

Implementasi Teknologi Deep Learning untuk Diagnostik Stroke Otak Berbasis CNN-LSTM-FNN

Anggita Nur Holifah^{*1}, Imam Tahyudin²

¹Sistem Informasi, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Indonesia

²Magister Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Indonesia

Email: [1anggitanur32@gmail.com](mailto:anggitanur32@gmail.com), [2imam.tahyudin@amikompurwokerto.ac.id](mailto:imam.tahyudin@amikompurwokerto.ac.id)

Abstrak

Stroke otak merupakan salah satu penyebab utama kematian dan kecacatan di dunia, dengan dampak besar pada sistem kesehatan dan ekonomi global. Penelitian ini bertujuan untuk mengembangkan model prediksi dini stroke otak berbasis deep learning dengan mengintegrasikan algoritma Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), dan Feedforward Neural Network (FNN). Dataset yang digunakan terdiri atas citra medis seperti CT scan dan MRI, data temporal, serta informasi klinis lainnya, yang diproses menggunakan teknik preprocessing dan augmentasi data. CNN berfungsi untuk mengekstraksi fitur dari citra medis, LSTM untuk menganalisis data sekuensial, dan FNN untuk mengolah data terstruktur. Hasil penelitian menunjukkan bahwa CNN mencapai akurasi tertinggi sebesar 97%, diikuti oleh LSTM dengan 94%, dan FNN sebesar 70%. Integrasi ketiga algoritma ini menghasilkan model prediksi yang lebih akurat dan komprehensif dibandingkan pendekatan individual. Pendekatan ini berpotensi meningkatkan akurasi diagnosis stroke, mempercepat pengambilan keputusan medis, serta mendukung pengelolaan perawatan pasien yang lebih efisien, sehingga dapat mengurangi beban pada sistem kesehatan global.

Kata kunci: CNN, deep learning, FNN, LSTM, prediksi dini, stroke otak

Smart Cerebral Stroke Diagnostics Based on Intelligence System: Utilizing CNN-LSTM-FNN Algorithms in the Deep Learning Revolution

Abstract

Stroke is one of the leading causes of death and disability worldwide, significantly impacting global healthcare systems and economies. This study aims to develop an early stroke prediction model based on deep learning by integrating Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Feedforward Neural Network (FNN) algorithms. The dataset used includes medical images such as CT scans and MRIs, temporal data, and other clinical information, processed using preprocessing and data augmentation techniques. CNN is utilized for feature extraction from medical images, LSTM for analyzing sequential data, and FNN for processing structured data. The results indicate that CNN achieved the highest accuracy of 97%, followed by LSTM with 94%, and FNN with 70%. The integration of these three algorithms produced a more accurate and comprehensive prediction model compared to individual approaches. This integrative approach has the potential to enhance the accuracy of stroke diagnosis, accelerate medical decision-making, and support more efficient patient care management, thereby reducing the burden on global healthcare systems.

Keywords: CNN, deep learning, early prediction, FNN, LSTM, stroke

1. PENDAHULUAN

Stroke otak merupakan salah satu penyebab utama kematian dan kecacatan di seluruh dunia, dengan dampak besar terhadap kesehatan masyarakat dan sistem ekonomi. Kondisi ini terjadi akibat gangguan aliran darah ke otak, yang menyebabkan kerusakan pada jaringan otak yang bersifat permanen. Secara global, stroke menyebabkan lebih dari 6 juta kematian setiap tahun, menjadikannya sebagai penyebab kematian terbesar kedua setelah penyakit jantung[1]. Menurut data dari Organisasi Kesehatan Dunia (WHO), stroke juga berperan besar dalam kecacatan jangka panjang, yang mencakup gangguan bicara, kesulitan bergerak, dan penurunan kemampuan kognitif pada penderita yang selamat. Situasi ini memberikan tekanan besar pada sistem perawatan kesehatan, baik di negara maju maupun negara berkembang[2].

Seiring perkembangan teknologi, penerapan deep learning dalam bidang medis telah memberikan hasil yang menjanjikan, khususnya dalam menganalisis data citra dan sinyal biologis. Model seperti Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), dan Feedforward Neural Network (FNN) telah digunakan secara luas untuk memproses data medis[3]. CNN unggul dalam ekstraksi fitur dari data citra, LSTM mampu menangani data sekuensial seperti rekam medis berbasis waktu, sementara FNN berguna untuk analisis data terstruktur seperti informasi demografis atau hasil pemeriksaan klinis. Penelitian sebelumnya menunjukkan bahwa model ini dapat digunakan secara terpisah untuk meningkatkan akurasi prediksi pada berbagai kondisi medis, termasuk stroke otak[4]. Pendekatan yang menggabungkan kekuatan dari ketiga algoritma, yaitu CNN, LSTM, dan FNN, untuk memprediksi stroke otak secara dini masih sangat terbatas dalam penelitian yang ada. Sebagai contoh, beberapa studi terdahulu yang mengintegrasikan CNN dengan LSTM telah berhasil meningkatkan akurasi deteksi penyakit lain, seperti kelainan pada citra X-ray, dengan mencapai akurasi hampir 99%. Salah satu penelitian yang relevan[2] yang menggabungkan CNN dan LSTM untuk mendeteksi kelainan pada citra medis dan menunjukkan hasil yang signifikan dalam meningkatkan akurasi deteksi penyakit paru-paru dan kelainan lainnya. Pendekatan serupa dapat diadaptasi untuk prediksi dini stroke otak, yang tidak hanya bergantung pada data citra medis, tetapi juga dapat mengintegrasikan informasi temporal dan struktural[5]. Dengan memanfaatkan data yang lebih beragam ini, potensi untuk mencapai akurasi yang lebih tinggi dalam mendeteksi stroke semakin terbuka lebar, mengingat bahwa stroke melibatkan banyak faktor yang bersifat dinamis dan temporal, seperti perubahan pembuluh darah atau perkembangan gejala yang dapat ditangkap dalam urutan waktu[6].

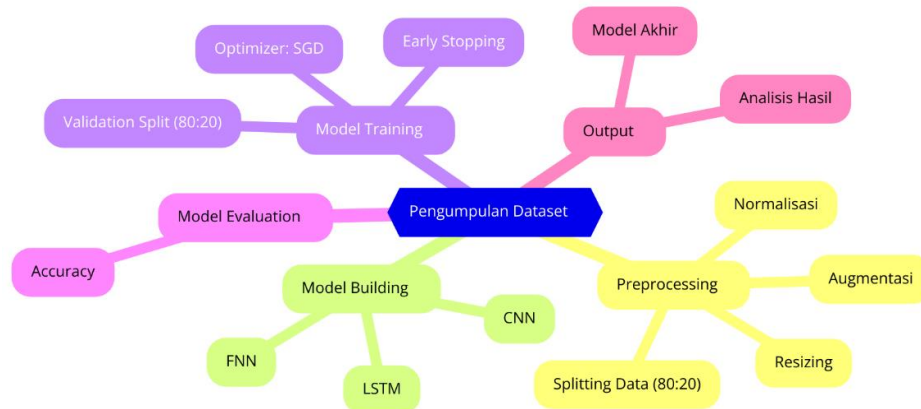
Beberapa studi terdahulu juga telah mengembangkan model serupa untuk meningkatkan prediksi stroke menggunakan berbagai jenis data. Misalnya[7] berhasil mengembangkan model deep learning dengan kombinasi CNN dan LSTM yang memanfaatkan citra MRI dan data klinis untuk memprediksi risiko stroke pada pasien dengan aplikasi Mobile AI Smart Hospital Platform. Penelitian ini membuktikan bahwa menggabungkan data spasial dari gambar medis dengan data temporal yang terkait dengan riwayat medis pasien dapat meningkatkan akurasi prediksi. Penelitian [8] menunjukkan bahwa penggunaan FNN untuk klasifikasi akhir setelah CNN dan LSTM menghasilkan akurasi yang lebih tinggi dalam deteksi dini stroke, dengan menggabungkan citra medis dan informasi klinis yang lebih terstruktur.

Namun, penelitian sebelumnya masih memiliki kekurangan dalam menggabungkan CNN, LSTM, dan FNN secara bersamaan untuk menghasilkan prediksi yang lebih holistik. Sebagian besar model hanya fokus pada kombinasi CNN-LSTM untuk menganalisis data spasial dan temporal, tanpa memanfaatkan potensi FNN dalam mengolah data terstruktur yang lebih kompleks. Seperti pada penelitian [9] yang hanya memanfaatkan model CNN. Penelitian ini bertujuan untuk mengembangkan sebuah model prediksi dini stroke otak yang menggabungkan CNN, LSTM, dan FNN melalui pendekatan deep learning yang inovatif. Model ini tidak hanya akan memanfaatkan data citra medis seperti CT scan atau MRI, tetapi juga memasukkan data temporal dan terstruktur yang dapat memberikan konteks lebih dalam terhadap kondisi pasien, seperti informasi demografis, riwayat penyakit, dan faktor risiko lainnya[10]. Dengan penggabungan berbagai jenis data ini, diharapkan model dapat menghasilkan prediksi yang lebih akurat dan komprehensif. Hal ini, pada gilirannya, akan membantu meningkatkan kecepatan dan ketepatan dalam pengambilan keputusan klinis, memberikan kontribusi signifikan terhadap perawatan pasien stroke yang lebih efektif dan responsif. Pendekatan ini berpotensi memberikan solusi yang lebih efisien dalam deteksi dini stroke, yang dapat mempercepat diagnosis dan meningkatkan hasil pemulihan pasien[11].

2. METODE PENELITIAN

Penelitian ini mengadopsi teknologi deep learning dalam analisis citra medis, khususnya citra CT scan dan MRI, yang dikenal mampu memberikan gambaran yang jelas dan detail tentang kondisi otak. Dalam penelitian ini, kami mengembangkan model deep learning yang dapat mengklasifikasikan berbagai jenis stroke, seperti Ischemic Stroke, Hemorrhagic Stroke, dan Transient Ischemic Attack (TIA). Alur penelitian ini akan dijelaskan dalam beberapa tahapan penting, mulai dari pengumpulan data citra medis, preprocessing data, pembangunan model deep learning, pelatihan model, evaluasi performa, hingga analisis hasil untuk menentukan algoritma mana yang memberikan kinerja terbaik dalam mendeteksi stroke otak[12].

Gambar 1 menggambarkan alur penelitian dalam memprediksi stroke otak menggunakan deep learning. Proses dimulai dengan pengumpulan dataset citra medis, kemudian dilanjutkan dengan preprocessing data, pembangunan dan pelatihan model, hingga evaluasi performa model. Setiap tahap penting untuk memastikan akurasi model dalam mendeteksi stroke.



Gambar 1. Alur Penelitian

2.1. Pengumpulan Data

Langkah pertama dalam penelitian ini adalah mencari dataset medis berupa gambar MRI atau CT scan yang menunjukkan penyakit stroke otak. Dataset ini diambil dari platform Kaggle, yaitu <https://www.kaggle.com/datasets/iashiquil/brain-stroke-prediction-ct-scan-image-dataset>. Dataset ini mencakup berbagai kondisi stroke seperti Ischemic Stroke, Hemorrhagic Stroke, dan Transient Ischemic Attack[13]. Citra medis ini akan digunakan untuk melatih model deep learning dalam mendeteksi jenis stroke yang berbeda. Data ini bisa diambil dari rumah sakit atau repositori citra medis yang sudah tersedia, yang tentunya harus memiliki label yang jelas dan akurat untuk tiap kategori. Dataset yang digunakan terbagi menjadi:

- Data Test: Bagian ini mencakup sebagian besar gambar yang digunakan untuk melatih model, dengan tujuan agar model dapat belajar mendeteksi pola-pola dalam gambar CT scan yang mengindikasikan kondisi stroke.
- Data Validasi: Digunakan selama pelatihan untuk menghindari model mengalami overfitting dan untuk mengoptimalkan parameter model agar mendapatkan kinerja terbaik.
- Data Train: Bagian ini digunakan untuk mengukur kinerja model setelah pelatihan selesai, dengan data yang belum pernah dilihat oleh model sebelumnya, untuk menguji sejauh mana model dapat melakukan prediksi yang akurat.

2.2. Preprocessing Data

Pada tahap preprocessing, beberapa teknik akan diterapkan untuk mempersiapkan citra medis sebelum digunakan dalam pelatihan model. Pertama, gambar akan diubah ukurannya menjadi 224×224 piksel agar memiliki ukuran yang seragam dan sesuai dengan kebutuhan input model[14]. Selanjutnya, dilakukan augmentasi gambar dengan cara rotasi, flipping, zooming, dan perubahan kontras untuk menambah variasi data dan menghindari model dari overfitting. Teknik ini membantu model belajar lebih baik dari beragam variasi citra yang ada. Terakhir, setiap gambar akan dinormalisasi dengan mengubah nilai pikselnya ke rentang $[0, 1]$ atau $[-1, 1]$. Normalisasi ini penting untuk mempercepat proses pelatihan, membuat model lebih mudah memproses citra, dan meningkatkan performa secara keseluruhan[15].

2.3. Pembangunan Model

Pada tahap ini, tiga algoritma deep learning yang berbeda akan digunakan untuk mengembangkan model prediksi stroke, yaitu:

1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan model yang sangat efektif dalam mengidentifikasi pola-pola spasial dalam data gambar, seperti citra CT scan. CNN bekerja dengan cara mengonvolusi gambar input untuk mengekstrak fitur-fitur penting yang terdapat dalam gambar, seperti batasan objek, tekstur, dan pola tertentu yang relevan dengan diagnosis medis, dalam hal ini untuk mendeteksi gejala stroke[16]. Dalam penelitian ini, CNN akan diterapkan dengan menggunakan model yang telah dilatih sebelumnya (transfer learning), seperti VGG16 atau ResNet. Transfer learning memungkinkan kita memanfaatkan model yang telah dilatih pada dataset besar untuk mempelajari representasi citra yang mendalam, sehingga dapat membantu mempercepat proses pelatihan dan meningkatkan akurasi deteksi stroke pada citra CT scan yang lebih spesifik.

$$y(i, j) = (x * w)(i, j) + b \quad (1)$$

2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan jenis jaringan saraf yang termasuk dalam kategori Recurrent Neural Network (RNN), yang sangat baik dalam mengolah data yang memiliki hubungan temporal atau spasial. Meskipun LSTM lebih sering digunakan untuk data urutan seperti teks atau suara, dalam konteks ini LSTM dapat digunakan untuk menangani dimensi tambahan pada citra yang mungkin mengandung informasi spasial yang lebih kompleks atau perubahan yang terjadi secara bertahap dalam gambar. Setelah fitur-fitur penting diekstraksi oleh CNN, LSTM akan mengolah data tersebut untuk menangkap hubungan temporal atau spasial yang lebih dalam, yang bisa memberikan konteks tambahan terkait kondisi stroke, misalnya pola perkembangan gejala atau variasi dalam penggambaran citra[17].

$$h_t = o_t \tanh(C_t) \quad (2)$$

3. Feedforward Neural Network (FNN)

Feedforward Neural Network (FNN) akan digunakan sebagai tahap klasifikasi akhir setelah fitur-fitur dari CNN dan CNN-LSTM diekstraksi. FNN adalah model yang lebih sederhana, terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output. FNN berfungsi untuk memetakan hasil ekstraksi fitur yang telah diperoleh dari CNN dan LSTM ke dalam kategori yang relevan, yaitu stroke atau non-stroke. Meskipun FNN tidak sekompleks CNN atau LSTM dalam hal pengolahan data citra, FNN memiliki kemampuan yang baik dalam memproses informasi yang telah diproses sebelumnya dan mengklasifikasikan hasilnya dengan akurasi yang cukup baik[18]. Dalam sistem ini, FNN berfungsi sebagai penentu akhir dalam proses diagnosis, memastikan bahwa fitur-fitur yang telah dipelajari oleh CNN dan LSTM dapat diterjemahkan dengan baik menjadi kategori yang sesuai untuk pasien.

$$a^{(l)} = f(W^{(l)} \cdot a^{(l-1)} + b^{(l)}) \quad (3)$$

2.4. Pelatihan Model

Pada tahap pelatihan model, data yang sudah diproses akan dibagi menjadi tiga bagian: data train, data validasi, dan data test. Data pelatihan digunakan untuk melatih model, sementara data validasi digunakan untuk memantau kinerja model dan mencegah overfitting. Proses pelatihan melibatkan penggunaan algoritma optimasi seperti Adam atau SGD untuk meminimalkan fungsi kerugian (loss function) dengan cara memperbarui parameter model, seperti bobot dan bias, melalui teknik backpropagation[19]. Model akan dilatih dalam beberapa epoch (iterasi penuh terhadap seluruh dataset) hingga mencapai hasil yang optimal. Selain itu, untuk mencegah model terlalu mempelajari data (overfitting), teknik seperti early stopping dapat diterapkan, yang menghentikan pelatihan jika model tidak menunjukkan peningkatan pada data validasi setelah beberapa iterasi. Pada tahap ini, model dilatih dengan pengaturan berikut:

1. Optimizer

Stochastic Gradient Descent (SGD) digunakan sebagai algoritma optimasi untuk meminimalkan fungsi kerugian (loss function) melalui pembaruan parameter model, seperti bobot dan bias, menggunakan metode backpropagation.

2. Jumlah Epoch dan Early Stopping

Pelatihan dilakukan selama beberapa epoch, namun akan dihentikan lebih awal apabila tidak terjadi peningkatan performa pada data validasi setelah sejumlah iterasi tertentu, untuk mencegah overfitting.

3. Validation Split (80:20)

Dataset dibagi menjadi 80% untuk data pelatihan dan 20% untuk data validasi, guna memantau kinerja model selama proses pelatihan.

2.5. Evaluasi

Setelah model dilatih, langkah selanjutnya adalah mengevaluasi kinerjanya menggunakan data yang belum pernah dilihat sebelumnya, yaitu data validasi dan data uji. Dalam menilai performa model prediktif, penelitian ini menggunakan beberapa metrik utama untuk menilai kemampuan model dalam memprediksi penyakit stroke, seperti presisi, recall, F1 score, dan akurasi. Kinerja model dievaluasi dengan membandingkan proporsi

prediksi yang benar terhadap add up to keseluruhan prediksi. Penghitungan ini dilakukan menggunakan equation yang relevan untuk masing-masing metrik evaluasi[20].

1. Akurasi digunakan untuk Mengukur persentase prediksi yang benar dari total prediksi yang dibuat, baik untuk kelas positif maupun negatif. Akurasi menunjukkan seberapa sering model memberikan hasil yang tepat secara keseluruhan.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

2. Mengukur kemampuan model dalam mendeteksi seluruh kasus positif dengan benar. Semakin tinggi recall, semakin baik model dalam menangkap seluruh kasus positif yang ada.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

3. Mengukur seberapa akurat prediksi positif yang dihasilkan oleh model. Semakin tinggi presisi, semakin sedikit kasus negatif yang salah diklasifikasikan sebagai positif

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

4. Rata-rata harmonis antara presisi dan recall, yang memberikan keseimbangan antara keduanya. F1 score berguna ketika penting untuk mempertimbangkan baik presisi maupun recall, terutama saat terdapat ketidakseimbangan kelas

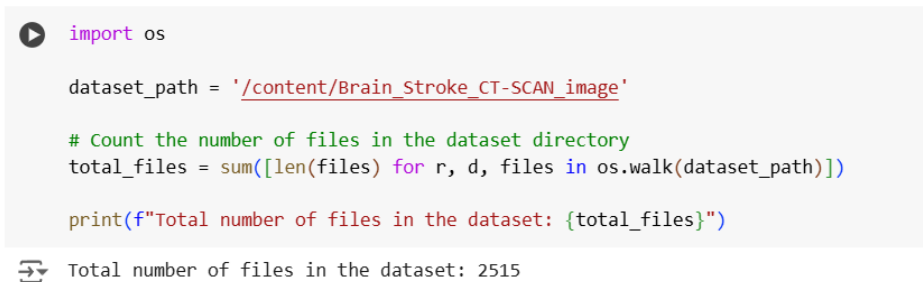
$$F1\ Score = \frac{2 \times precision \times recall}{precision + recall} \quad (7)$$

2.6. Analisis Hasil

Di tahap terakhir, hasil dari berbagai model (CNN, LSTM, dan FNN) akan dibandingkan. Peneliti akan melihat model mana yang menghasilkan kinerja terbaik dalam mendeteksi stroke otak berdasarkan metrik evaluasi yang telah disebutkan. Selain itu, peneliti juga akan menganalisis apakah menggabungkan ketiga model ini (misalnya, CNN-LSTM) dapat memberikan hasil yang lebih baik dibandingkan dengan menggunakan salah satu model saja. Hasil akhir dari penelitian ini adalah pemilihan model yang paling efektif untuk memprediksi stroke otak, yang diharapkan dapat digunakan dalam praktik klinis untuk membantu dokter dalam mendiagnosis stroke secara lebih cepat dan akurat[5].

3. HASIL DAN PEMBAHASAN

3.1. Deskripsi Dataset



```
import os

dataset_path = '/content/Brain_Stroke_CT-SCAN_image'

# Count the number of files in the dataset directory
total_files = sum([len(files) for r, d, files in os.walk(dataset_path)])

print(f"Total number of files in the dataset: {total_files}")
```

Total number of files in the dataset: 2515

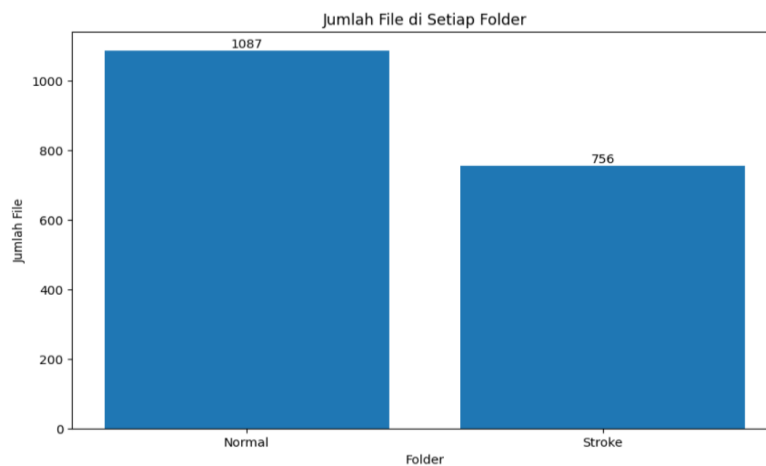
Gambar 2. Jumlah Dataset

Penelitian ini menggunakan dataset yang diambil dari platform Kaggle, dengan judul Brain Stroke Prediction CT Scan Image Dataset. Gambar-gambar ini akan digunakan untuk melatih model deep learning dalam

mengklasifikasikan jenis-jenis stroke, yaitu Ischemic Stroke, Hemorrhagic Stroke, dan Transient Ischemic Attack (TIA). Total gambar yang tersedia dalam dataset ini berjumlah 2515 gambar yang dikelompokkan berdasarkan kategori stroke. Setiap gambar mewakili kondisi otak dari pasien yang terdiagnosis dengan salah satu jenis stroke yang telah disebutkan di atas.

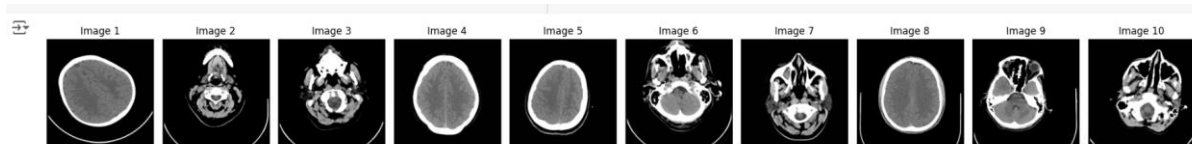
Gambar 2 menunjukkan jumlah dataset secara keseluruhan yang kemudian dataset ini dibagi menjadi tiga bagian utama untuk pelatihan model yaitu train, test, dan validation.

Gambar 3 menunjukkan jumlah file pada folder train yang menunjukkan bahwa jumlah data normal pada folder train berjumlah 1087 sedangkan data stroke berjumlah 756.

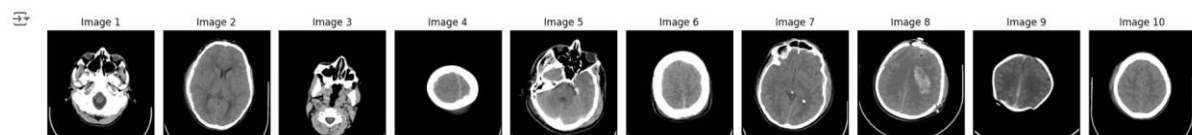


Gambar 3. Jumlah Dataset pada File Train

3.2. Preprocessing Data



Gambar 4. Normal pada Data Test



Gambar 5. Stroke pada Data Test

Pada gambar 4 dan 5 menunjukkan 10 gambar teratas dari data test dengan kelas normal dan stroke, Gambar 4 dan 5 sudah mengalami tiga tahap preprocessing utama. Pertama, resize gambar dilakukan untuk memastikan ukuran gambar konsisten, menjadi 224x224 piksel. Kedua, augmentasi gambar diterapkan dengan 590amper seperti rotasi, flipping, zooming, dan perubahan kontras, untuk menambah variasi data tanpa perlu mengambil gambar baru, yang membantu model lebih fleksibel dan mengurangi overfitting. Terakhir, normalisasi dilakukan untuk mengubah nilai piksel gambar ke rentang 0 hingga 1, mempermudah model dalam memproses gambar secara efisien.

3.3. Pembangunan dan Pelatihan Model

Penelitian [18] menggunakan metode CNN dan LSTM saja hasilnya belum maksimal dengan akurasi CNN hanya 0,76 dan LSTM 0,70, meskipun algoritma deep learning yang didasarkan pada NLP memiliki keunggulan dalam memprediksi hasil klinis, terdapat beberapa kekurangan yang harus diperhatikan. Salah satunya adalah ketergantungan pada kualitas data yang ada; jika laporan radiologi atau data teks lainnya memiliki struktur yang buruk atau kesalahan, model DL dapat menghasilkan prediksi yang tidak akurat. Selain itu, meskipun CNN efektif dalam mengekstraksi fitur spasial dari teks, masih ada tantangan terkait kemampuan model dalam

menangani konteks jangka panjang dan perbedaan gaya penulisan laporan, yang dapat mempengaruhi performa dalam situasi yang lebih kompleks. Pada penelitian kali ini penulis mencoba menggunakan 3 model yaitu CNN, LSTM dan FNN dengan hasil sebagai berikut :

1. Model CNN

```
# Define CNN model
cnn_model = Sequential([
    # Convolutional layers
    Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)), #
    MaxPooling2D((2, 2)), # Downsample
    Conv2D(64, (3, 3), activation='relu'), # More filters for deeper fe
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),

    # Flatten the features
    Flatten(),

    # Fully connected layers
    Dense(128, activation='relu'),
    Dropout(0.5), # Prevent overfitting
    Dense(64, activation='relu'),
    Dense(2, activation='softmax') # Output layer (2 classes: 'Normal' ,
])
```

Gambar 6. Arsitektur Model CNN

Gambar 6 yang ditampilkan menunjukkan input pelatihan dari deep learning pada model algoritma CNN yang digunakan untuk klasifikasi gambar.

- Input Shape (Bentuk Input): (128, 128, 3): Ini mendefinisikan ukuran gambar yang akan diproses oleh model. Artinya, model mengharapkan gambar dengan tinggi 128 piksel, lebar 128 piksel, dan memiliki 3 saluran warna (RGB). Ini adalah format gambar yang umum digunakan.
 - Convolutional Neural Network (CNN): Conv2D(32, (3, 3), activation='relu'): Lapisan ini melakukan konvolusi pada gambar input dengan 32 filter berukuran 3x3. Fungsi aktivasi 'relu' digunakan untuk memperkenalkan non-linearitas pada output.
 - MaxPooling2D(2, 2): Lapisan ini mengurangi dimensi spasial dari fitur peta dengan mengambil nilai maksimum dari setiap jendela 2x2. Ini membantu mengurangi komputasi dan ekstrak fitur yang lebih penting.
- #### 2. Model LSTM

```
# Define the model
lstm_model = Sequential([
    # Feature extraction with CNN
    Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)), # Input s
    MaxPooling2D((2, 2)), # Downsample
    Conv2D(64, (3, 3), activation='relu'), # Another Conv2D layer
    MaxPooling2D((2, 2)),

    # Flatten the spatial features
    Flatten(),

    # Reshape the features for LSTM input
    Dense(256, activation='relu'), # Fully connected layer
    Reshape((8, 32)), # Reshape into (timesteps=8, features=32)

    # Add LSTM for sequential learning
    LSTM(64, return_sequences=False), # LSTM layer with output of size 64

    # Output layer
    Dense(2, activation='softmax') # Adjust the output layer based on your clas
])
```

Gambar 7. Arsitektur Model LSTM

Gambar 7 yang ditampilkan menunjukkan arsitektur model yang digunakan pada model algoritma LSTM, Berikut penjelasan singkat kode:

- (128, 128, 3): Ini mendefinisikan ukuran gambar yang akan diproses oleh model. Artinya, model mengharapkan gambar dengan tinggi 128 piksel, lebar 128 piksel, dan memiliki 3 saluran warna (RGB). Ini adalah format gambar yang umum digunakan.

- b. Convolutional Neural Network (CNN): Conv2D(32, (3, 3), activation='relu'): Lapisan ini melakukan konvolusi pada gambar input dengan 32 filter berukuran 3x3. Fungsi aktivasi 'relu' digunakan untuk memperkenalkan non-linearitas pada output.
- c. MaxPooling2D(2, 2): Lapisan ini mengurangi dimensi spasial dari fitur peta dengan mengambil nilai maksimum dari setiap jendela 2x2. Ini membantu mengurangi komputasi dan ekstrak fitur yang lebih penting.
3. Model FNN

```
# Step 2: Building the FNN Model
def build_fnn(input_shape=(150, 150, 3), num_classes=1):
    model = Sequential()
    model.add(Flatten(input_shape=input_shape)) # Flatten the 2D
    model.add(Dense(512, kernel_regularizer=regularizers.l2(0.01)))
    model.add(LeakyReLU(alpha=0.2)) # Leaky ReLU act
    model.add(BatchNormalization()) # Batch normaliz
    model.add(Dense(256, kernel_regularizer=regularizers.l2(0.01)))
    model.add(LeakyReLU(alpha=0.2)) # Leaky ReLU act
    model.add(BatchNormalization()) # Batch normaliz
    model.add(Dense(num_classes, activation='sigmoid')) # Output la
```

Gambar 8. Arsitektur Model FNN

Gambar 8 menunjukkan arsitektur model algoritma FNN dengan ketentuan input shape: (150, 150, 3) ini menandakan bahwa model ini dirancang untuk memproses gambar dengan ukuran 150 piksel x 150 piksel dengan 3 saluran warna (RGB). Ini adalah ukuran input standar untuk banyak model penglihatan. Parameter num_classes menentukan berapa banyak jenis objek yang ingin kita kenali dengan model ini. Jika kita hanya ingin memprediksi satu nilai angka (misalnya, harga suatu barang), maka nilainya adalah 1. Namun, jika kita ingin membedakan antara beberapa jenis objek (misalnya, kucing, anjing, dan burung), maka nilainya akan lebih dari 1.

3.4. Pembahasan

1. Model CNN

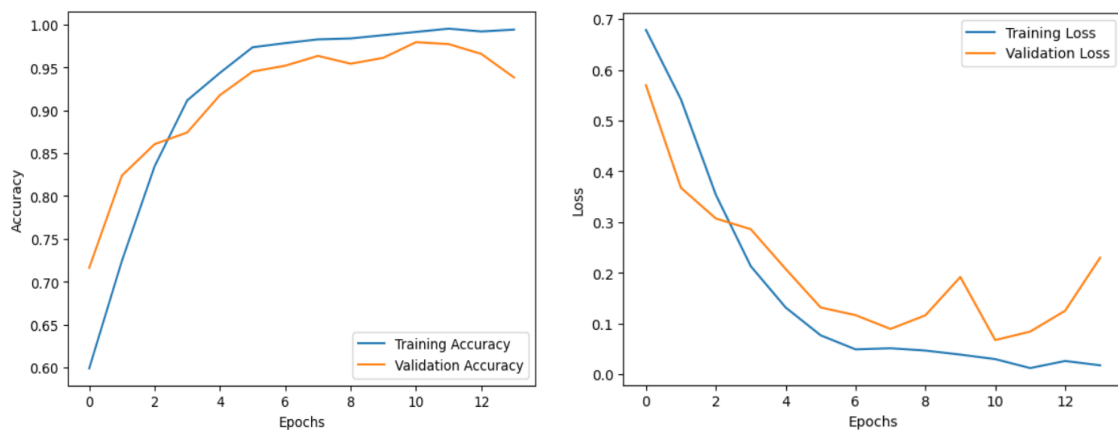
14/14 ————— 6s 435ms/step - accuracy: 0.9830 - loss: 0.0424
Test Loss: 0.06739673018455505
Test Accuracy: 0.9794050455093384

Gambar 9. Test Accuracy Model CNN

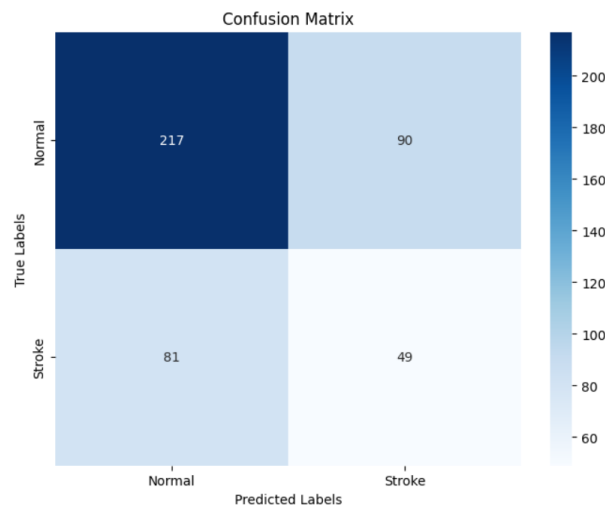
Gambar 9 menampilkan performa yang sangat baik pada iterasi terakhirnya. Proses pelatihan telah menyelesaikan 14 epoch, dengan setiap epoch memerlukan waktu pemrosesan sekitar 6 detik dan rata-rata 435 milidetik per 592amper592. Model ini mencapai 592amper592 akurasi data training, yaitu 98.30%, dengan nilai loss yang sangat minimal sebesar 0.0424. Performa model pada data testing juga menunjukkan hasil yang setara, dengan akurasi mencapai 97.94% dan nilai loss sebesar 0.0674.

Gambar 10 menunjukkan proses pelatihan model machine learning dengan dua grafik. Grafik pertama menggambarkan akurasi yang meningkat pesat seiring bertambahnya epoch, dengan akurasi untuk data pelatihan dan validasi yang 592amper sejajar, menandakan model belajar dengan baik. Grafik kedua menunjukkan penurunan loss yang signifikan di awal, namun setelah epoch ke-6, loss untuk data validasi mulai berfluktuasi dan sedikit meningkat, yang bisa mengindikasikan adanya overfitting. Secara keseluruhan, meskipun model belajar dengan baik, ada tanda-tanda overfitting di akhir pelatihan yang bisa dihindari jika pelatihan dihentikan lebih awal, sekitar epoch ke-8 atau ke-10.

Dari gambar 11 confusion matrix di atas menunjukkan hasil prediksi model dari pasien yang diuji, 217 pasien normal berhasil diprediksi dengan tepat sebagai normal, dan 49 pasien stroke diprediksi dengan tepat sebagai stroke. Namun, model juga melakukan kesalahan, yaitu 90 pasien normal yang diprediksi sebagai stroke (false positive) dan 81 pasien stroke yang diprediksi sebagai normal (false negative). Ini menunjukkan bahwa model masih memiliki kekurangan dalam membedakan kedua kondisi, yang dapat memengaruhi keakuratan diagnosis.



Gambar 10. Visualisasi Accuracy dan Lost Epoch



Gambar 11. Confusion Matrix

2. Model LSTM

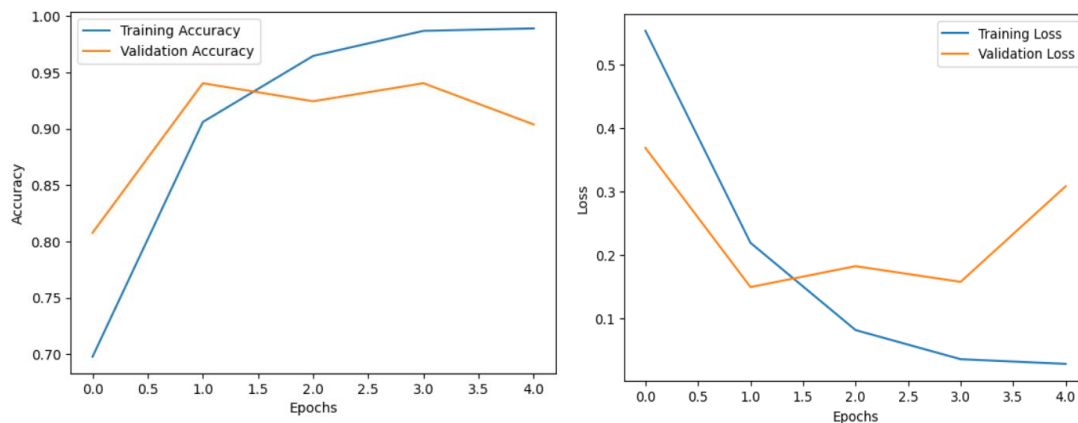
14/14 — 4s 256ms/step - accuracy: 0.9599 - loss: 0.1217
 Test Loss: 0.14935302734375
 Test Accuracy: 0.9405034184455872

Gambar 12. Test Accuraccy Model LSTM

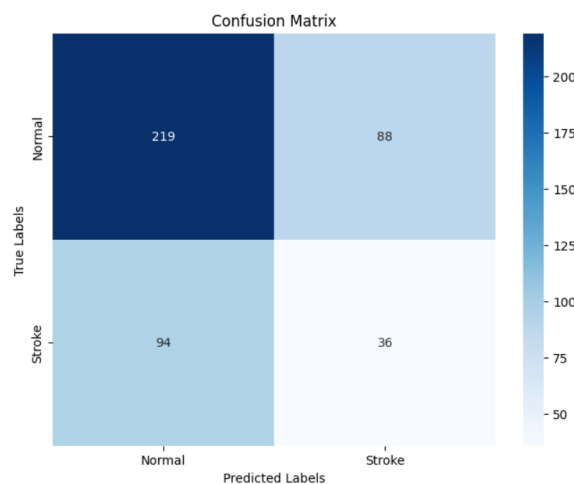
Pada gambar 12, evaluasi model menunjukkan hasil yang sangat baik pada epoch terakhir (14/14), dengan waktu pelatihan hanya 4 detik dan rata-rata 25ms per langkah. Pada data pelatihan, akurasi mencapai 95.99% dengan loss 0.1217, sementara pada data pengujian, akurasi mencapai 94.05% dengan loss 0.1493. Perbedaan akurasi antara data pelatihan dan pengujian yang kecil (sekitar 1.94%) mengindikasikan bahwa model dapat menggeneralisasi dengan baik tanpa mengalami overfitting yang signifikan.

Gambar 13 menunjukkan grafik yang menggambarkan proses pelatihan model machine learning selama 4 epoch. Pada awal pelatihan, akurasi untuk data training dimulai sekitar 70%, sementara akurasi validasi mencapai 80%. Seiring berjalannya waktu, model menunjukkan peningkatan signifikan pada data training hingga hampir mencapai 100%, namun untuk data validasi, akurasi tertinggi tercatat pada epoch pertama sekitar 94%, kemudian menurun secara bertahap hingga 90% pada epoch keempat.

Pada grafik loss, terlihat bahwa loss pada data training terus menurun secara konsisten dari 0.5 hingga mendekati nol. Di sisi lain, loss pada data validasi menurun pada epoch pertama, tetapi kemudian mulai meningkat secara bertahap dan cukup signifikan hingga epoch keempat. Pola ini menunjukkan adanya overfitting setelah epoch pertama, di mana model mulai terlalu fokus pada data training dan kehilangan kemampuan untuk menggeneralisasi pada data validasi. Oleh karena itu, untuk hasil yang lebih optimal, pelatihan model sebaiknya dihentikan setelah epoch pertama untuk mencegah overfitting dan memperoleh model yang lebih seimbang.



Gambar 13. Grafik Accuracy dan Loss Epoch



Gambar 14. Confusion Matrix

Confusion matrix gambar 14 ini menunjukkan hasil prediksi model dalam mengklasifikasikan kondisi normal dan stroke. Model berhasil memprediksi dengan benar 219 kasus normal dan 36 kasus stroke. Namun, terdapat kesalahan prediksi, yaitu 88 kasus normal yang salah diprediksi sebagai stroke (false positive) dan 94 kasus stroke yang salah diprediksi sebagai normal (false negative).

3. Model FNN

Gambar 15 menunjukkan hasil evaluasi bahwa model belum mencapai performa yang optimal pada epoch terakhir (14/14). Model hanya memperoleh akurasi testing sebesar 70.25% dengan loss yang cukup tinggi, yaitu 0.7414. Proses pelatihan memakan waktu 1 detik dengan rata-rata 95ms per langkah. Akurasi yang relatif rendah dan loss yang tinggi menandakan bahwa model belum mampu mengenali pola data dengan baik dan masih memerlukan perbaikan lebih lanjut.

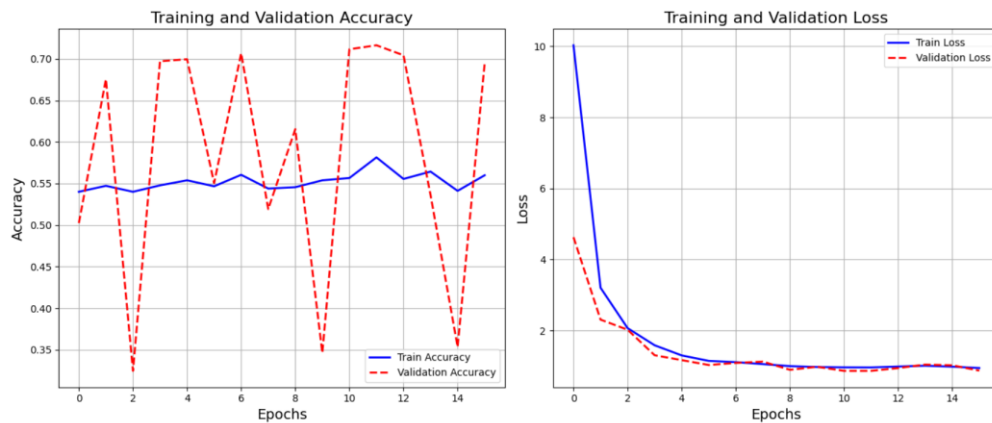
Grafik pada gambar 16 menunjukkan bahwa performa model masih kurang memuaskan. Pada grafik sebelah kiri, akurasi training (garis biru) relatif stabil namun tetap rendah, sekitar 55%, sementara akurasi validasi (garis merah putus-putus) sangat fluktuatif, berkisar antara 35% hingga 70%. Di grafik sebelah kanan, meskipun nilai loss untuk training dan validasi menurun dari angka awal yang tinggi (sekitar 10 untuk training), nilai loss kemudian stagnan di sekitar angka 1 setelah epoch ke-6. Pola ini menunjukkan bahwa model kesulitan dalam proses pembelajaran dan gagal menemukan pola yang tepat dalam data.

Gambar 17 menunjukkan confusion matrix yang menggambarkan hasil prediksi model dalam mengklasifikasikan kondisi normal dan stroke. Dari total prediksi, model berhasil memprediksi 153 kasus normal sebagai normal (True Negative) dan 104 kasus stroke sebagai stroke (True Positive). Namun, terdapat kesalahan prediksi, yaitu 64 kasus normal yang salah diprediksi sebagai stroke (False Positive) dan 47 kasus stroke yang salah diprediksi sebagai normal (False Negative). Meskipun model menunjukkan performa yang cukup seimbang, masih ada ruang untuk perbaikan. Keberhasilan dalam mendeteksi 104 kasus stroke merupakan hal positif, namun 47 kasus stroke yang tidak terdeteksi (false negative) perlu menjadi perhatian utama karena dapat berisiko fatal.

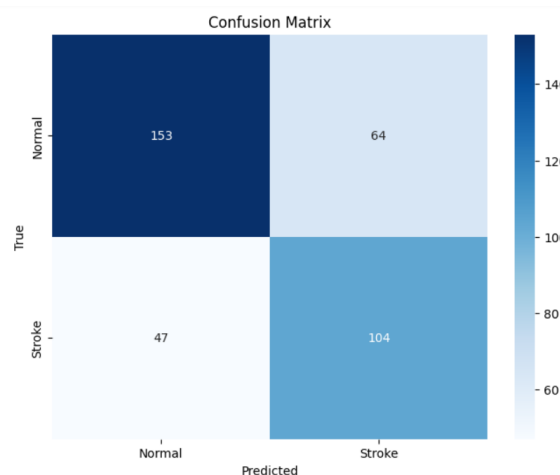
bagi pasien. Selain itu, 64 kasus false positive juga perlu diperbaiki untuk mencegah diagnosis berlebihan yang dapat menyebabkan kecemasan pada pasien.

Test Loss: 0.7414
 Test Accuracy: 0.7025
 14/14 [=====] - 1s 95ms/step

Gambar 15. Test Accuracy Model FNN



Gambar 16. Visualisasi Accuracy dan Los Epoch

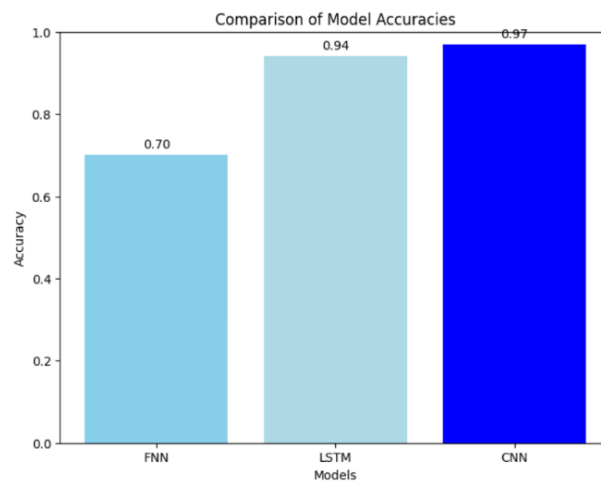


Gambar 17. Confusion Matrix

3.5. Perbandingan Hasil

Berdasarkan grafik pada gambar 18, yang menunjukkan performa terbaik dalam hal prediksi adalah *Convolutional Neural Network* (CNN), yang berhasil meraih tingkat akurasi tertinggi mencapai 97%. Diikuti oleh *Long Short-Term Memory* (LSTM) dengan akurasi 94%, sementara *Fully Connected Neural Network* (FNN) mencatatkan akurasi terendah yaitu 70%. Temuan ini menunjukkan bahwa arsitektur CNN sangat efisien dalam mengolah jenis data yang digunakan dalam penelitian ini, jika dibandingkan dengan kedua model lainnya.

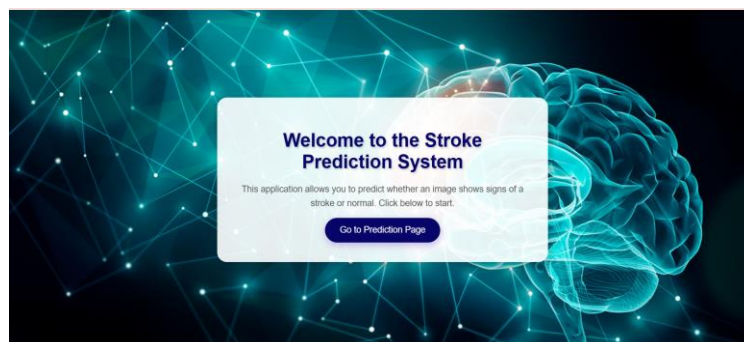
Penerapan model deep learning berbasis NLP dalam telemedicine memungkinkan analisis otomatis terhadap citra medis, seperti MRI atau CT scan, yang dikirimkan oleh pasien dari lokasi terpencil. Dengan algoritma CNN, model ini dapat mengidentifikasi pola-pola abnormal dalam citra dan menghasilkan laporan radiologi yang detail. Ini mempercepat proses diagnosis, memfasilitasi pemantauan kondisi pasien, dan meningkatkan akurasi prediksi hasil klinis, seperti kemungkinan stroke atau tumor. Selain itu, model ini memberikan akses lebih cepat bagi pasien di daerah terpencil untuk mendapatkan layanan medis, serta mendukung pengambilan keputusan klinis yang lebih akurat dan proaktif, dengan integrasi yang lancar dengan sistem telemedicine lainnya.



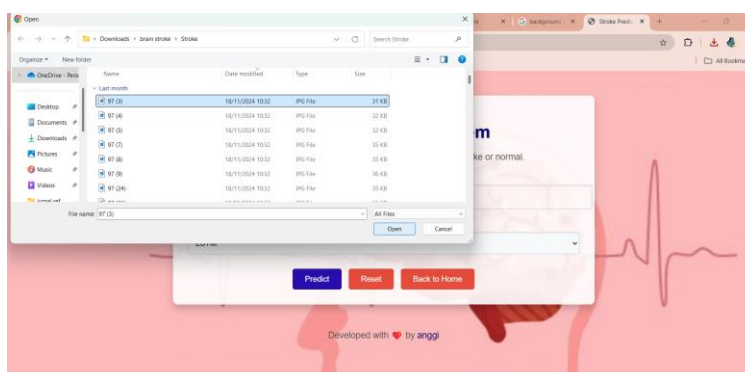
Gambar 18. Perbandingan Model FNN, LSTM, CNN

4. IMPLEMENTASI PREDIKSI STROKE

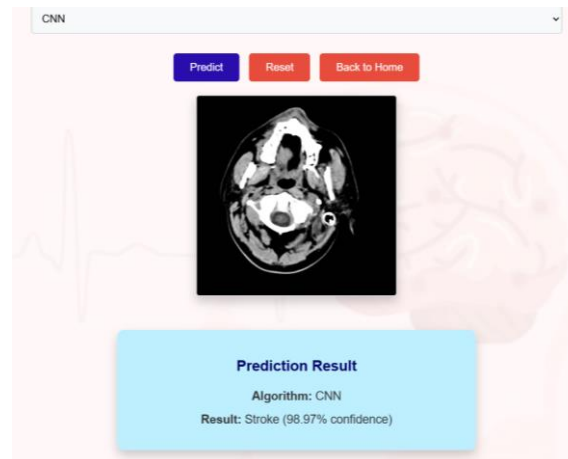
Setelah semua model (CNN, LSTM, dan FNN) dievaluasi, tahap berikutnya adalah mengintegrasikan ketiga model tersebut ke dalam aplikasi berbasis web yang akan digunakan untuk klasifikasi penyakit otak. Pada halaman utama aplikasi ini, pengguna dapat mengunggah gambar ct scan penyakit otak untuk dianalisis oleh model. Sistem ini akan melakukan prediksi berdasarkan model yang sudah dilatih dan memberikan hasil klasifikasi kulit secara langsung di halaman utama.



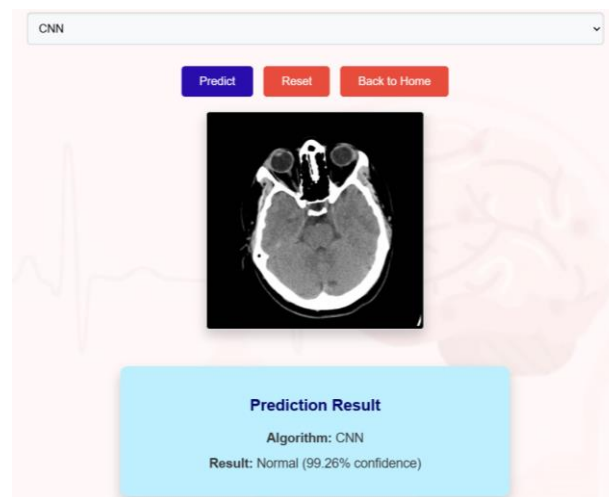
Gambar 19. Dashboard Brain Stroke Prediction



Gambar 20. Insert Gambar Prediction



Gambar 21. Result Stroke



Gambar 22. Result Normal

Gambar 19 menggambarkan halaman utama Dashboard dari sistem, sementara Gambar 20 menunjukkan halaman Klasifikasi, yang memiliki dua form input. Form pertama memungkinkan pengguna memilih model yang ingin digunakan untuk proses klasifikasi, sementara form kedua adalah tempat untuk mengunggah gambar medis (misalnya stroke otak) yang akan diprediksi. Setelah gambar diunggah, pengguna dapat menekan tombol Prediksi untuk memulai proses prediksi dan mendapatkan hasil klasifikasi dari model yang dipilih.

Gambar 21 dan 22 memperlihatkan hasil prediksi dari model setelah diimplementasikan ke dalam sistem berbasis web. Sistem ini berhasil dengan akurat memprediksi perbedaan antara stroke otak dengan normal. Dengan teknologi deep learning menggunakan model CNN, LSTM, dan FNN, yang dilatih untuk mengidentifikasi perbedaan antara kondisi otak yang normal dan yang terdeteksi stroke otak, aplikasi ini memberikan hasil yang dapat diandalkan.

5. KESIMPULAN

Pada penelitian sebelumnya[19] menghasilkan akurasi yang tidak terlalu tinggi dengan hasil, GRU: 0,82 biLSTM: 0,76 LSTM: 0,65 FFNN: 0,59. Pada penelitian ini penggunaan kombinasi model meningkatkan hasil akurasi yang cukup signifikan. Hasil penelitian menunjukkan bahwa model Convolutional Neural Network (CNN) memberikan performa terbaik dengan akurasi mencapai 97%, menandakan efektivitasnya dalam memproses data dan mengekstraksi fitur dari gambar secara otomatis, jauh lebih unggul dibandingkan model lainnya sementara kombinasi CNN-LSTM-FNN meningkatkan efisiensi prediksi dengan hasil model Long Short-Term Memory (LSTM) juga menunjukkan akurasi yang solid sebesar 94%, meskipun sedikit lebih rendah dari CNN, LSTM tetap efektif untuk data dengan urutan atau dependensi waktu, namun masih kalah dalam akurasi dibandingkan CNN. Di sisi lain, model Fully Connected Neural Network (FNN) hanya mencapai akurasi 70%, menunjukkan bahwa

FNN kurang mampu menangani data kompleks ini secara efektif. Studi di masa depan diharapkan dapat mengeksplorasi integrasi data genomik untuk meningkatkan akurasi lebih lanjut.

DAFTAR PUSTAKA

- [1] I Putu Agus Aryawan, I Nyoman Purnama, and Ketut Queena Fredlina, "Analisis Perbandingan Algoritma Cnn Dan Svm Pada Klasifikasi Ekspresi Wajah," *Jurnal Teknologi Informasi dan Komputer*, vol. 9, no. 4, pp. 399–408, 2023, doi: 10.36002/jutik.v9i4.2545.
- [2] R. Sari, A. E. Minarno, and Y. Azhar, "Implementasi Jaringan CNN-LSTM Untuk Deteksi Citra X-Ray Dada Covid-19," *Jurnal Repositor*, vol. 4, no. 4, pp. 451–462, 2024, doi: 10.22219/repositor.v4i4.32290.
- [3] A. Foresta *et al.*, "Heart Beat Prediction Based on Lstm Model on Raspberry Pi," vol. 10, no. 7, pp. 1555–1562, 2023, doi: 10.25126/jtiik.2024118015.
- [4] Y. Yu, B. Parsi, W. Speier, C. Arnold, M. Lou, and F. Scalzo, "LSTM network for prediction of hemorrhagic transformation in acute stroke," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11767 LNCS, pp. 177–185, 2019, doi: 10.1007/978-3-030-32251-9_20.
- [5] B. Akter, A. Rajbongshi, S. Sazzad, R. Shakil, J. Biswas, and U. Sara, "A Machine Learning Approach to Detect the Brain Stroke Disease," *Proceedings - 4th International Conference on Smart Systems and Inventive Technology, ICSSIT 2022*, no. January, pp. 897–901, 2022, doi: 10.1109/ICSSIT53264.2022.9716345.
- [6] Md. M. Islam, S. Akter, Md. Rokunojjaman, J. H. Rony, A. Amin, and S. Kar, "Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique," *International Journal of Electronics and Communications Systems*, vol. 1, no. 2, pp. 57–62, 2021, doi: 10.24042/ijecs.v1i2.10393.
- [7] B. M. Elbagoury, L. Vladareanu, V. Vlădăreanu, A. B. Salem, A. M. Travediu, and M. I. Roushdy, "A Hybrid Stacked CNN and Residual Feedback GMDH-LSTM Deep Learning Model for Stroke Prediction Applied on Mobile AI Smart Hospital Platform," *Sensors*, vol. 23, no. 7, 2023, doi: 10.3390/s23073500.
- [8] W. Hastomo, Sugiyanto, and Sudjiran, "Convolution Neural Network Arsitektur Mobilenet-V2 Untuk Mendeteksi Tumor Otak," *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, vol. 5, no. 1, pp. 17–21, 2021.
- [9] F. N. Fiqri, "Sibyl EEG: Klasifikasi Aktivitas Otak pada Subjek Alkoholik dengan Elektroensefalografi (EEG) menggunakan Deep Learning," 2020.
- [10] S. Rahman, M. Hasan, and A. K. Sarkar, "Prediction of Brain Stroke using Machine Learning Algorithms and Deep Neural Network Techniques," *European Journal of Electrical Engineering and Computer Science*, vol. 7, no. 1, pp. 23–30, 2023, doi: 10.24018/ejece.2023.7.1.483.
- [11] V. Bandi, D. Bhattacharyya, and D. Midhunchakkravarthy, "Prediction of brain stroke severity using machine learning," *Revue d'Intelligence Artificielle*, vol. 34, no. 6, pp. 753–761, 2020, doi: 10.18280/RIA.340609.
- [12] D. R. Chandranegara, Z. Sari, M. B. Dewantoro, H. Wibowo, and W. Suharso, "Implementation of Generative Adversarial Network (GAN) Method for Pneumonia Dataset Augmentation," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 4, no. May, 2023, doi: 10.22219/kinetik.v8i2.1675.
- [13] S. A. Mostafa, D. S. Elzanfaly, and A. E. Yakoub, "A Machine Learning Ensemble Classifier for Prediction of Brain Strokes," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, pp. 258–266, 2022, doi: 10.14569/IJACSA.2022.0131232.
- [14] M. S. Sirsat, E. Fermé, and J. Câmara, "Machine Learning for Brain Stroke: A Review," *Journal of Stroke and Cerebrovascular Diseases*, vol. 29, no. 10, 2020, doi: 10.1016/j.jstrokecerebrovasdis.2020.105162.
- [15] D. Rika Widianita, "No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する分散構造分析Title," *AT-TAWASSUTH: Jurnal Ekonomi Islam*, vol. VIII, no. I, pp. 1–19, 2023.
- [16] Y. A. Choi *et al.*, "Deep learning-based stroke disease prediction system using real-time bio signals," *Sensors*, vol. 21, no. 13, 2021, doi: 10.3390/s21134269.
- [17] A. A. SHELEMO, "No Titleبليب," *Nucl. Phys.*, vol. 13, no. 1, pp. 104–116, 2023.
- [18] T. S. Heo *et al.*, "Prediction of stroke outcome using natural language processing-based machine learning of radiology report of brain MRI," *J Pers Med*, vol. 10, no. 4, p. 286, 2020.

- [19] M. Kaur, S. R. Sakhare, K. Wanjale, and F. Akter, “[Retracted] Early Stroke Prediction Methods for Prevention of Strokes,” *Behavioural Neurology*, vol. 2022, no. 1, p. 7725597, 2022.