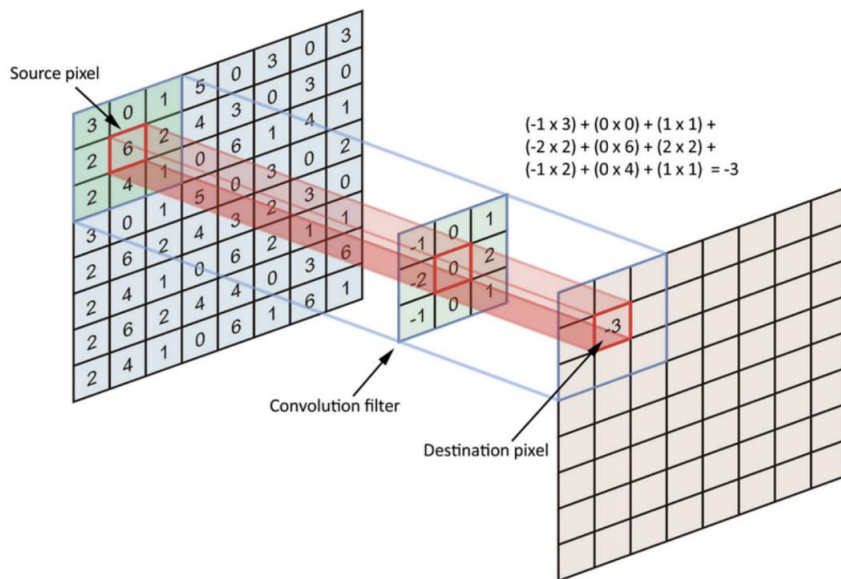
Proses ini memastikan bahwa dataset siap digunakan untuk melatih dan menguji model deep learning dengan format data yang sesuai. Validasi bentuk data juga penting untuk menghindari kesalahan selama proses pelatihan model. Visualisasi hasil pembagian data ini membantu mengonfirmasi bahwa dataset telah diolah sesuai kebutuhan analisis lebih lanjut.

```

Model: "sequential_4"
-----
Layer (type)                Output Shape                Param #
-----
Conv2D_Layer1 (Conv2D)      (None, 64, 64, 32)         2432
MaxPooling2D_Layer1 (MaxPo
oling2D)                    (None, 32, 32, 32)         0
Dropout_Layer1 (Dropout)    (None, 32, 32, 32)         0
Conv2D_Layer2 (Conv2D)      (None, 32, 32, 32)         25632
MaxPooling2D_Layer2 (MaxPo
oling2D)                    (None, 16, 16, 32)         0
Dropout_Layer2 (Dropout)    (None, 16, 16, 32)         0
Conv2D_Layer3 (Conv2D)      (None, 16, 16, 32)         9248
MaxPooling2D_Layer3 (MaxPo
oling2D)                    (None, 8, 8, 32)           0
Dropout_Layer3 (Dropout)    (None, 8, 8, 32)           0
...
Total params: 177284 (692.52 KB)
Trainable params: 177284 (692.52 KB)
Non-trainable params: 0 (0.00 Byte)
-----
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
    
```

Gambar 7. Output Arsitektur Layer

Arsitektur dimulai dengan layer konvolusi pertama (Conv2D), yang memiliki 32 filter dengan ukuran kernel tertentu dan menghasilkan output shape (None, 64, 64, 32). "None" pada dimensi batch menunjukkan bahwa ukuran batch dapat bervariasi, sedangkan dimensi lain menunjukkan ukuran spasial (64x64) dan jumlah filter (32). Setelah layer konvolusi, terdapat layer pooling (MaxPooling2D) yang mengurangi dimensi spasial menjadi (None, 32, 32, 32). Layer dropout juga digunakan untuk mencegah overfitting dengan secara acak menonaktifkan beberapa neuron selama pelatihan.



Gambar 7. Gambaran Arsitektur Layer

Proses ini diulangi dengan menambahkan layer konvolusi dan pooling tambahan, masing-masing dengan jumlah filter yang meningkat (misalnya, 32 diikuti oleh 64 filter). Penggunaan filter yang semakin banyak bertujuan untuk menangkap fitur yang lebih kompleks dari data gambar. Dropout terus digunakan setelah setiap blok konvolusi dan pooling untuk menjaga generalisasi model.

Jumlah total parameter yang dilatih model adalah 177,284, yang merupakan jumlah bobot dan bias yang akan diperbarui selama proses pelatihan. Model ini dirancang untuk mengekstraksi fitur dari data gambar secara bertahap, dari pola sederhana pada layer awal hingga pola yang lebih kompleks pada layer yang lebih dalam.

```

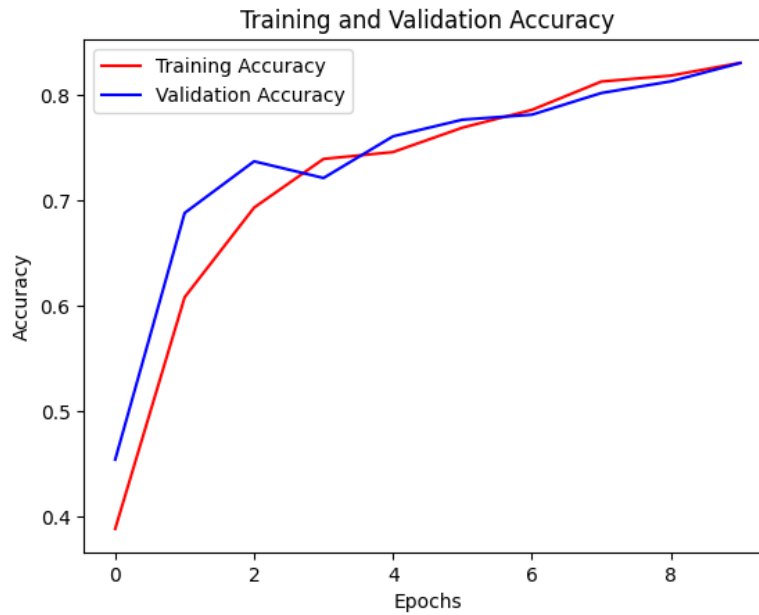
Epoch 1/10
178/178 [=====] - 61s 329ms/step - loss: 1.2362 - accuracy: 0.3884 - val_loss: 1.0895 - val_accuracy: 0.4541
Epoch 2/10
178/178 [=====] - 62s 347ms/step - loss: 0.9644 - accuracy: 0.6083 - val_loss: 0.8234 - val_accuracy: 0.6883
Epoch 3/10
178/178 [=====] - 60s 338ms/step - loss: 0.7872 - accuracy: 0.6934 - val_loss: 0.6957 - val_accuracy: 0.7373
Epoch 4/10
178/178 [=====] - 61s 342ms/step - loss: 0.6908 - accuracy: 0.7396 - val_loss: 0.6520 - val_accuracy: 0.7215
Epoch 5/10
178/178 [=====] - 62s 346ms/step - loss: 0.6598 - accuracy: 0.7461 - val_loss: 0.5829 - val_accuracy: 0.7611
Epoch 6/10
178/178 [=====] - 73s 408ms/step - loss: 0.5880 - accuracy: 0.7693 - val_loss: 0.5307 - val_accuracy: 0.7769
Epoch 7/10
178/178 [=====] - 80s 448ms/step - loss: 0.5506 - accuracy: 0.7862 - val_loss: 0.5038 - val_accuracy: 0.7816
Epoch 8/10
178/178 [=====] - 95s 535ms/step - loss: 0.4936 - accuracy: 0.8131 - val_loss: 0.4228 - val_accuracy: 0.8022
Epoch 9/10
178/178 [=====] - 104s 582ms/step - loss: 0.4639 - accuracy: 0.8187 - val_loss: 0.4551 - val_accuracy: 0.8133
Epoch 10/10
178/178 [=====] - 76s 426ms/step - loss: 0.4472 - accuracy: 0.8309 - val_loss: 0.4043 - val_accuracy: 0.8307
Total runtime: 733.5690202713013 seconds
    
```

Gambar 8. Proses Epoch

Setiap epoch menampilkan beberapa metrik penting yang digunakan untuk mengevaluasi performa model selama pelatihan. Metrik ini meliputi nilai kerugian (loss) pada data pelatihan dan validasi, serta akurasi (accuracy) pada kedua dataset tersebut. Nilai kerugian digunakan untuk mengukur seberapa baik model memprediksi label data, dengan nilai lebih rendah menunjukkan kinerja yang lebih baik. Akurasi menunjukkan persentase prediksi yang benar dari model, baik pada data pelatihan maupun validasi.

Hasil dari epoch pertama menunjukkan nilai kerugian yang tinggi pada data pelatihan (loss=1.2362) dan akurasi yang rendah (accuracy=0.3884). Hal ini menunjukkan bahwa model sedang memulai proses pelatihan dan belum sepenuhnya memahami pola dalam data. Namun, seiring bertambahnya epoch, nilai kerugian terus menurun, dan akurasi meningkat, baik pada data pelatihan maupun validasi. Ini mengindikasikan bahwa model berhasil belajar dari data dan meningkatkan performanya secara bertahap. Pada epoch terakhir, nilai kerugian pada data pelatihan mencapai 0.4472, dengan akurasi sebesar 0.8309. Akurasi validasi juga meningkat secara signifikan menjadi 0.8307, yang menunjukkan kemampuan model untuk melakukan generalisasi pada data yang belum pernah dilihat sebelumnya. Namun, meskipun akurasi meningkat, tetap terdapat beberapa kesalahan dalam klasifikasi. Salah satu contoh deteksi error adalah pada sampel gambar yang memiliki fitur mirip dengan kelas lain, misalnya dalam klasifikasi objek, model terkadang salah mengidentifikasi objek dengan latar belakang yang kompleks atau kondisi pencahayaan yang buruk. Kesalahan ini dapat terjadi karena model masih belum cukup robust terhadap variasi dalam data.

Analisis lebih lanjut menunjukkan bahwa kesalahan klasifikasi banyak terjadi pada data dengan fitur yang tumpang tindih antar kelas. Berdasarkan teori deep learning, kesalahan ini dapat disebabkan oleh kurangnya jumlah data pelatihan atau keterbatasan dalam arsitektur model yang digunakan. Menurut teori representasi dalam jaringan saraf konvolusi (CNN), fitur yang lebih kompleks dipelajari pada lapisan-lapisan yang lebih dalam. Oleh karena itu, kesalahan ini dapat dikurangi dengan menambahkan jumlah data latih atau meningkatkan kompleksitas model, misalnya dengan menambahkan lapisan konvolusi tambahan atau menggunakan teknik augmentasi data untuk memperkaya variasi dalam dataset. Selain itu, kode menghitung dan mencetak total waktu pelatihan, yang adalah 733 detik untuk seluruh proses 10 epoch. Informasi ini berguna untuk mengukur efisiensi pelatihan model, terutama jika dilakukan dengan dataset besar atau pada perangkat keras dengan sumber daya terbatas. Waktu pelatihan yang relatif singkat menunjukkan bahwa model dapat dilatih secara efisien, namun tetap perlu dianalisis lebih lanjut mengenai trade-off antara waktu pelatihan dan performa model. Secara keseluruhan, hasil pengujian menunjukkan bahwa model berhasil dilatih dengan baik, dengan peningkatan konsisten pada akurasi dan penurunan kerugian pada data pelatihan serta validasi. Namun, analisis kesalahan menunjukkan bahwa model masih memiliki keterbatasan dalam mengenali data dengan fitur yang ambigu. Oleh karena itu, pengembangan lebih lanjut dapat mencakup peningkatan kompleksitas model, augmentasi data, atau penggunaan teknik regularisasi yang lebih baik untuk meningkatkan performa model dalam menghadapi variasi data yang lebih luas.

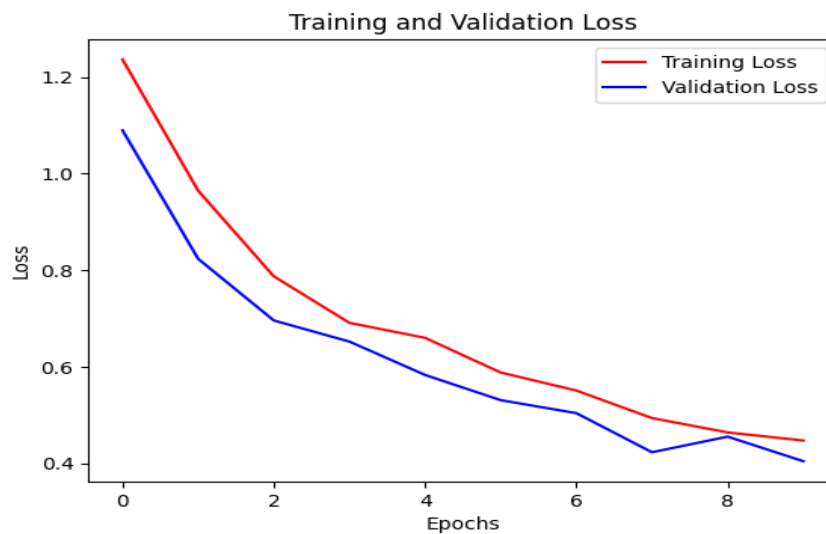


Gambar 9. Grafik Epoch Training Akurasi

Sumbu x merepresentasikan jumlah epoch, yaitu jumlah iterasi penuh pelatihan model terhadap data pelatihan. Sumbu y merepresentasikan akurasi, yaitu persentase prediksi yang benar dari model. Garis merah menggambarkan akurasi pelatihan, sedangkan garis biru menggambarkan akurasi validasi.

Pada awal pelatihan (epoch pertama), terlihat bahwa akurasi pelatihan cukup rendah, yang menunjukkan bahwa model belum mengenali pola dengan baik dalam data. Namun, seiring bertambahnya epoch, akurasi pelatihan meningkat dengan cepat, yang menunjukkan bahwa model mulai belajar dan memahami pola dalam data pelatihan. Akurasi validasi juga meningkat secara signifikan, mencerminkan bahwa model tidak hanya belajar dari data pelatihan tetapi juga mampu menggeneralisasi pada data validasi.

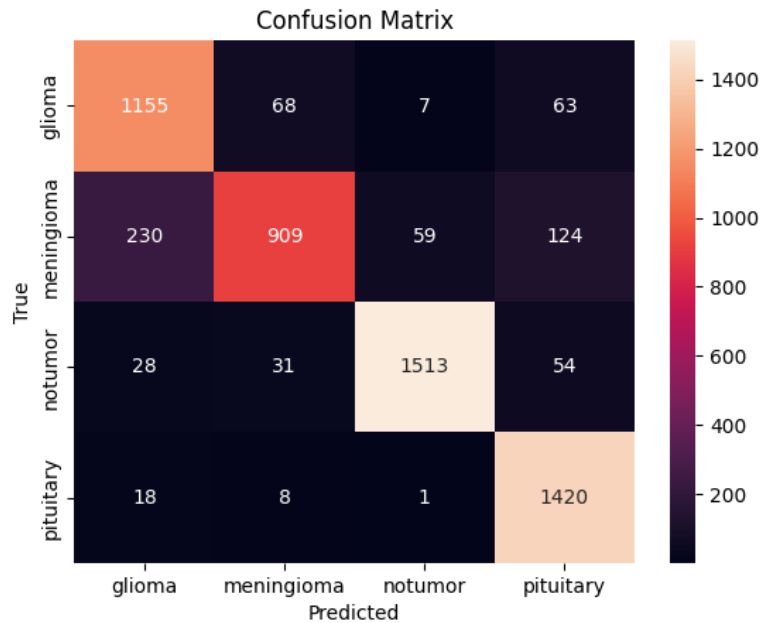
Pada epoch terakhir, baik akurasi pelatihan maupun validasi mencapai nilai yang tinggi, dengan akurasi validasi sedikit lebih tinggi dibandingkan akurasi pelatihan. Ini menunjukkan bahwa model tidak mengalami overfitting, di mana performa pada data pelatihan jauh lebih baik daripada pada data validasi. Model tampaknya mampu belajar secara efektif tanpa kehilangan kemampuan generalisasi. Grafik ini mencerminkan keberhasilan pelatihan model yang stabil, dengan performa yang terus meningkat dan tidak menunjukkan tanda-tanda signifikan overfitting atau underfitting. Ini adalah indikator bahwa model memiliki kapasitas yang baik untuk digunakan pada data pengujian atau implementasi nyata.



Gambar 10. Grafik Epoch Training Loss

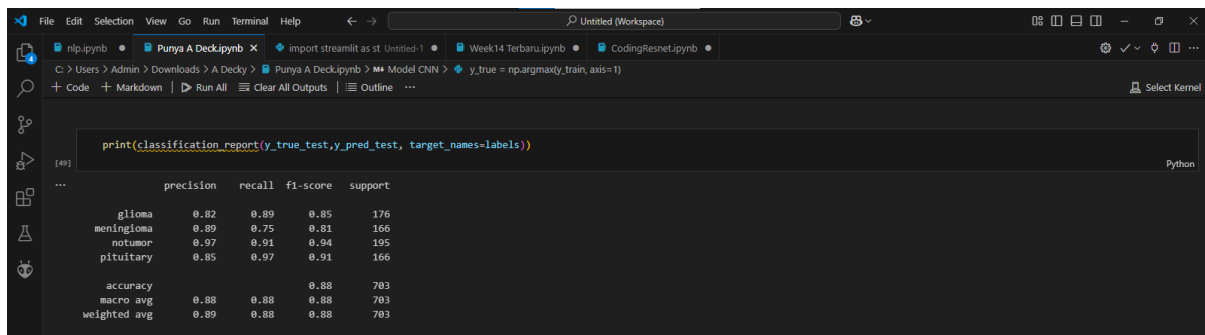
Pada awal pelatihan (epoch pertama), nilai kerugian pada data pelatihan dan validasi tinggi, yang menunjukkan bahwa model belum mengenali pola dalam data dengan baik. Seiring bertambahnya epoch, nilai kerugian pada kedua data secara konsisten menurun. Hal ini menandakan bahwa model semakin memahami pola dalam data pelatihan dan mampu menghasilkan prediksi yang lebih akurat. Perbedaan kecil antara nilai kerugian pada data pelatihan dan validasi menunjukkan bahwa model tidak mengalami overfitting. Jika model overfitting, biasanya kerugian pada data pelatihan akan sangat rendah, sedangkan pada data validasi tetap tinggi. Dalam grafik ini, kerugian pada data validasi bahkan lebih rendah daripada kerugian pada data pelatihan pada epoch tertentu, yang menunjukkan bahwa model bekerja dengan baik dan mampu generalisasi pada data baru.

Pada akhir pelatihan, nilai kerugian pada kedua data mendekati angka 0.4, yang menunjukkan bahwa model telah mencapai performa yang cukup baik. Grafik ini mengindikasikan bahwa pelatihan berjalan stabil, dan model tidak mengalami masalah besar seperti underfitting atau overfitting, sehingga model dapat digunakan untuk prediksi pada data pengujian atau implementasi nyata.



Gambar 11. Confusion Matrix

Diagonal utama matriks (contohnya, 1155 untuk kategori glioma, 909 untuk meningioma, 1513 untuk notumor, dan 1420 untuk pituitary) menunjukkan jumlah sampel yang diklasifikasikan dengan benar, yaitu label prediksi sesuai dengan label sebenarnya. Semakin tinggi nilai pada diagonal utama, semakin baik performa model untuk kategori tersebut.



Gambar 12. Evaluasi Model

Precision mengukur seberapa baik model dalam menghasilkan prediksi yang benar dibandingkan dengan semua prediksi untuk suatu kelas tertentu. Semakin tinggi precision, semakin sedikit prediksi yang salah untuk kelas tersebut. Recall mengukur kemampuan model untuk menangkap semua sampel yang sebenarnya berada dalam suatu kelas. F1-score, yang merupakan rata-rata harmonis dari precision dan recall, memberikan metrik gabungan untuk menilai performa model secara keseluruhan pada setiap kelas. Support menunjukkan jumlah sampel asli dalam setiap kategori kelas, yang memberikan konteks tentang distribusi data.

Dari laporan ini, terlihat bahwa model memiliki performa yang cukup konsisten di semua kategori dengan akurasi total sebesar 88%. Kategori notumor memiliki performa terbaik dengan precision, recall, dan F1-score yang tinggi, masing-masing 0.97, 0.91, dan 0.94. Kategori lainnya, seperti glioma dan pituitary, juga memiliki F1-score yang cukup tinggi, menunjukkan bahwa model mampu mengenali pola pada dataset dengan baik. Kategori meningioma menunjukkan nilai yang sedikit lebih rendah, khususnya pada recall, yang berarti ada beberapa sampel meningioma yang tidak dikenali dengan benar oleh model. Hal ini dapat mengindikasikan perlunya optimasi lebih lanjut pada model atau data pelatihan untuk kelas tersebut.

3. KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa penggunaan algoritma Convolutional Neural Network (CNN) untuk mendeteksi tumor otak berbasis citra MRI dapat menghasilkan model dengan tingkat akurasi yang tinggi dan performa yang konsisten. Teknologi ini mampu secara otomatis menganalisis dan mengklasifikasikan berbagai jenis tumor, seperti glioma, meningioma, pituitary, dan non-tumor, dengan hasil yang menjanjikan. Dengan pendekatan ini, tantangan diagnosis konvensional, seperti waktu proses yang lama, subjektivitas interpretasi, dan keterbatasan sumber daya di beberapa daerah, dapat diminimalkan.

Penelitian ini juga menunjukkan bahwa penggunaan dataset yang terstruktur dengan baik, preprocessing data yang tepat, serta arsitektur CNN yang dioptimalkan memberikan kontribusi signifikan terhadap keberhasilan model. Hasil evaluasi, seperti precision, recall, F1-score, dan analisis melalui confusion matrix, mengonfirmasi kemampuan model dalam membedakan kategori tumor dengan akurasi tinggi, terutama pada kategori tertentu seperti non-tumor.

Keberhasilan model ini tidak hanya memberikan kontribusi pada pengembangan teknologi diagnostik berbasis AI, tetapi juga membuka peluang untuk implementasi lebih luas dalam sistem kesehatan. Penelitian lebih lanjut disarankan untuk mengeksplorasi penggunaan dataset yang lebih besar dan beragam, serta pengintegrasian model ini ke dalam aplikasi praktis untuk mendukung tenaga medis dalam diagnosis tumor otak secara cepat dan akurat.

UCAPAN TERIMAKASIH

Kami mengucapkan terima kasih kepada orang tua tercinta atas dukungan moral dan motivasinya yang tiada henti. Penghargaan juga kami sampaikan kepada Ketua Program Studi Teknik Elektro Universitas Islam Nusantara atas bimbingan dan arahnya yang berharga. Tak lupa, terima kasih kepada teman-teman yang memberikan masukan, semangat, dan dukungan selama penelitian ini. Semua kontribusi ini sangat berarti bagi keberhasilan penelitian kami

REFERENCES

- [1] R. R. Wahid, F. T. Anggraeni, dan B. Nugroho, "Brain Tumor Classification with Hybrid Algorithm Convolutional Neural Network-Extreme Learning Machine," *IJCONSIST J.*, vol. 3, no. 1, hlm. 29–33, Des 2021, doi: 10.33005/ijconsist.v3i1.53.
- [2] Amaliah Faradibah, D. Widyawati, A. U. T. Syahar, dan S. R. Jabir, "Comparison Analysis of Random Forest Classifier, Support Vector Machine, and Artificial Neural Network Performance in Multiclass Brain Tumor Classification," *Indones. J. Data Sci.*, vol. 4, no. 2, hlm. 54–63, Jul 2023, doi: 10.56705/ijodas.v4i2.73.
- [3] D. Husen, "Klasifikasi Citra MRI Tumor Otak Menggunakan Metode Convolutional Neural Network," *Bit-Tech*, vol. 7, no. 1, hlm. 143–152, Agu 2024, doi: 10.32877/bt.v7i1.1576.
- [4] M. H. A. Waskito, A. N. Sihananto, dan A. Junaidi, "Klasifikasi Penyakit Kronis Melalui Mata Menggunakan Algoritma Convolutional Neural Network Dengan Model MobileNet-V3," *Uranus J. Ilm. Tek. Elektro Sains Dan Inform.*, vol. 2, no. 2, hlm. 48–60, Jun 2024, doi: 10.61132/uranus.v2i2.120.
- [5] W. P. Astuti, "Penggunaan Aplikasi Machine Learning (ML) dalam Kurikulum Perubahan Iklim," 2024.
- [6] N. Pratama, M. Liebenlito, dan Y. Irene, "Perbandingan Model Klasifikasi Transfer Learning Convolutional Neural Network Tumor Otak menggunakan Citra Magnetic Resonance Imaging," *J. Sehat Indones. JUSINDO*, vol. 6, no. 01, hlm. 308–318, Jan 2024, doi: 10.59141/jsi.v6i01.81.
- [7] D. Sanjaya dan S. Budi, "Prediksi Pencapaian Target Kerja Menggunakan Metode Deep Learning dan Data Envelopment Analysis," *J. Tek. Inform. Dan Sist. Inf.*, vol. 6, no. 2, Agu 2020, doi: 10.28932/jutisi.v6i2.2678.

- [8] F. C. R., F. Indriyani, dan I. R. Rahadjeng, “Klasifikasi Tumor Otak Berbasis Magnetic Resonance Imaging Menggunakan Algoritma Convolutional Neural Network,” *Digit. Transform. Technol.*, vol. 3, no. 2, hlm. 918–924, Jan 2024, doi: 10.47709/digitech.v3i2.3469.
- [9] R. R. Siregar dan A. M. Husein, “Klasifikasi Tumor Otak pada gambar Magnetic Resonance Images (MRI) dengan Pendekatan Pembelajaran Mendalam.”
- [10] S. N. Fadilah, D. C. R. Novitasari, dan L. Hakim, “Pengaruh Reduksi Fitur Pada Klasifikasi Kanker Paru Menggunakan CNN Dengan Arsitektur GoogLeNet,” *J. Fourier*, vol. 12, no. 1, hlm. 20–32, Apr 2023, doi: 10.14421/fourier.2023.121.20-32.
- [11] R. P. Sari, “Apa itu Convolutional Neural Networks? Pengertian & Cara Kerja.” [Daring]. Tersedia pada: https://aihub.id/pengetahuan-dasar/apa-itu-cnn#google_vignette.
- [12] Seo, “Confusion Matrix: Pengertian, Fungsi, dan Cara Membacanya.” [Daring]. Tersedia pada: <https://course-net.com/blog/confusion-matrix/#:~:text=Confusion%20matrix%20merupakan%20tabel%20yang%20berfungsi%20untuk%20menilai,memberikan%20informasi%20rinci%20mengenai%20klasifikasi%20benar%20dan%20salah>.