

KLASIFIKASI CITRA SIMBOL MATEMATIKA MENGGUNAKAN METODE *CONVOLUTION NEURAL NETWORK* (CNN)

Angga Warjaya, Alfina Richi, Hermawan Syahputra, Fanny Ramadhani

Ilmu Komputer, Universitas Negeri Medan

Jl. William Iskandar Ps. V, Kenangan Baru, Kecamatan Percut Sei Tuan,

Kabupaten Deli Serdang, Sumatera Utara, Indonesia 20221

anggawarjaya.4213250020@mhs.unimed.ac.id

ABSTRAK

Pemanfaatan notasi matematika sangat penting dalam menjelaskan konsep dan kerangka matematika, berfungsi sebagai alat dasar untuk komunikasi dan pemecahan masalah. Namun, gaya tulisan tangan yang beragam yang digunakan oleh individu menghadirkan tantangan unik dalam mengidentifikasi simbol matematika secara akurat karena perbedaan evolusi tulisan tangan dari waktu ke waktu. Munculnya teknologi pengenalan pola, terutama dalam pengenalan pola tulisan tangan, menekankan pentingnya mengembangkan aplikasi untuk mendeteksi dan menafsirkan simbol matematika tulisan tangan, dengan fokus pada penggunaan *Convolution Neural Network* (CNN) untuk kategorisasi otomatis, menampilkan tingkat akurasi yang menjanjikan. Studi ini menunjukkan kemampuan luar biasa model CNN untuk mengkategorikan simbol matematika dengan tingkat akurasi tinggi 99,25%, menunjukkan potensi signifikan metodologi CNN dalam mengklasifikasikan pola simbol matematika secara efektif dan perlunya eksplorasi lebih lanjut untuk meningkatkan kinerja klasifikasi dalam domain ini.

Kata kunci : Klasifikasi, CNN, Simbol Matematika

1. PENDAHULUAN

Dalam bidang matematika, pemanfaatan notasi matematika memainkan peran penting dalam penggambaran ide atau struktur matematika. Memanfaatkan simbol matematika membantu dalam perumusan dan penyelesaian tantangan matematika. Penggunaan simbol-simbol ini, kejadian umum, terkadang melibatkan simbol-simbol tertentu yang tampaknya asing. Simbol matematika memiliki kapasitas untuk menunjukkan beragam entitas termasuk penambahan, pengurangan, pembagian, theta, alfa, lambda, dan berbagai karakter lainnya.

Tantangan mengidentifikasi simbol matematika terletak pada karakteristik yang berbeda dari gaya tulisan tangan. Variasi dalam gaya penulisan menimbulkan hambatan yang signifikan dalam hal pengenalan simbol dalam bidang matematika. Pada dasarnya, tulisan tangan manusia menunjukkan bentuk yang tidak dapat dibedakan namun tidak identik. Ini menyiratkan bahwa tulisan tangan manusia cenderung berkembang terus menerus, mengingat bahwa gaya penulisan masing-masing individu bervariasi berdasarkan faktor-faktor seperti kemiringan tulisan seseorang, gaya goresan, dan tingkat rekursivitas yang terlibat [1].

Dengan kemajuan waktu, telah terjadi kemajuan paralel dalam sains dan teknologi, yang mengarah pada pengembangan teknologi pengenalan pola. Salah satu bidang khusus dalam bidang ini adalah pengenalan pola penulisan. Di era kontemporer, kemampuan mengenali pola tulisan telah menjadi persyaratan penting bagi manusia. Kebutuhan ini muncul dari inefisiensi yang terkait dengan tulisan tangan manual, yang seringkali mengharuskan pengetikan konten tertulis berikutnya ke komputer.

Akibatnya, skenario ini telah mendorong pembuatan aplikasi smartphone yang dirancang untuk mendeteksi dan menafsirkan tulisan tangan. Pengenalan pola tulisan tangan menghadirkan tantangan yang lebih kompleks dibandingkan dengan pengetikan, terutama karena beragam variasi yang dapat diamati dalam gaya tulisan tangan [2].

Penelitian ini akan memeriksa kategorisasi otomatis pola simbol matematika melalui pemanfaatan jaringan saraf. Pendekatan yang akan digunakan dalam penyelidikan ini adalah *Convolutional Neural Network* (CNN), jenis model tertentu yang dikenal karena kemampuannya dalam menangani data gambar. Sejumlah upaya penelitian telah menunjukkan tingkat akurasi yang menjanjikan dalam mengenali pola dalam gambar. Misalnya, studi Alvin Kusuma membandingkan kinerja dua arsitektur CNN, ResNet34 dan DenseNet121, dengan yang terakhir menunjukkan akurasi yang unggul [3].

Selanjutnya, Martina dkk. melakukan penelitian yang melibatkan kategorisasi menggunakan CNN yang melibatkan formula yang ditampilkan dan sebaris, mencapai skor F1 masing-masing 0,98% dan 0,91% [4].

Selain itu, Desti melakukan analisis komparatif arsitektur CNN, mengungkapkan bahwa LaNet5 mengungguli Mesin Pembelajaran Ekstrem dengan tingkat akurasi 98,04% [5].

Studi-studi ini secara kolektif menggarisbawahi potensi CNN dalam mengklasifikasikan pola simbol matematika secara efektif, sehingga menyoroti signifikansinya dalam bidang tugas pengenalan gambar. Selanjutnya, mereka menekankan pentingnya mengeksplorasi berbagai arsitektur dan metodologi CNN untuk meningkatkan akurasi klasifikasi dan

kinerja dalam domain ini. Temuan dari upaya penelitian ini berfungsi sebagai wawasan berharga untuk kemajuan lebih lanjut dalam klasifikasi otomatis pola simbol matematika menggunakan jaringan saraf.

2. TINJAUAN PUSTAKA

2.1. Definisi Simbol Matematika

Simbol dalam ranah matematika mencakup definisi yang luas, termasuk huruf, angka, atau tanda yang mewakili nilai numerik, operasi, atau bahkan hasil penalaran matematika. Ini menyoroti sifat matematika sebagai bahasa simbolis, di mana semua cara kerja rumit dari pikiran matematika diartikulasikan melalui representasi simbolis. Dalam domain matematika murni, simbol-simbol ini mungkin tampak sederhana bagi mata yang tidak terlatih, tanpa konotasi dunia nyata langsung; namun, mereka memperoleh signifikansi dari prinsip-prinsip deduktif yang mendasari konsep matematika atau dari makna kontekstual yang lebih luas [6].

Akibatnya, simbol matematika berfungsi sebagai penggambaran visual dari konstruksi matematika yang rumit, membantu dalam komunikasi dan pemahaman ide-ide dan hubungan yang kompleks. Beragam simbol matematika yang biasa ditemui dalam kehidupan sehari-hari, seperti yang diilustrasikan pada Gambar 1, menggarisbawahi pentingnya representasi simbolik ini dalam memfasilitasi wacana matematika dan proses pemecahan masalah.

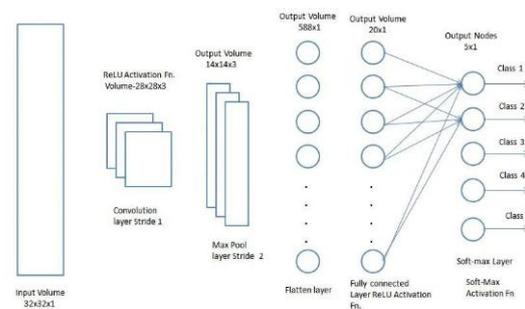
2.2. Tulisan Tangan

Pengenalan tulisan tangan adalah tugas yang mudah diselesaikan oleh manusia, namun menimbulkan tantangan besar bagi mesin atau komputer. Kesulitan muncul dari ketidakkonsistenan yang melekat dalam tulisan tangan manusia, dengan masing-masing individu menunjukkan gaya penulisan yang unik. Pengenalan tulisan tangan, juga dikenal sebagai pengenalan tulisan tangan, mengacu pada kemampuan sistem komputer untuk menafsirkan dan memecahkan kode input tulisan tangan yang diperoleh dari berbagai sumber seperti kertas, foto, atau dokumen, yang kemudian dipindai menggunakan pemindai optik, biasa disebut sebagai pengenalan karakter optik. Proses pengenalan tulisan tangan memerlukan analisis pola untuk setiap karakter individu, diikuti dengan pengenalan simbol atau karakter baru melalui perbandingan piksel demi piksel dengan pola yang disimpan. Proses rumit ini menyoroti kompleksitas yang terlibat dalam mengajar mesin untuk menguraikan dan menafsirkan tulisan tangan manusia secara akurat. Selain itu, tantangan dalam pengenalan tulisan tangan menggarisbawahi upaya berkelanjutan dalam penelitian dan pengembangan untuk meningkatkan akurasi dan efisiensi teknologi ini, terutama dalam aplikasi yang membutuhkan interpretasi teks tulisan tangan yang tepat dan andal. Sebagai kesimpulan, seluk-beluk pengenalan tulisan tangan memerlukan pemahaman menyeluruh tentang pola penulisan manusia dan

proses teknologi yang terlibat dalam menerjemahkan pola-pola ini ke dalam bentuk digital [7].

2.3. Convolution Neural Network (CNN)

Convolutional Neural Network, umumnya dikenal sebagai CNN, adalah teknik pembelajaran mesin yang berada di bawah payung model *Multi-Layer Perceptron* (MLP), yang secara khusus dirancang untuk pemrosesan struktur data dua dimensi. Pendekatan khusus ini dikategorikan sebagai Deep Neural Network karena konfigurasi tingkat jaringan dan aplikasi ekstensif di bidang analisis data gambar. CNN menggunakan dua teknik mendasar dalam operasinya: yang pertama adalah proses klasifikasi yang menggunakan mekanisme feedforward, dan yang kedua melibatkan fase pembelajaran di mana backpropagasi digunakan untuk menyempurnakan parameter jaringan. Metodologi ini secara kolektif berkontribusi pada efektivitas dan keserbagunaan CNN dalam berbagai aplikasi, menjadikannya alat penting dalam domain pembelajaran mesin dan kecerdasan buatan [8].



Gambar 1. Arsitektur CNN

Dalam ilustrasi yang diberikan pada Gambar 2, dapat diamati bahwa *Convolutional Neural Network* (CNN) terdiri dari berbagai lapisan, yang meliputi lapisan input, lapisan output, dan serangkaian lapisan tersembunyi yang diposisikan di antaranya. Lapisan tersembunyi ini mencakup lapisan konvolusi, lapisan penggabungan, serta lapisan yang terhubung sepenuhnya. Intinya, CNN mahir mengidentifikasi elemen-elemen dasar seperti tepi, garis, tekstur, dan pola tidak rumit lainnya yang ada dalam gambar [9].

Melalui konsolidasi operasi konvolusi dalam lapisan konvolusi, beragam fitur spesifik dapat dianalisis dengan cermat dalam lapisan yang sama. Penggunaan beberapa lapisan mendorong pengembangan pengaturan hierarkis di mana lapisan awal memperoleh pengetahuan tentang fitur dasar seperti garis atau sudut, dan kemudian memberikan informasi ini ke lapisan berikutnya untuk mengidentifikasi fitur yang lebih rumit seperti angka atau rambu lalu lintas. Perkembangan ini menggarisbawahi kapasitas jaringan untuk secara progresif memahami dan membedakan atribut yang semakin kompleks yang tertanam dalam data yang diprosesnya. Selain itu, struktur hierarkis yang ditimbulkan oleh beberapa lapisan memfasilitasi

kemampuan jaringan untuk menyaring sejumlah besar informasi visual dan mengekstrapolasi abstraksi tingkat tinggi dari data input. Pembagian tugas di antara lapisan yang berbeda dalam CNN mengarah pada sistem canggih yang mampu secara progresif menyempurnakan pemahamannya tentang fitur visual dengan cara yang mencerminkan proses kognitif yang sering dikaitkan dengan penglihatan manusia. Akibatnya, interaksi rumit dari berbagai lapisan dalam CNN memungkinkannya untuk membedah dan menafsirkan data visual dengan tingkat presisi dan efisiensi yang luar biasa [10].

3. METODE PENELITIAN

Dalam penelitian yang dilakukan, algoritma *Convolutional Neural Network* (CNN) digunakan untuk tujuan mengategorikan gambar yang menggambarkan tulisan simbol matematika. CNN menonjol sebagai teknik yang sangat disukai dan efisien dalam ranah Deep Learning yang dirancang khusus untuk klasifikasi data gambar, menunjukkan efektivitas yang luar biasa. Pemanfaatan perangkat lunak Python 3.10.12 disertai dengan paket Keras sangat merampingkan seluk-beluk yang terlibat dalam pengembangan dan pelaksanaan algoritma CNN, sehingga meningkatkan efisiensi keseluruhan proses.

Dengan menerapkan algoritma CNN, serangkaian operasi konvolusi dijalankan pada data untuk mengidentifikasi dan menetapkan pola khas yang penting untuk klasifikasi akurat gambar tulisan simbol matematika. Mengingat sifatnya sebagai sistem pembelajaran yang mendalam, CNN dicirikan oleh banyak lapisan, masing-masing melayani fungsi tertentu dalam jaringan. Studi ini menempatkan penekanan yang signifikan pada perumusan arsitektur CNN yang kuat untuk memastikan produksi hasil klasifikasi gambar yang tepat di berbagai kategori. Arsitektur CNN terdiri dari beragam lapisan, meliputi lapisan *convolution and activation*, lapisan *pooling*, lapisan *flatten*, lapisan *fully connected*, dan lapisan *output layers* yang penting untuk menyelesaikan proses klasifikasi.

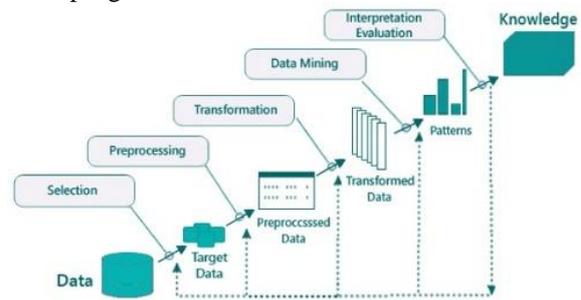
3.1. Sumber Data

Data sekunder, yang mengacu pada data yang diperoleh secara tidak langsung atau dari sumber yang sudah ada sebelumnya, digunakan dalam studi khusus ini. Gambar tulisan symbol matematika yang dimasukkan dalam penelitian ini bersumber dari laman kaggle, dengan waktu akses pada 14 Juni 2024. Link akses: <https://www.kaggle.com/datasets/xainano/hand-written-math-symbols>

3.2. Teknik Analisis Data

Data yang digunakan sebagai referensi untuk penelitian diverifikasi selama prosedur analisis penelitian. Setelah akuisisi data sekunder, analisis data dilakukan dengan menggunakan metodologi Knowledge Discovery in Database (KDD) merupakan metode untuk mengekstraksi pengetahuan dari

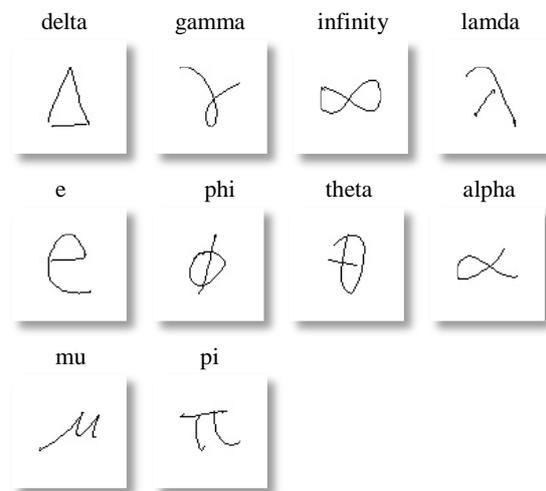
database tertentu yang terdiri dari tabel yang saling berhubungan dan saling terkait. Pengetahuan yang diperoleh dari prosedur ini dapat berfungsi sebagai dasar pengetahuan.



Gambar 2. Metode Knowledge Discovery In Database (KDD)

3.3. Data Selection

Pengumpulan data citra melalui website kaggle yang dapat memperoleh banyak data citra. Proses pengumpulan dan seleksi data yang dilakukan harus relevan terhadap analisis yang dilakukan. Dalam penelitian ini Jumlah data yang digunakan dalam penelitian ini yaitu sebanyak 1000 data gambar yang terdiri dari 10 jenis gambar tulisan yang dimana kumpulan data ini akan membentuk 10 kelas yang masing masing kelasnya berisi 100 data gambar.



Gambar 3. Sampel Dataset

3.4. Data Preprocessing

Pada tahap ini dilakukannya pra proses data dimana, data yang sudah dikumpulkan selanjutnya akan dibagi menjadi dua bagian yaitu data pelatihan sebesar 60 % dan data validasi diberikan proporsi sebesar 40 %. Kemudian mengatur parameter data gambar menggunakan generator dan keras menjadi 224 x 224 piksel. Selanjutnya melakukan normalisasi rescale, agar rentang nilai piksel menjadi lebih spesifik berkisar antara 0 dan 1, menentukan batch size yang sesuai, mengidentifikasi kelas-kelas yang akan digunakan untuk membangun model. Dan langkah terakhir dalam preprocessing data ialah menampilkan sampel gambar yang digunakan dalam proses

pengujian model dan validasi untuk memastikan model telah belajar dengan baik.

4. HASIL DAN PEMBAHASAN

Verifikasi kinerja sistem merupakan langkah penting yang tidak dapat diabaikan, oleh karena itu perlunya proses pengujian. Dalam proses pengujian ini, pelatihan model *Convolutional Neural Network* (CNN) dilakukan dengan cermat untuk memastikan akurasi dan keandalan dalam evaluasi kinerja sistem. Prosedur pelatihan model CNN diuraikan dengan cermat dan dijalankan secara sistematis untuk mengoptimalkan kemampuan sistem dan memvalidasi efisiensinya selama fase pengujian.

Model: "sequential_2"

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 7, 7, 2048)	20861480
conv2d_10 (Conv2D)	(None, 7, 7, 64)	1179712
max_pooling2d_2 (MaxPoolin g2D)	(None, 3, 3, 64)	0
dropout_2 (Dropout)	(None, 3, 3, 64)	0
flatten_2 (Flatten)	(None, 576)	0
dense_2 (Dense)	(None, 10)	5770

Total params: 22046962 (84.10 MB)
 Trainable params: 1185482 (4.52 MB)
 Non-trainable params: 20861480 (79.58 MB)

Gambar 3. Model CNN

Arsitektur model terdiri dari beberapa lapisan, dimulai dengan lapisan Xception, dikenali sebagai lapisan fungsional yang menunjukkan bentuk keluaran yang dilambangkan sebagai (None, 7, 7, 2048) dan mencakup jumlah parameter substansial 20.861.480. Mengikuti lapisan Xception, ada lapisan Conv2D yang menghasilkan bentuk keluaran yang ditentukan sebagai (None, 7, 7, 64) dengan penghitungan parameter 1.179.712. Jumlah parameter ini berasal dari penerapan kernel 3x3 yang menampilkan 2048 saluran input dan 64 saluran keluaran

Selanjutnya, lapisan MaxPooling2D diperkenalkan ke dalam arsitektur, ditandai dengan tidak ada parameter tambahan dan menghasilkan bentuk output yang dilambangkan sebagai (None, 3, 3, 64). Selain itu, lapisan berikutnya dalam urutan adalah lapisan Dropout, yang juga tidak memperkenalkan parameter baru sambil mempertahankan bentuk keluaran yang identik dengan lapisan sebelumnya. Pengaturan lapisan ini menampilkan perkembangan sistematis dalam desain model, yang berpuncak pada representasi efisien dari pola data yang mendasarinya.

Setelah selesainya lapisan sebelumnya, lapisan Flatten bertanggung jawab untuk mengubah output menjadi bentuk vektor, khususnya (None, 576), yang berasal dari produk 3 * 3 * 64. Setelah ini, lapisan terakhir dalam model adalah lapisan Padat dengan bentuk keluaran (None, 10), berisi total 5.770 parameter, dihitung sebagai (576 * 10) + 10. Secara total, seluruh model terdiri dari 22.046.962 parameter, dengan 1.185.482 parameter yang ditetapkan sebagai

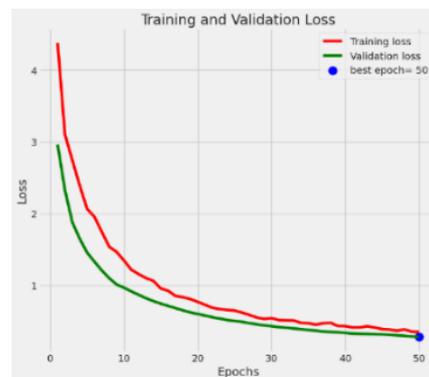
dapat dilatih, sedangkan 20.861.480 parameter sisanya dikategorikan sebagai tidak dapat dilatih. Alokasi parameter ini adalah hasil dari perhitungan dalam setiap lapisan arsitektur *Convolutional Neural Network* (CNN) dan tidak dipengaruhi oleh jumlah kategori gambar berbeda yang digunakan untuk tujuan klasifikasi. Tujuan utama dari model ini berpusat di sekitar pelaksanaan tugas klasifikasi, di mana 10 neuron keluaran secara khusus disesuaikan agar sesuai dengan jumlah kelas yang diantisipasi untuk prediksi dan kategorisasi yang akurat.

```

Epoch 40/50 [-----] - 6s 1s/step - loss: 0.4286 - accuracy: 0.9833 - val_loss: 0.3341 - val_accuracy: 1.0000
Epoch 41/50 [-----] - 7s 1s/step - loss: 0.4119 - accuracy: 0.9800 - val_loss: 0.3262 - val_accuracy: 1.0000
Epoch 42/50 [-----] - 6s 1s/step - loss: 0.4120 - accuracy: 0.9800 - val_loss: 0.3222 - val_accuracy: 1.0000
Epoch 43/50 [-----] - 7s 2s/step - loss: 0.4273 - accuracy: 0.9667 - val_loss: 0.3282 - val_accuracy: 1.0000
Epoch 44/50 [-----] - 6s 1s/step - loss: 0.4099 - accuracy: 0.9733 - val_loss: 0.3173 - val_accuracy: 1.0000
Epoch 45/50 [-----] - 6s 1s/step - loss: 0.3883 - accuracy: 0.9800 - val_loss: 0.3136 - val_accuracy: 1.0000
Epoch 46/50 [-----] - 7s 1s/step - loss: 0.3824 - accuracy: 0.9817 - val_loss: 0.3072 - val_accuracy: 1.0000
Epoch 47/50 [-----] - 6s 1s/step - loss: 0.3644 - accuracy: 0.9800 - val_loss: 0.3084 - val_accuracy: 1.0000
Epoch 48/50 [-----] - 7s 1s/step - loss: 0.3836 - accuracy: 0.9800 - val_loss: 0.2937 - val_accuracy: 1.0000
Epoch 49/50 [-----] - 6s 1s/step - loss: 0.3535 - accuracy: 0.9850 - val_loss: 0.2864 - val_accuracy: 1.0000
Epoch 50/50 [-----] - 7s 1s/step - loss: 0.3487 - accuracy: 0.9817 - val_loss: 0.2795 - val_accuracy: 1.0000
  
```

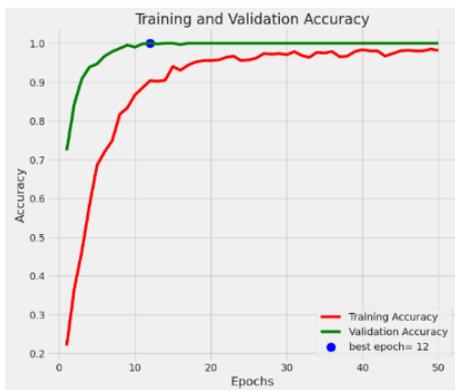
Gambar 4. Hasil Training dan Testing Model CNN

Hasil dari data pelatihan dan data pengujian dengan 50 iterasi dianalisis. Temuan yang digambarkan pada gambar 6 menggambarkan bahwa setiap iterasi menghasilkan nilai akurasi dan nilai kerugian untuk kumpulan data pelatihan dan pengujian. Nilai akurasi sangat penting dalam mengevaluasi kemandirian model yang telah dikembangkan, karena mereka menunjukkan persentase prediksi yang benar yang dibuat oleh model. Sebaliknya, nilai kerugian berfungsi sebagai metrik untuk mengukur kesalahan yang dibuat oleh jaringan saraf selama proses pembelajaran, dengan tujuan utamanya adalah meminimalkan kesalahan ini. Interaksi antara nilai akurasi dan kerugian sangat penting dalam menilai kinerja dan menyempurnakan parameter model untuk meningkatkan kemampuan prediktifnya. Sangat penting untuk mencapai keseimbangan antara mengoptimalkan akurasi dan meminimalkan kerugian untuk memastikan efektivitas model dalam membuat prediksi akurat pada data yang tidak terlihat. Upaya untuk meningkatkan kinerja model harus fokus pada penyempurnaan akurasi prediktifnya sekaligus mengurangi kerugian keseluruhan yang terjadi selama proses pelatihan.



Gambar 5. Grafik Hasil Loss Training dan Testing Model CNN

Representasi visual ini menggambarkan kurva kerugian mengenai pelatihan dan validasi model klasifikasi karakter di 50 iterasi, di mana sumbu x horizontal menunjukkan jumlah iterasi dan sumbu y vertikal menandakan nilai kerugian. Kurva kerugian pelatihan yang digambarkan dengan warna merah menunjukkan penurunan substansial dari sekitar 4,5 pada iterasi awal dan akhirnya stabil sekitar 0,5 pada saat iterasi ke-50 tercapai. Demikian pula, kurva kerugian validasi yang diilustrasikan dalam warna hijau juga menunjukkan penurunan penting selama iterasi awal dan konvergen ke nilai stabil mendekati 0,5 pada iterasi ke-50. Pada iterasi ke-50, titik biru menyoroti iterasi paling optimal, di mana kerugian validasi jatuh ke titik terendah, menunjukkan kinerja puncak model ketika dievaluasi terhadap kumpulan data validasi. Secara keseluruhan, representasi grafis ini berfungsi sebagai bukti bahwa model secara efektif belajar dari data pelatihan sambil mengelola untuk menghindari masalah overfitting yang signifikan.



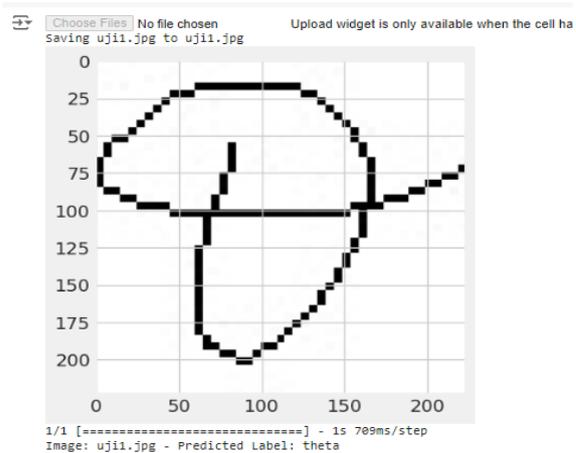
Gambar 6. Grafik Hasil Akurasi Training dan Testing Model CNN

Visualisasi ini menggambarkan perkembangan pelatihan dan akurasi validasi model tertentu dalam rentang 50 iterasi. Garis merah pada grafik mewakili akurasi pelatihan, sedangkan garis hijau sesuai dengan akurasi validasi. Awalnya, akurasi pelatihan dimulai sekitar 0,2 dan mengalami pendakian yang cepat, yang berpuncak pada nilai mendekati 1,0 sekitar iterasi ke-10. Selanjutnya, akurasi pelatihan mencapai dataran tinggi dan mempertahankan keadaan tunak mendekati 1,0 sampai akhir dari 50 iterasi. Sebaliknya, akurasi validasi menunjukkan lonjakan awal, mencapai level sekitar 0,95 dalam 10 iterasi awal sebelum menampilkan kecenderungan untuk stabil dengan fluktuasi kecil sepanjang periode pelatihan. Yang perlu diperhatikan adalah adanya penanda biru yang menunjukkan iterasi optimal, khususnya iterasi ke-12, yang menunjukkan akurasi validasi tertinggi pada grafik. Representasi grafis ini secara efektif mengkomunikasikan bahwa model telah menjalani pelatihan yang efektif tanpa menghadapi masalah overfitting yang signifikan, terbukti dalam akurasi validasi tinggi dan konsisten yang berkelanjutan setelah mencapai puncaknya.

	alpha	delta	e	gamma	infinity	lambda	mu	phi	pi	theta
Actual alpha	38	1	0	0	1	0	0	0	0	0
Actual delta	0	40	0	0	0	0	0	0	0	0
Actual e	0	0	40	0	0	0	0	0	0	0
Actual gamma	0	0	0	39	0	1	0	0	0	0
Actual infinity	0	0	0	0	40	0	0	0	0	0
Actual lambda	0	0	0	0	0	40	0	0	0	0
Actual mu	0	0	0	0	0	0	40	0	0	0
Actual phi	0	0	0	0	0	0	0	40	0	0
Actual pi	0	0	0	0	0	3	0	0	37	0
Actual theta	0	0	0	0	0	0	0	0	0	40

Gambar 7. Konfusi Matriks

Konfusi matriks yang muncul dari model klasifikasi simbol matematika menunjukkan kinerja luar biasa dengan tingkat akurasi keseluruhan melonjak hingga 99,25% yang mengesankan. Di antara 10 kelas karakter yang menjalani pengujian, yang meliputi alpha, delta, e, gamma, infinity, lambda, mu, phi, pi, dan theta, diamati bahwa hampir setiap kelas dikategorikan secara akurat. Meskipun demikian, beberapa contoh kesalahan klasifikasi dicatat, terutama yang melibatkan karakter alfa yang secara keliru diberi label sebagai delta dan gamma, serta karakter pi yang tidak akurat diidentifikasi sebagai lambda. Khususnya, kelas delta, epsilon, infinity, lambda, mu, phi, dan theta menunjukkan klasifikasi sempurna tanpa kesalahan. Temuan ini sangat menunjukkan kemandirian model yang tinggi; Namun, masih ada kebutuhan mendesak untuk peningkatan lebih lanjut untuk mengatasi terjadinya kesalahan dalam membedakan karakter serupa. Berikut adalah hasil dari uji coba yang dilakukan penulis dengan menginput citra tulisan kedalam model yang sudah ada.



Gambar 8. Hasil klasifikasi

Gambar 8, menampilkan representasi visual dari simbol matematika yaitu "theta" dalam bentuk piksel. Uji coba menunjukkan bahwa model mampu

mengidentifikasi simbol "theta" pada citra uji. Waktu yang diperlukan untuk memproses gambar adalah 1 detik. Ini menunjukkan bahwa model mampu memproses gambar dengan kecepatan yang relatif cepat.

5. KESIMPULAN DAN SARAN

Prosedur pelatihan model CNN yang ekstensif dijalankan untuk mengkategorikan simbol matematika dengan menggunakan arsitektur rumit yang mencakup berbagai lapisan seperti Xception, Conv2D, MaxPooling2D, Dropout, Flatten, dan lapisan Dense. Arsitektur model ini terdiri dari total 22.046.962 parameter, di mana 1.185.482 parameter ditetapkan sebagai dapat dilatih sementara 20.861.480 parameter sisanya dikategorikan sebagai tidak dapat dilatih. Hasil yang diperoleh dari fase pelatihan dan pengujian yang mencakup 50 iterasi menunjukkan tingkat akurasi yang luar biasa pada kumpulan data pelatihan dan validasi, mencapai stabilitas yang hampir sempurna sekitar 1,0 dan 0,95 sesuai dalam iterasi awal. Selanjutnya, grafik kurva kerugian menggambarkan penurunan substansial, akhirnya stabil mendekati 0,5 pada iterasi ke-50, menunjukkan bahwa model memahami data secara efektif tanpa menghadapi masalah overfitting yang signifikan.

Analisis matriks kebingungan menggambarkan kinerja model yang luar biasa, menawarkan tingkat akurasi keseluruhan 99,25%, terlepas dari beberapa kesalahan klasifikasi kecil yang diamati. Dengan tingkat akurasi yang luar biasa dan klasifikasi yang tepat untuk sebagian besar kelas simbol, penelitian ini akhirnya menyimpulkan bahwa implementasi model CNN terbukti sangat mahir dalam tugas mengklasifikasikan simbol matematika, meskipun saran dibuat untuk peningkatan akurasi lebih lanjut terutama untuk karakter yang sangat mirip.

DAFTAR PUSTAKA

- [1] A. M. A. Turnip, N. Fadillah, dan Munawir, "Pengenalan Tulisan Tangan Karakter Aksara Batak Toba dengan Metode Convolutional Neural Network (CNN)," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 2, hal. 242–252, 2023, doi: 10.26418/jp.v9i2.64242.
- [2] L. S. Riva *et al.*, "PERBANDINGAN ALGORITMA CNN DAN ANN DENGAN PROJECTION HISTOGRAM UNTUK KLASIFIKASI CITRA TULISAN TANGAN BERUPA ANGKA," in *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, 2022, hal. 767–779.
- [3] A. K. Putra dan H. Bunyamin, "Pengenalan Simbol Matematika dengan Metode Convolutional Neural Network (CNN)," *J. Strateg.*, vol. 2, no. November, hal. 426, 2020.
- [4] M. M. Taslim, K. Gunadi, dan A. N. Tjondrowiguno, "Deteksi Rumus Matematika pada Halaman Dokumen Digital dengan Metode Convolutional Neural Network," *J. Infra*, vol. 7, no. 2, hal. 123–129, 2019.
- [5] M. Toyib, T. D. K. Pratama, dan I. Aqil, "Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data," *Algoritma. J. Mat. Ilmu Pengetah. Alam, Kebumihan dan Angkasa*, vol. 2, no. 3, hal. 108–120, 2024.
- [6] M. R. Alwanda, R. P. K. Ramadhan, dan D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algoritma.*, vol. 1, no. 1, hal. 45–56, 2020, doi: 10.35957/algoritma.v1i1.434.
- [7] R. Rikendry dan A. Maharil, "Perbandingan Arsitektur Vgg16 Dan Resnet50 Untuk Rekognisi Tulisan Tangan Aksara Lampung," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 3, no. 2, hal. 236–243, 2022, doi: 10.33365/jatika.v3i2.2030.
- [8] T. Hidayati dan B. F. Putra, "Implementasi Deep Learning Untuk Image Classification menggunakan Convolutional Neural Network Pada Citra Wayang (Studi Kasus : SDN Leuwibatu 03)," *Sci. Sacra J. Sains, Teknol. dan Masy.*, vol. 4, no. 1, hal. 1–7, 2024.
- [9] A. S. Ningrum dan A. W. Utami, "Sistem Deteksi Tulisan Tangan Aksara Hangeul Satu Silabel Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Framework Flask," *J. Emerg. Inf. Syst. Bus. Intell.*, vol. 04, no. 04, hal. 9–16, 2023.
- [10] A. F. Abdul Fadlil, A. Maftukhah, S. Sunardi, dan T. Sutikno, "Butterfly Image Identification Using Multilevel Thresholding Segmentasi and Convolution Neural Network Classification with Alexnet Architecture," *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, hal. 1775–1785, 2024, doi: 10.12785/ijcds/1501125.