



MODEL KLASIFIKASI CONVOLUTIONAL NEURAL NETWORK PADA SISTEM PENERJEMAH AUDIO AKSARA SUNDA

Darryl Nathanael^{1)*}, Ito Wasito¹⁾

¹Universitas Pradita, Tangerang Selatan, Indonesia

Email: darryl.nathanael@student.pradita.ac.id

Abstrak

Beberapa upaya telah dilakukan untuk melestarikan aksara Sunda dan salah satunya adalah pembuatan aplikasi pengenalan aksara Sunda. Penelitian terdahulu menunjukkan bahwa teknologi pemrosesan gambar dapat digunakan untuk mengklasifikasi gambar berupa alfabet-alfabet aksara Sunda. Namun belum ada teknologi serupa yang mampu menghasilkan output berupa audio untuk memudahkan pengejaan dan pelafalan dalam pembelajaran. Tujuan penelitian ini adalah untuk mengusulkan penggunaan pemrosesan gambar untuk melakukan klasifikasi alfabet-alfabet aksara Sunda dalam bentuk tulisan tangan digital dengan output berupa teks dan audio menggunakan algoritma Convolutional Neural Network (CNN). Metode penelitian yang digunakan pada penelitian ini adalah kualitatif, karena menggunakan gambar sebagai data primer, kemudian dilanjutkan dengan pendekatan *true experimental*. Aksara dalam bentuk gambar tulisan tangan digital dimasukkan ke sistem untuk dilatih, kemudian sistem diberikan gambar di luar data pelatihan untuk diklasifikasi agar sistem dapat memprediksi label klasifikasi dari masing-masing alfabet. Sistem kemudian menerjemahkan masing-masing gambar yang dibaca ke dalam bentuk audio dan teks. Hasil berupa akurasi pengujian penelitian mencapai 96% dengan output berupa teks dan audio. Penelitian ini diharapkan dapat membantu memudahkan pengguna sistem dalam pelafalan kata yang terdapat dalam bahasa sunda, yang terkadang susah untuk diucapkan oleh orang di luar suku orang sunda serta ditampilkan terjemahan bahasa sunda dalam aksara sunda.

Kata kunci: aksara sunda; audio; convolutional neural network; klasifikasi; pemrosesan gambar.

CONVOLUTIONAL NEURAL NETWORK CLASSIFICATION MODEL IN SUNDANESE AUDIO TRANSLATOR SYSTEM

Abstract

Several efforts have been made to preserve the Sundanese script and one of them is the creation of a Sundanese script recognition application. Previous research has shown that image processing technology can be used to classify images in the form of Sundanese alphabets. But there is no similar technology that is able to produce output in the form of audio to facilitate spelling and pronunciation in learning. The purpose of this study is to propose the use of image processing to classify Sundanese script alphabets in the form of digital handwriting with output in the form of text and audio using the Convolutional Neural Network (CNN) algorithm. The research method used in this study is qualitative, because it uses images as primary data, then continued with a true experimental approach. Characters in the form of digital handwritten images are fed into the system to be trained, then the system is given images outside the training data to be classified so that the system can predict the classification labels of each alphabet. The system then translates each of the read images into audio and text form. The results in the form of research testing accuracy reached 96% with output in the form of text and audio. This research is expected to help facilitate system users in pronunciation of words contained in Sundanese, which are sometimes difficult to pronounce by people outside the Sundanese tribe and Sundanese translation in Sundanese script is displayed.

Keywords: sundanese script; audio; convolutional neural network; classification; image processing.

Submitted: 29 Mei 2023

Reviewed: 29 Mei 2023

Accepted: 12 Juli 2023

Published: 27 Juli 2023

PENDAHULUAN

Aksara Sunda adalah salah satu budaya suku Sunda, yaitu penduduk asli Indonesia yang mendiami pulau Jawa bagian barat, yang sekarang ini merupakan provinsi Jawa Barat dan Banten (Septiadi, 2018). Penggunaan bahasa Sunda sudah jarang digunakan oleh masyarakat, hal ini terlihat dari sejumlah 500 budaya kesenian Sunda nyaris punah akibat rendahnya kesadaran masyarakat Sunda dalam menjaga budayanya (Maulana, 2013). Kekhawatiran tersebut didukung oleh data dari UNESCO *Atlas of Worlds Languages* pada tahun 2020 yang menyebutkan bahwa terdapat 2.500 bahasa di dunia yang terancam punah dan 230 bahasa yang telah punah sejak 1950 akibat tidak adanya lagi penutur kata dari bahasa-bahasa tersebut (Usman, 2020). Salah satu cara untuk menghindari kepunahan bahasa daerah seperti aksara sunda adalah dengan membentuk sistem penerjemah audio aksara sunda dengan mengimplementasikan penggunaan *deep learning* yaitu *image processing*. Pola aksara Sunda dapat dikenali oleh sistem berbasis *deep learning* untuk proses klasifikasi, dalam bentuk tulisan tangan (Salsabila et al., 2019). Penelitian (Purnama et al., 2022) menggunakan *convolutional neural network* (CNN) dalam mengklasifikasi aksara Sunda dengan akurasi yang cukup baik, bahkan tergolong tinggi. Penelitian ini membentuk arsitektur dengan 3 (tiga) layer *convolutional* dengan *weight* masing-masing sebesar 32, 64, 128 unit. Penelitian ini juga dilengkapi dengan fungsi aktivasi ReLu dan tambahan layer MaxPooling untuk mengurangi jumlah matriks. Setelahnya, 1 (satu) buah *hidden layer* juga ditambahkan dengan fungsi aktivasi softmax dan kernel sebesar 1024 kernel. Sistem pada penelitian ini kemudian dikembangkan menjadi aplikasi *mobile* dengan antarmuka yang terbilang cukup mudah digunakan, meskipun tidak berfokus pada pembuatan antarmuka layaknya penelitian (Tamyiz & Umam, 2017), (Sugiri et al., 2019), dan (Nugraha et al., 2021). Hasil pengujian menunjukkan bahwa algoritma CNN dengan pengujian secara objektif, yaitu dengan penginputan gambar ke sistem secara manual, berhasil mencapai 98% dengan output berupa citra aksara digital dan terjemahannya ke dalam bentuk teks.

Dibandingkan dengan penelitian lain, seperti penelitian (Setyansyah et al., 2019) yang menggunakan metode *active contour*, penelitian (Salsabila et al., 2019) yang menggunakan *Histogram of Oriented Gradients* (HOG) dan gabungannya dengan *Support Vector Machine* (SVM) seperti penelitian (Taufiqurahman, 2019), penelitian (Apriyanto, 2018) yang menggunakan *Feature Extraction*, atau penelitian (Riansyah et al., 2017) yang menggunakan metode *Learning Vector Quantization*, penelitian-penelitian terdahulu yang menggunakan CNN menunjukkan bahwa CNN lebih unggul dalam hal akurasi untuk melakukan klasifikasi citra atau gambar.

Beberapa penelitian terdahulu seperti yang sudah dibahas di atas menunjukkan bahwa CNN dapat digunakan untuk melakukan klasifikasi citra aksara Sunda dengan baik, bahkan unggul diantara beberapa metode atau algoritma lainnya. CNN dapat dijelaskan sebagai bagian dari *deep neural network* yang menggunakan arsitektur khusus agar dapat terbiasa untuk mengklasifikasi gambar (Chollet, 2021). CNN mencakup tiga ide dasar sebagai tiang pendiri algoritma tersebut. Ide-ide tersebut antara lain *local receptive fields* atau bidang reseptif lokal, *shared weights and biases*, dan *pooling layers* atau lapisan penyatuan. Ide-ide dasar tersebut membuat *convolutional network* dapat melakukan pelatihan data dengan cepat. Kecepatannya dalam melakukan pelatihan data membuat CNN bahkan digunakan untuk membantu *deep network* lainnya untuk dapat mengklasifikasi gambar dengan baik (Nielsen, 2015). Untuk menunjang CNN dalam melakukan klasifikasi citra atau gambar, penelitian ini menggunakan *library* TensorFlow. TensorFlow adalah *library* milik Google untuk *machine learning* berskala besar yang menyederhanakan komputasi yang seringkali rumit dengan merepresentasikannya sebagai grafik dan memetakan bagian grafik secara efisien ke mesin dalam kluster atau ke prosesor dari satu mesin (Shukla & Fricklas, 2018). TensorFlow dirancang untuk

merampingkan proses pengembangan dan pelaksanaan *advanced analytics applications* untuk pengguna seperti *data scientists*, *statisticians*, dan *predictive modelers* (Vaughan, 2018). Platform TensorFlow membantu penggunaannya menerapkan praktik terbaik untuk *data automation* atau otomasi data, *model tracking* atau pelacakan model, *performance monitoring* atau pemantauan kinerja, dan *model retraining* atau pelatihan ulang model.

Setelah melakukan studi literatur, belum ditemukan adanya teknologi serupa yang mampu menghasilkan *output* berupa audio untuk memudahkan pengejaan dan pelafalan dalam pembelajaran. Hasil dari penelitian-penelitian terdahulu hanya sebatas pengenalan pola aksara setelah proses klasifikasi dan terjemahannya ke dalam bentuk teks. Beberapa diantaranya mengharuskan pengguna melakukan input data berupa tulisan atau alfabet aksara Sunda secara manual menggunakan *keyboard* dan beberapa diantaranya hanya sebatas menguji keunggulan dari algoritma yang digunakan saja.

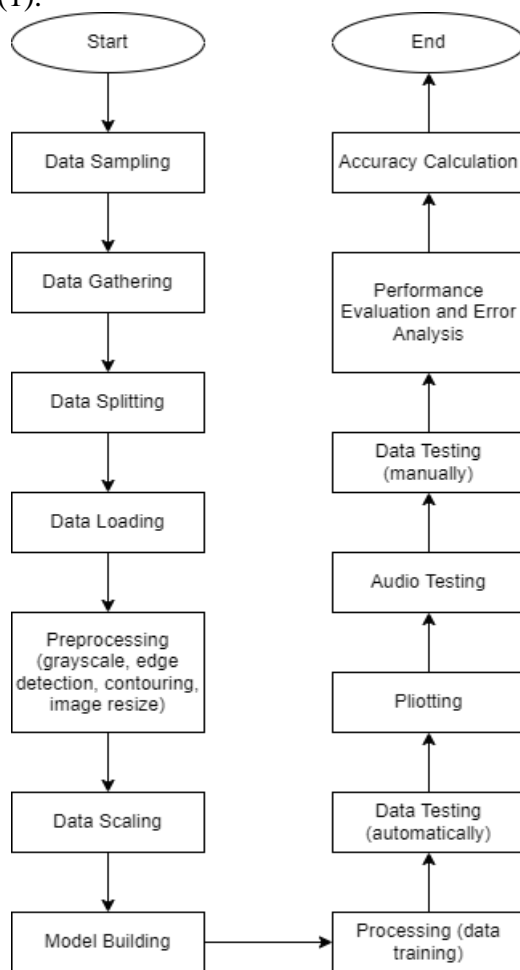
Membahas kembali penelitian (Purnama et al., 2022), penelitian tersebut menunjukkan bahwa algoritma CNN dengan pengujian secara objektif, yaitu dengan penginputan gambar ke sistem secara manual, berhasil mencapai 98% dengan output berupa citra aksara digital dan terjemahannya ke dalam bentuk teks. Penelitian ini belum dilengkapi dengan terjemahannya ke audio. Penelitian (Kahfie, 2022) menggunakan CNN dengan bantuan Extreme Learning Machine (ELM) dan backpropagation dalam melakukan klasifikasi citra aksara Sunda Ngalagena dalam bentuk tulisan tangan. Penelitian ini hanya memberikan output berupa hasil akurasi sebesar 94,391% dengan waktu pelatihan sebesar 2,226 detik yang dievaluasi oleh metode K-Fold Cross Validation. Penelitian (Rahmawati et al., 2021) menggunakan CNN dalam melakukan klasifikasi aksara Sunda Swara dalam bentuk tulisan tangan. Hasil dari penelitian ini menunjukkan akurasi pengujian mencapai 85,71% pada dataset dari penelitian (Amalia et al., 2020) dan 92,02% dengan menguji dataset kedua. Output yang diberikan oleh penelitian ini hanyalah perbandingan metode atau algoritma CNN dengan Jaringan Saraf Tiruan (JST). Penelitian (Kirana et al., 2020) menggunakan CNN dalam melakukan klasifikasi aksara Sunda Ngalagena dan Swara dalam berbagai bentuk. Hasil dari penelitian ini menunjukkan akurasi pengujian sebesar 72,41% dengan menggunakan citra dari buku elektronik, 100% dengan citra dari huruf komputer, 87,1% dengan citra yang ditangkap oleh kamera ponsel, dan 85,5% dengan citra yang ditangkap oleh alat pemindai. Output yang diberikan oleh penelitian ini hanya terpusat kepada keandalan CNN dalam melakukan klasifikasi citra dalam berbagai bentuk.

Keterbatasan penelitian-penelitian tersebut membuat pengguna sulit mempelajari cara membaca, melafalkan, atau mengeja masing-masing alfabet dari aksara Sunda dan menjadi alasan mengapa aksara Sunda sudah mulai ditinggalkan. Penelitian ini mengusulkan penggunaan *image processing* dengan algoritma CNN untuk melakukan klasifikasi alfabet-alfabet aksara Sunda Ngalagena dalam bentuk tulisan tangan digital. Penerapan teknik klasifikasi citra dapat dipadukan dengan aksara Sunda sebagai penerjemah dari bentuk gambar atau citra ke dalam bentuk teks dan audio pelafalan atau pengejaan masing-masing alfabet. Penelitian ini diharapkan membantu memudahkan pengguna sistem dalam pelafalan kata yang terdapat dalam bahasa sunda yang terkadang susah untuk diucapkan oleh orang diluar suku sunda sehingga mempermudah orang yg ingin mempelajari bahasa sunda dan aksaranya. Selain itu juga diharapkan dengan adanya penelitian ini, calon pembelajar aksara Sunda maupun calon peneliti dengan topik yang relevan dapat menjadikan penelitian ini sebagai salah satu referensi.

METODE

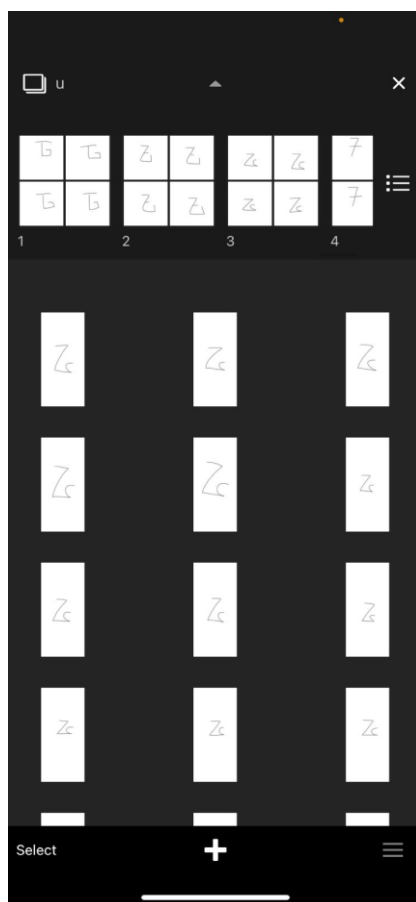
Metode penelitian ini menggunakan data kualitatif, karena menggunakan gambar sebagai data penelitian, yang kemudian dilanjutkan dengan pendekatan *True Experimental* pada bagian

data loading, *data preprocessing*, dan *data processing*. Gambaran dari tahapan penelitian akan ditunjukkan pada gambar (1).

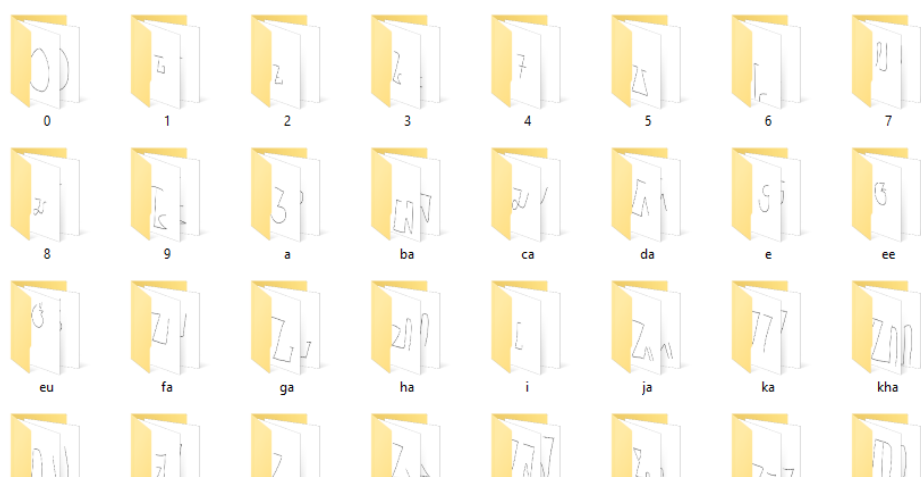


Gambar 1. Bagan Umum Metode Penelitian yang Diusulkan

Penelitian diawali dengan tahap *data sampling*. Data yang diambil untuk dijadikan sample data dalam proses *data sampling* merupakan data dalam bentuk gambar tulisan tangan digital dari masing-masing alfabet dalam aksara Sunda yang dibuat pada handphone dengan bantuan tools berupa aplikasi Sketchbook dengan format .png. Adapun aksara yang digunakan adalah Aksara Ngalagena, Aksara Swara, dan Angka. Pembagian kelas klasifikasi dilakukan berdasarkan masing-masing alfabet pada jenis-jenis aksara Sunda yang digunakan dengan total 42 kelas dengan ketentuan penulisan dan pengambilan gambar masing-masing alfabet dilakukan sebanyak 30 kali. Kelas-kelas tersebut terdiri dari 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, ba, ca, da, e, é, eu, fa, ga, ha, i, ja, ka, kha, la, ma, na, nga, nya, o, pa, qa, ra, sa, sya, ta, u, va, wa, xa, ya, za. Pembuatan *dataset* dilakukan sebagai teknik pengumpulan data atau *data gathering*. Tahap *data sampling* dan *data gathering* berturut-turut dapat dilihat pada gambar (2) dan gambar (3).



Gambar 2. Data Sampling



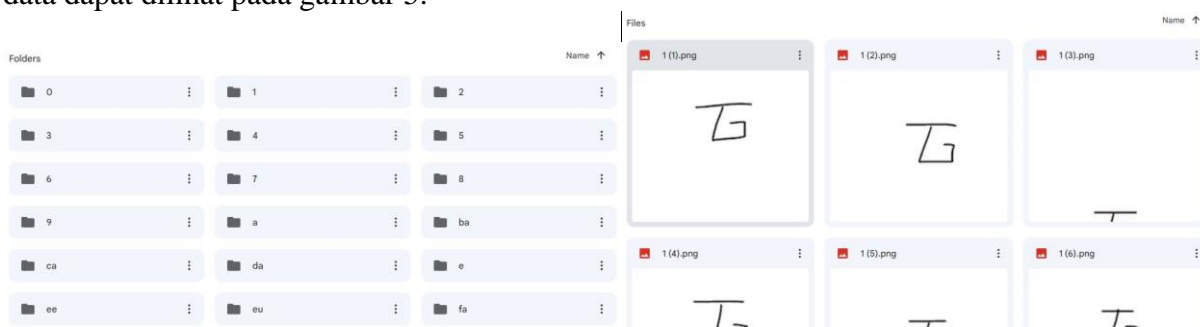
Gambar 3. Data Gathering

Penelitian lalu masuk ke tahap berikutnya, yaitu tahap *data splitting*. *Data splitting* adalah proses pemisahan data untuk pelatihan dan data untuk pengujian. Tahap ini meliputi pembagian kelas, pembagian data, pengacakan data, dan eksplorasi data. Pada tahap ini, data dibagi ke dalam dua *folder* secara manual, yaitu folder Train dan Test. Perbandingan data pada keduanya dibuat sebesar 8:2. *Data splitting* dapat dilihat pada gambar 4.



Gambar 4. Pembagian Dataset Menjadi *Train* dan *Test*

Setelah melakukan *splitting*, penelitian kemudian dilanjutkan ke tahap data loading atau akuisisi data. *Dataset* yang berisi gambar dari masing-masing alfabet pada aksara Sunda dengan jenis file berupa .png di unggah pada Google Drive untuk mempermudah Google Colab memanggil dataset. Pembacaan citra aksara ke dalam sistem dilakukan dengan menggunakan fungsi *imread* (Ferdiano, 2019). Proses pemanggilan dapat dilakukan dengan mendeklarasikan path dari dataset dengan terlebih dahulu melakukan mount terhadap Google Drive. Akuisisi data dapat dilihat pada gambar 5.



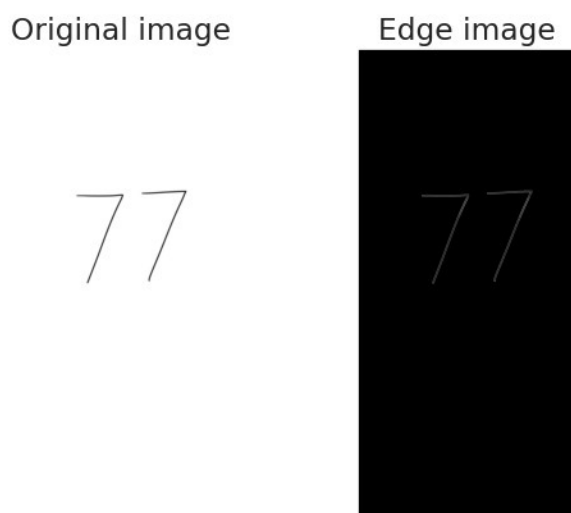
Gambar 5. Data Loading

Setelah data sudah berhasil diakuisisi pada tahap *data loading*, penelitian kemudian lanjut ke tahap *preprocessing*. Tujuan dari tahapan ini adalah untuk meningkatkan kualitas dari citra masukan yang akan diapat (Farhan et al., 2017). Data yang sudah dikumpulkan pada tahap *data gathering* kemudian mengalami proses *preprocessing*. Tahap ini mencakup *grayscale*, *edge tracking*, *contouring*, dan *image resize*.

Grayscale adalah proses konversi warna *default* gambar dari BGR atau *Blue, Green, Red* diubah menjadi GRAY. *Grayscale* adalah salah satu langkah awal untuk menerapkan algoritma Canny Edge Detection. Tujuan dari pemberlakuan *grayscale* adalah untuk mengubah nilai intensitas dari seluruh piksel pada masing-masing gambar berdasarkan derajat keabuan. Secara matematika, pemberlakuan *grayscale* pada data berupa gambar dapat dilakukan menggunakan rumus (1) (Taufiqurahman, 2019).

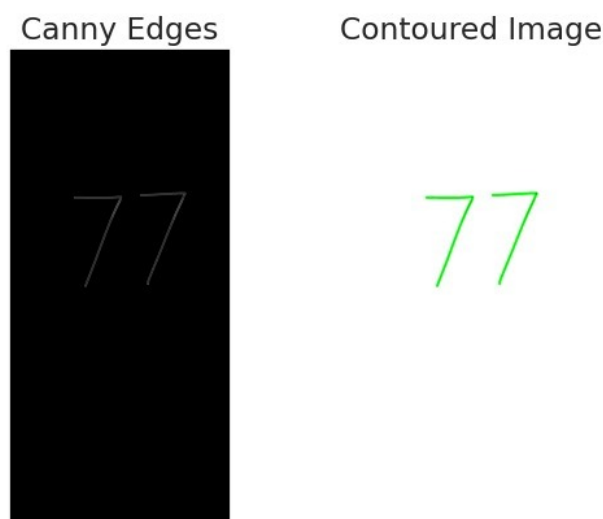
$$Grayscale = 0.299R + 0.587G + 0.144B \quad (1)$$

Edge tracking adalah proses pencarian tepi-tepi pada gambar setelah mengalami konversi warna *grayscale*. Tujuan dari deteksi tepi ini adalah untuk menegaskan garis-garis tepi pada gambar yang tentunya akan memudahkan sistem untuk kemudian mendeteksi gambar dengan kualitas yang relatif lebih jelas atau tidak *blurry*. Perbedaan data sebelum dan sesudah mengalami *edge tracking* yang tertera pada gambar 6.



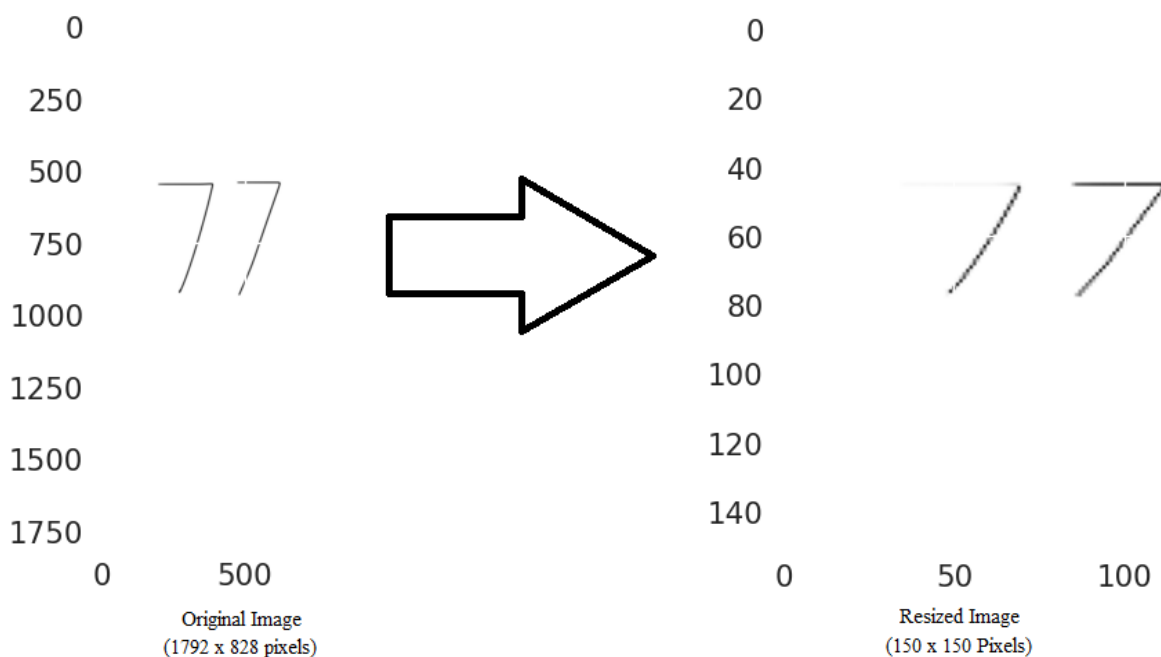
Gambar 6. Perbedaan data berupa gambar asli dengan data yang sudah mengalami *edge tracking*

Contouring merupakan proses pembentukan model kurva yang menjadi penghubung seluruh titik kontinu atau yang berkelanjutan, yang memiliki warna dan/atau intensitas yang sama. Tujuan dari pemberlakuan *contouring* ini adalah untuk mempermudah sistem dalam melakukan analisis bentuk maupun deteksi untuk pengenalan objek. *Contour* dapat bekerja dengan baik dengan menghasilkan akurasi yang baik apabila gambar yang digunakan adalah gambar *grayscale*. Perbedaan data sebelum dan sesudah mengalami *contouring* tertera pada gambar (7).



Gambar 7. Perbedaan Data Berupa Gambar Sebelum dan Sesudah Mengalami Proses *Contouring*

Tahap *image resize* diperlukan karena model klasifikasi membutuhkan data masukan dengan piksel berukuran sama 150x150 agar proses komputasi menjadi lebih ringan. *Resize* berarti mengubah ukuran gambar agar seluruh data memiliki jumlah pixel yang seragam (Riansyah et al., 2017). Proses ini dilakukan dengan menggunakan fungsi `cv2.resize(image, (150,150))` dengan tambahan fungsi *loop* dari *for* untuk membantu memudahkan komputasi. Contoh *image resize* tertera pada gambar 8.



Gambar 8. Data Mengalami Tahap *Image Resize*

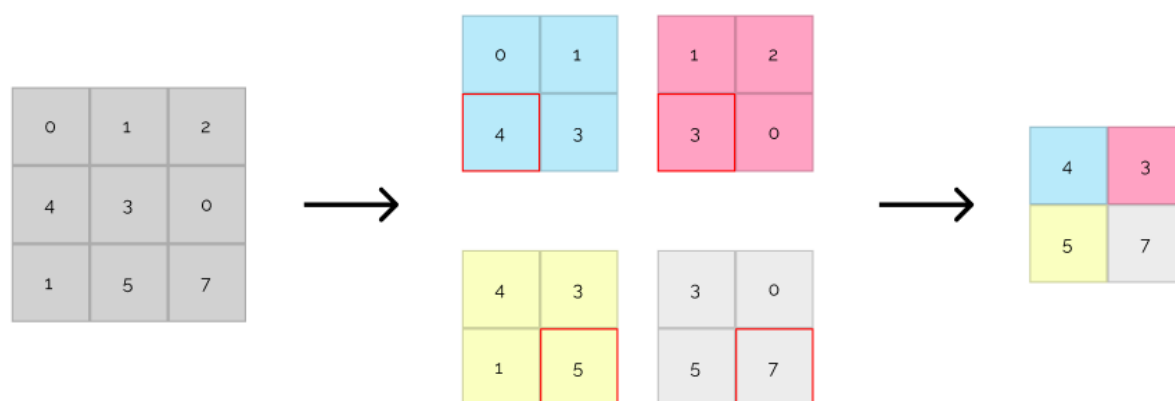
Pada tahap *preprocessing*, penelitian kemudian masuk ke tahap *data scaling*. *Data scaling* adalah proses pembagian seluruh nilai pada data dengan nilai maksimum pada data. Nilai maksimum pada data yang berupa gambar adalah nilai piksel terbesar pada masing-masing gambar. Nilai dari piksel lainnya pada gambar tersebut akan dibagi oleh nilai maksimum. Tujuan dari *data scaling* adalah untuk membuat jumlah piksel menjadi hanya antara angka 0 dan 1, sehingga memudahkan komputasi dalam membuat perbandingan dalam bentuk persentase. Fungsi `max()` dapat digunakan pada data berupa gambar untuk mengetahui nilai piksel maksimum pada data tersebut. *Data scaling* pada *data train* dan *data test* berturut-turut dapat dilakukan dengan persamaan (2) dan (3).

$$\text{train_images} = \text{train_images} / \text{image.max}() \quad (2)$$

$$\text{test_images} = \text{test_images} / \text{image.max}() \quad (3)$$

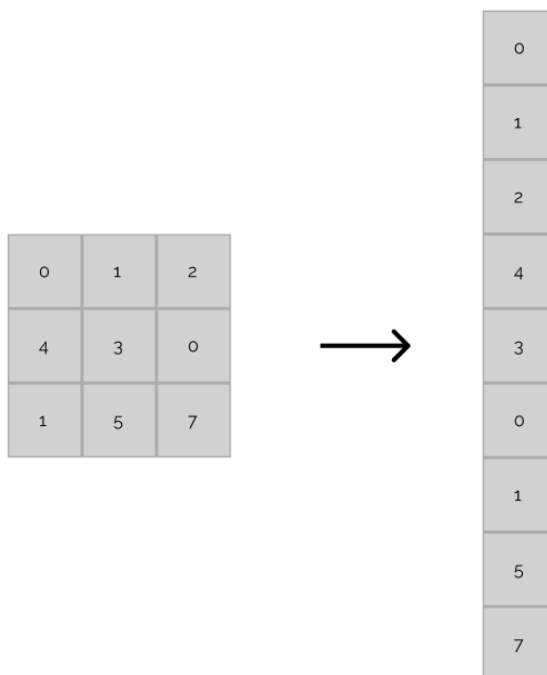
Setelah berhasil melakukan normalisasi, penelitian berlanjut ke pembuatan model atau *model building*. *Model building* adalah tahap pembuatan model yang terdiri dari beberapa lapisan (*layer*) *neural network* dengan masing-masing lapisan dilengkapi dengan beberapa parameter tersendiri. Parameter yang dimaksud seperti pada penelitian ini adalah *weight*, *filter/kernel size*, jumlah *stride*, *image input size*, dan fungsi aktivasi yang digunakan. *Weight* berfungsi untuk menentukan berapa jumlah *filter/kernel* yang akan digunakan dalam sekali proses pengolahan data pada masing-masing *layer*. Semakin besar jumlahnya, semakin berat proses komputasinya, namun proses pengolahan citra akan lebih banyak. Semakin banyak proses pengolahan citra tidak menjamin hasil yang lebih baik. *Filter/kernel size* adalah ukuran *filter/kernel* dalam satuan piksel yang akan digunakan dalam proses pengolahan data pada masing-masing *layer* dengan jumlah maksimum yaitu sama dengan ukuran gambar yang di-*input*. Jumlah *stride* dapat diartikan sebagai jumlah pergerakan *filter/kernel* setiap langkah dalam satuan piksel. Secara *default*, *stride* memiliki besaran 1x1 piksel. Semakin besar *stride* yang digunakan, maka beban komputasi akan semakin ringan. Namun jumlah informasi pada gambar akan semakin berkurang, atau dengan kata lain akan ada piksel yang terlewat. *Image input size* adalah ukuran gambar yang akan dimasukkan ke dalam model untuk mengalami pengolahan citra. Fungsi aktivasi adalah fungsi yang bertugas untuk mengaktifkan atau

menonaktifkan neuron atau piksel. Pada penelitian ini, fungsi aktivasi yang digunakan adalah Rectified Linear Unit (ReLU) dan Softmax. ReLU adalah fungsi aktivasi yang mengubah nilai negatif pada neuron atau piksel menjadi 0 (nol). Sedangkan Softmax adalah fungsi aktivasi klasifikasi yang mampu mengelompokkan output ke dalam salah satu dari beberapa kelas (*multiclass classification*). Penelitian ini menggunakan algoritma CNN sebagai algoritma klasifikasi yang terdiri dari *convolution layers*, yang pada penelitian ini terdiri dari lapisan Conv2D dan MaxPooling2D, serta *fully connected layers*, di mana lapisan-lapisan yang digunakan adalah lapisan Flatten, Dense, dan Dropout. Lapisan Conv2D adalah lapisan yang berfungsi untuk mengolah citra berbasis 2 (dua) dimensi menggunakan *filter/kernel* dengan tujuan untuk memperjelas atau memperhalus citra tersebut (konvolusi) (Gazali et al., 2012). Cara kerja konvolusi adalah dengan mengubah nilai piksel asli dengan nilai piksel baru, yang berdekatan dengan nilai piksel aslinya. Proses pengolahan citra menggunakan teknik konvolusi ini tidak akan mengubah ukuran citra. Lapisan MaxPooling2D adalah lapisan pengolahan citra berbasis 2 (dua) dimensi yang berfungsi untuk mengurangi ukuran matriks dari *output* yang diberikan oleh lapisan sebelumnya. Pengurangan tersebut dilakukan dengan hanya mengambil 1 (satu) nilai piksel maksimum sesuai *pool size*. Secara *default*, *pool size* memiliki besaran 2x2 piksel dengan *stride* sebesar 1x1 piksel. Artinya setiap 2x2 piksel dengan jarak langkah 1x1 piksel dari piksel pertama, hanya akan diambil 1 (satu) piksel dengan nilai tertinggi. Visualisasi pengurangan ukuran matriks menggunakan lapisan MaxPooling2D tertera pada gambar 9.



Gambar 9. Visualisasi Cara Kerja Max Pooling Layer Dengan 2x2 Pool Size dan 2x2 Stride

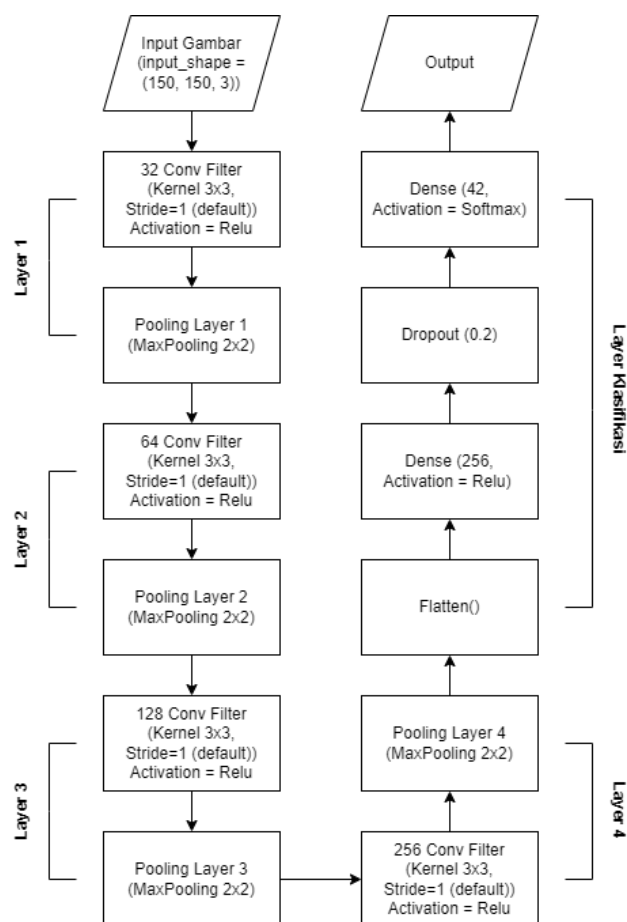
Lapisan Flatten berfungsi untuk memadatkan tinggi dan lebar dari ukuran *output* yang diterimanya. Pemadatan tinggi dan lebar ini dilakukan dengan mengambil seluruh nilai piksel dari *output* (*feature map*) yang diterima dari lapisan sebelumnya dan mengubah ukuran matriks menjadi 1 (satu) dimensi (vektor). Visualisasi pemadatan tinggi dan lebar ukuran *output* ditampilkan pada gambar 10.



Gambar 10. Visualisasi Cara Kerja *Flatten Layer*

Lapisan Dense dapat diartikan sebagai lapisan yang berfungsi untuk memadatkan nilai-nilai dari *ouput* yang diterima dari lapisan sebelumnya menjadi satu nilai sesuai dengan *output size*. Untuk lapisan Dense terakhir, *output size* disesuaikan dengan jumlah kelas klasifikasi. Hal tersebut dilakukan agar sistem dapat mengelompokkan *output* ke dalam kelas yang sesuai. Lapisan Dropout adalah lapisan regulasi yang berfungsi untuk memilih beberapa neuron secara acak untuk kemudian dikeluarkan dari pelatihan.

Model disusun untuk menerima *input*-an citra atau gambar dengan ukuran 150x150 piksel, sesuai dengan hasil dari proses *image resize* pada tahap *preprocessing*. Model terdiri dari 4 (empat) *convolutional layer*, dengan *weight* berturut-turut sebesar 32, 64, 128, dan 256, dan 1 (satu) *classification layer*. Masing-masing *convolutional layer* dilengkapi dengan 1 (satu) *max pooling layer*, *kernel* dengan ukuran 3x3 piksel, fungsi aktivasi ReLu, serta menggunakan *stride* secara *default*, yaitu 1. *Classification layer* dibuka dengan *flatten layer*, kemudian dilanjut dengan *dense layer* dengan *weight* sebesar 256 dan menggunakan fungsi aktivasi ReLu. *Classification layer* juga dilengkapi oleh *dropout layer* dengan *droupout parameter* sebesar 0,2 atau 20%. Kemudian, *classification layer* ditutup dengan *dense layer* dengan fungsi aktivasi *softmax* dan *weight* sesuai dengan jumlah kelas klasifikasi pada penelitian ini, yaitu 42 kelas. Diagram alur dari lapisan-lapisan *neural network* yang menjadi penyokong penelitian sesuai dengan algoritma yang digunakan serta tabel arsitektur parameter CNN yang digunakan pada penelitian ini secara berturut-turut dapat dilihat sebagaimana tertera pada gambar 11 dan tabel 1.



Gambar 11. Diagram alur lapisan-lapisan *neural network* dengan algoritma CNN

Tabel 1. Arsitektur CNN

Layer Type	Parameters	Output Shape	Parameters
Image Input	-	150x150x3	-
Conv 1 + ReLu	32 Filter (3x3)	148x148x32	896
Max Pooling	(2x2)	74x74x32	0
Conv 2 + ReLu	64 Filter (3x3)	72x72x64	18496
Max Pooling	(2x2)	36x36x64	0
Conv 3 + ReLu	128 Filter (3x3)	34x34x128	73856
Max Pooling	(2x2)	17x17x128	0
Conv 4 + ReLu	256 Filter (3x3)	15x15x256	295168
Max Pooling	(2x2)	7x7x256	0
Flatten	-	12544x1x1	0
Dense + ReLu	256 Neurons = 1 Layer	256	3211520
Dropout	20%	-	0
Dense + Softmax	42 Classes	42	10794
Total Params (Trainable)		3.610.730 Params	

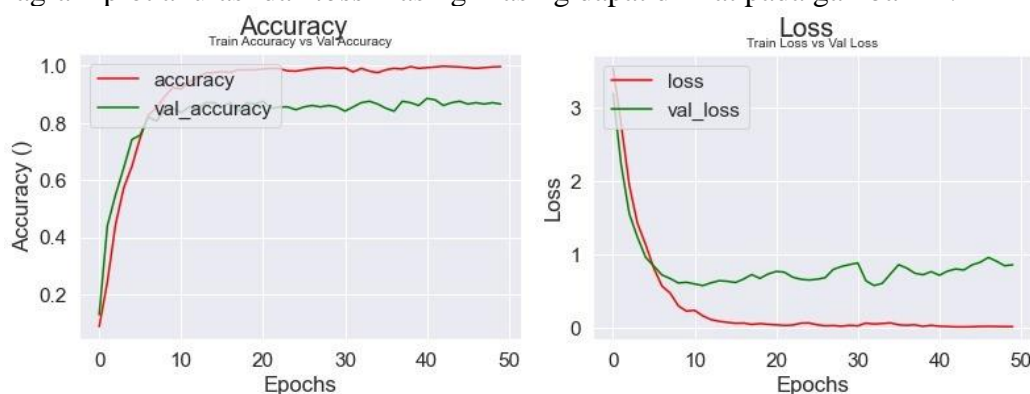
Setelah selesai melakukan pembuatan model, penelitian kemudian dilanjutkan ke tahap *data processing*. Metode penelitian yang digunakan pada penelitian ini, terutama dalam tahap *data processing*, adalah pendekatan *True Experimental*, dengan melakukan *data training* secara berulang-ulang terhadap *dataset* yang dimiliki menggunakan aplikasi yang menerapkan algoritma CNN sebagai algoritma pemrosesan gambar dalam tool OpenCV dan library

TensorFlow. *Data training* adalah proses dimana model dilatih menggunakan *dataset* yang sudah mengalami *splitting* dan *scaling*. Beberapa parameter juga diperlukan untuk melakukan proses ini. Setelah melakukan *data training*, aplikasi kemudian diuji untuk pengimplementasian ke alfabet-alfabet dalam aksara Sunda untuk melakukan klasifikasi berdasarkan hasil dari *data training* yang dilakukan sebelumnya ke dalam kelas-kelas yang telah dibuat. Jika akurasi belum menunjukkan angka yang diharapkan, *data processing* terus dilakukan secara berulang sampai pengujian akurasi menggunakan aplikasi tersebut mencapai hasil berupa akurasi yang memuaskan.

HASIL DAN PEMBAHASAN

Data yang sudah diakuisisi kemudian akan dilatih pada proses pelatihan data atau *data training* menggunakan model yang sudah dibuat. Pembagian data pelatihan dan data validasi pada penelitian ini adalah 8:2. Jumlah *epoch* yang digunakan sebanyak 50 kali dengan *validation cycle* sebanyak 7 *cycle*. Dengan model dan dataset ini, akurasi pelatihan berhasil mencapai 99,63% dan akurasi validasi berhasil mencapai 88,12%. Sementara itu, model yang sudah dilatih kemudian diuji menggunakan data pengujian sebagai hasil dari evaluasi model. Akurasi pengujian yang didapat dari model ini berhasil mencapai 82,14%.

Setelah melakukan *data training* dan *testing*, penelitian dilanjutkan ke tahap *performace plotting and evaluation* untuk melihat performa dari model yang dibuat dan kemudian dievaluasi. Pada tahap ini, hasil dari *data training* divisualisasikan ke dalam bentuk diagram plot. Diagram plot akurasi dan *loss* masing-masing dapat dilihat pada gambar 12.



Gambar 12. Diagram Plot Akurasi dan Loss Dari Pelatihan dan Validasi

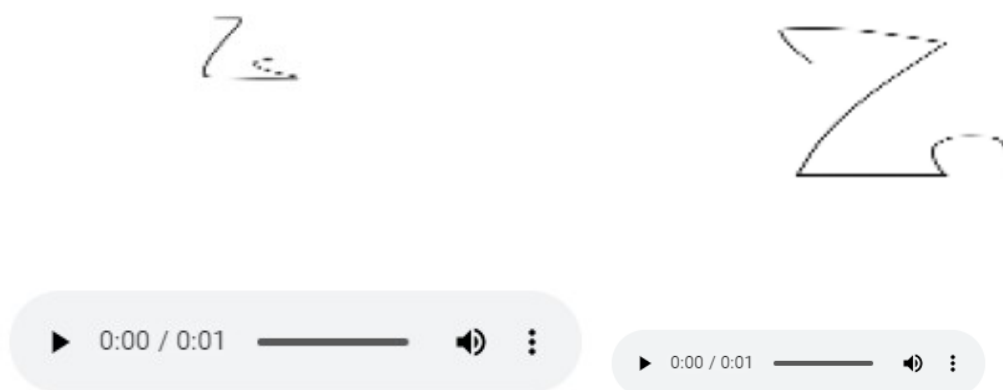
Berdasarkan gambar 12, akurasi pelatihan dan akurasi validasi yang dihasilkan mulai dari *epoch* ke-10 sudah mulai stabil, namun tetap secara perlahan meningkat, begitu juga dengan *loss* dari validasi data. Pada grafik *loss*, dapat dilihat bahwa *validation loss* pada awal *epoch* lebih rendah daripada *train loss*. Namun seiring bertambahnya *epoch*, *validation loss* akan lebih tinggi dari pada *train loss* dan semakin bertambah tinggi, sedangkan *train loss* perlahan-lahan menurun. Selain itu, *gap* atau jarak antara *train loss* dan *validation loss* pada gambar dapat dikategorikan sebagai *gap* yang besar. Melihat bahwa akurasi penelitian berhasil mencapai 99,63% namun akurasi validasi dan pengujian tidak mencapai 90%, didapatkan bahwa model yang diajukan masih *overfitting*. Artinya, model dapat dengan baik mengklasifikasi data pada *training set*, sedangkan kesulitan dalam melakukan klasifikasi terhadap data pada *validation set* dan *train set*. Bila disesuaikan dengan penelitian (Brownlee, 2019) dan (Soleymani, 2022), adanya *gap* pada grafik *loss* yang besar adalah ciri-ciri model yang *overfitting*. Solusi untuk masalah ini menurut buku (El Naqa & Murphy, 2015) adalah pada pembuatan model yang seharusnya sesederhana mungkin, tetapi tidak boleh terlalu sederhana, karena sebaliknya hal tersebut malah dapat menyebabkan *underfitting*. Selain itu, melihat bahwa *validation loss* pada

epoch tertentu mulai meningkat, didapatkan bahwa pelatihan data yang terlalu lama juga menyebabkan *overfitting* pada model yang diajukan. Untuk mencegahnya menurut buku (Chollet, 2021), pengurangan *epoch* dapat diberlakukan pada proses *data training*. Pemberlakuan *early stopping* seiring dengan berjalannya proses *data training* juga dapat mencegah terjadinya proses pelatihan data yang terlalu lama sehingga dapat mencegah terjadinya *overfitting*. Jika dibandingkan dengan penelitian-penelitian terdahulu, dataset yang digunakan pada penelitian ini masih tergolong sedikit, yang mungkin menjadi faktor tambahan terjadinya *overfitting*.

Setelah melakukan uji dan evaluasi performa, penelitian lanjut ke tahap pengetesan model atau *data testing*, sekaligus untuk melihat *error* atau galat dari model yang diajukan. Pada tahap ini, model diuji dengan penginputan gambar ke sistem secara otomatis menggunakan fungsi dari model yang diajukan (objektif) dengan melakukan prediksi kelas atau label klasifikasi berdasarkan gambar-gambar yang diperoleh dari data pengujian. Gambar secara acak diambil untuk diprediksi, kemudian gambar akan diklasifikasi berdasarkan kelas-kelas yang ada. Kemudian hasilnya ditampilkan dalam bentuk diagram plot, dan dilengkapi dengan kemampuannya menghasilkan output berupa audio yang sesuai dengan hasil prediksi dan klasifikasi seperti yang tertera pada gambar 13.

Image #20: Class: 3

Image #671: Class: o



Gambar 13. Hasil Pengujian Model

Gambar 13 menunjukkan output dari sistem ketika diuji dengan penginputan gambar ke sistem secara manual (subjektif). Sistem berhasil mengklasifikasi gambar *input*-an dan memberikan prediksi kelas alfabet sesuai dengan gambar. Selain itu sistem juga berhasil memberikan *audio output* sesuai dengan pelafalan masing-masing kelas alfabet. Untuk analisis galat, prediksi dan klasifikasi sekaligus dilakukan sebanyak 25 kali untuk melihat peluang munculnya galat seperti yang tertera pada gambar 14.

Some examples of images of the dataset

Some examples of images of the dataset



Gambar 14. Hasil Prediksi dan Klasifikasi Label Data Pengujian Secara Acak

Galat dari hasil klasifikasi dengan penginputan gambar ke sistem secara manual (subjektif) yang tertera pada gambar 14 dapat dievaluasi dan dianalisis untuk mengetahui akurasi pengujianya. Evaluasi galat dari hasil klasifikasi pada gambar 14 dilakukan dengan mencocokkan masing-masing gambar aksara Sunda dengan hasil klasifikasi labelnya secara manual. Hasil evaluasi analisis galat dari data yang tertera pada gambar 14 ditampilkan dalam bentuk tabel 2 dan tabel 3.

Tabel 2. Tabel Analisis Galat Pengujian Pertama

Prediksi Ke	Kelas Gambar	Kelas Klasifikasi	Benar/Salah
1	0	0	Benar
2	0	0	Benar
3	0	0	Benar
4	0	0	Benar
5	0	0	Benar
6	2	2	Benar
7	3	3	Benar
8	5	5	Benar
9	5	5	Benar
10	5	5	Benar
11	7	7	Benar
12	7	7	Benar
13	8	8	Benar
14	9	9	Benar
15	9	9	Benar
16	9	9	Benar
17	9	9	Benar
18	a	a	Benar
19	da	da	Benar
20	da	da	Benar
21	é	é	Benar
22	a	é	Salah

23	eu	eu	Benar
24	ga	ga	Benar
25	ha	ha	Benar

Tabel 3. Tabel Analisis Galat Pengujian Pertama

Prediksi Ke	Kelas Gambar	Kelas Klasifikasi	Benar/Salah
1	1	1	Benar
2	7	7	Benar
3	7	7	Benar
4	7	7	Benar
5	5	5	Benar
6	4	4	Benar
7	2	2	Benar
8	2	2	Benar
9	8	8	Benar
10	8	8	Benar
11	9	9	Benar
12	9	9	Benar
13	a	a	Benar
14	e	e	Benar
15	ca	ca	Benar
16	ca	ca	Benar
17	é	é	Benar
18	a	é	Salah
19	ka	ka	Benar
20	ka	ka	Benar
21	i	i	Benar
22	qa	qa	Benar
23	ra	ra	Benar
24	sya	sya	Benar
25	ta	ta	Benar

Berdasarkan tabel 2 dan 3, secara perhitungan masing-masing tabel, maka akurasi yang diperoleh dari analisis performansi sistem dapat dilakukan menggunakan perhitungan *accuracy* dari *confusion matrix*, yaitu rasio dari seluruh prediksi atau klasifikasi benar (positif dan negatif) dengan keseluruhan data, yang dapat dihitung menggunakan rumus (4) (Ungkawa et al., 2022).

$$Accuracy = \frac{24}{25} \times 100 = 96\% \quad (4)$$

Akurasi yang didapat menggunakan persamaan di atas dapat digolongkan sebagai akurasi yang tinggi jika dibandingkan dengan beberapa penelitian terdahulu yang juga menggunakan persamaan di atas. Salah satu penelitian terdahulu yang menggunakan algoritma CNN dengan data berupa tulisan tangan digital berhasil mencapai akurasi sebesar 98% secara subjektif (Purnama et al., 2022) dengan menggunakan persamaan di atas. Masih dengan algoritma yang sama, beberapa penelitian lainnya berhasil mencapai akurasi masing-masing sebesar 85.71% secara subjektif (Amalia et al., 2020) dan 92.02% secara subjektif (Rahmawati et al., 2021) menggunakan data berupa gambar tulisan tangan. Penelitian (Kirana et al., 2020) yang juga menggunakan CNN berhasil mencapai akurasi sebesar 72.4% secara subjektif menggunakan

data berupa citra dari buku elektronik, 100% secara subjektif menggunakan data berupa citra dari huruf komputer, 87,1% secara subjektif dengan citra yang ditangkap oleh kamera ponsel, dan 85,5% secara subjektif dengan citra yang ditangkap oleh alat pemindai.

Untuk penelitian (Kirana et al., 2020) yang berhasil mencapai akurasi pengujian sebesar 100%, akurasi tersebut didapatkan dengan pengujian data dalam bentuk citra dari huruf komputer, yang berarti seluruh data dalam masing-masing kelas akan memiliki bentuk yang sama persis dengan *noise* yang tidak akan jauh berbeda. Bahkan data pelatihan, data validasi, dan data pengujian akan berisi data yang serupa, karena tidak memungkinkan adanya perubahan bentuk masing-masing alfabet bila data yang digunakan berasal dari huruf komputer. Selain itu, penelitian ini berfokus pada klasifikasi aksara Sunda dalam bentuk citra tulisan tangan digital.

Berdasarkan hasil penelitian ini, model yang diajukan dengan menggunakan CNN pada penelitian ini tergolong baik, karena mampu melampaui hampir seluruh penelitian-penelitian terdahulu yang juga menggunakan CNN dalam hal akurasi pengujian yang didapat secara subjektif menggunakan perhitungan *accuracy* dari *confusion matrix*. Angka akurasi yang dicapai membuktikan bahwa masing-masing tahap pada penelitian ini berhasil berjalan dengan baik. Baik hasil maupun proses dan masing-masing tahap yang dilakukan pada penelitian ini juga menunjukkan arah yang sejalan dengan penelitian-penelitian terdahulu. Penelitian ini juga memberikan *output* yang tidak disajikan oleh penelitian-penelitian terdahulu, yaitu terjemahan citra aksara Sunda ke dalam bentuk audio. Dengan menggunakan CNN, penelitian ini berhasil menerjemahkan citra aksara Sunda ke dalam bentuk audio dengan akurasi yang tinggi, yaitu 96% secara subjektif, sehingga ketepatan sistem dalam menerjemahkannya dapat diandalkan.

KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan, kesimpulan yang didapat adalah *teknik image processing* dapat digunakan untuk mengklasifikasi aksara Sunda maupun aksara lainnya dengan memberikan *output* berupa teks dan audio sesuai dengan tujuan penelitian yang menjawab permasalahan berupa belum adanya teknologi serupa yang mampu menghasilkan *output* berupa audio untuk memudahkan pengejaan dan pelafalan dalam pembelajaran aksara Sunda. Akurasi pelatihan dari model yang diajukan mencapai 99,63% dan akurasi validasi yang didapatkan mencapai 88,12%. Sementara itu akurasi pengujian dengan penginputan gambar secara otomatis menggunakan fungsi dari model yang diajukan (objektif) mencapai 82,14%, dan pengujian dengan penginputan gambar ke sistem secara manual (subjektif), akurasi pengujian berhasil mencapai 96%. Untuk menjawab permasalahan dari penelitian ini yang belum ada pada penelitian-penelitian sebelumnya, sistem yang diajukan mampu memberikan output berupa teks dan audio pelafalan atau pengejaan alfabet aksara Sunda sesuai dengan prediksi terhadap label klasifikasi dari alfabet aksara Sunda dalam bentuk gambar tulisan tangan digital yang diberikan, dimana penelitian-penelitian terdahulu belum dapat memberikan output berupa audio. Sistem juga dapat diatur untuk dapat memutar audio secara otomatis ataupun sebaliknya setelah melakukan prediksi terhadap label klasifikasi, sesuai dengan keinginan penggunaannya nanti.

Sebagai saran dan kontribusi dalam pembelajaran, penelitian ini memberikan peluang kepada penelitian selanjutnya dalam bidang klasifikasi menggunakan teknik *image processing* dengan algoritma CNN yang mampu memberikan *ouput* berupa audio. Selain itu, penelitian selanjutnya juga dapat mengembangkan penelitian ini untuk membaca aksara atau tulisan per kata atau bahkan per kalimat, baik aksara Sunda maupun aksara atau tulisan lainnya. Untuk kedepannya, penelitian selanjutnya dapat meningkatkan akurasi pengujian dalam melakukan klasifikasi citra tulisan tangan digital menggunakan CNN. Selain itu, model yang diajukan oleh penelitian ini beserta dengan *output*-nya dapat dikembangkan menjadi aplikasi berbasis *website* maupun *mobile* dengan beberapa modifikasi yang memudahkan pembelajaran kepada

masyarakat, baik masyarakat Indonesia maupun masyarakat asing yang ingin mempelajari aksara Sunda. Penambahan *dataset* dari berbagai sumber, seperti tulisan tangan yang ditangkap oleh kamera *handphone*, aksara Sunda dalam bentuk tulisan komputer, dan lainnya, serta penambahan aksara Rarangkén dapat dilakukan untuk memaksimalkan potensi dari sistem klasifikasi ini.

DAFTAR PUSTAKA

- Amalia, N., Hidayat, E. W., & Aldya, A. P. (2020). Pengenalan Aksara Sunda Menggunakan Metode Jaringan Saraf Tiruan Backpropagation Dan Deteksi Tepi Canny. *CESS (Journal of Computer Engineering, System and Science)*, 5(1), 19-27. <https://doi.org/10.24114/cess.v5i1.14839>
- Apriyanto, D. (2018). Implementasi Optical Character Recognition Pada Kamus Aksara Sunda-Indonesia Menggunakan Algoritma Feature Extraction Berbasis Android. *UIN Sunan Gunung Djati Bandung*. <https://digilib.uinsgd.ac.id/20803/>.
- Brownlee, J. (2019). *How to use Learning Curves to Diagnose Machine Learning Model Performance*. Machine Learning Mastery. <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>. (url)
- Chollet, F. (2021). *Deep Learning with Python*. USA: Simon and Schuster.
- El Naqa, I., & Murphy, M. J. (2015). What is machine learning?, in machine learning in radiation oncology. *machine learning in radiation oncology, Cham: Springer*, 3-11.
- Farhan, A. A., Atmaja, R. D., & Aulia, S. (2017). Perancangan Dan Analisis Sistem Pengenalan Kata Aksara Sunda Menggunakan Metode Learning Vector Quantization Berbasis Pengolahan Citra. *E-Proceeding of Engineering*, 1-14.
- Ferdiano, J. (2019). Alih Aksara Sunda Tulisan Tangan Menggunakan Metode Ekstraksi Ciri Freeman Chain Code (FCC) dan Metode Klasifikasi K-Nearest Neighbor(KNN). *Universitas Sanata Dharma (Skripsi)*. <https://repository.usd.ac.id/35194/>
- Gazali, W., Soeparno, H., & Ohliati, J. (2012). Penerapan Metode Konvolusi Dalam Pengolahan Citra Digital. *Jurnal Mat Stat*, 12(2), 103-113.
- Kahfie, M. I. (2022). Klasifikasi Citra Tulisan Tangan Aksara Sunda dengan Metode CNN-ELM. In *Universitas Gadjah Mada*. <http://etd.repository.ugm.ac.id/penelitian/detail/212378>. (skripsi)
- Kirana, A., Hikmayanti, H., & Indra, J. (2020). Pengenalan Pola Aksara Sunda dengan Metode Convolutional Neural Network. *Scientific Student Journal for Information, Technology and Science*, 1(2), 95-100.
- Maulana, A. (2013). Kurang, Kesadaran Masyarakat Sunda Menjaga Budaya Sunda. *Bandung: UNPAD*.
- Nielsen, M. A. (2015). *Neural Networks And Deep Learning*. San Francisco, CA, USA: Determination press.
- Nugraha, S. S., R, C. M. S., & Rachman, A. N. (2021). Implementasi Pengenalan Bahasa Sunda Bergambar Untuk Anak Usia Dini Berbasis Android. *Scientific Articles of Informatics Students*, 4(1), 25-34.

- Purnama, A., Bahri, S., Gunawan, G., Hidayatulloh, T., & Suhada, S. (2022). Implementation of Deep Learning for Handwriting Imagery of Sundanese Script Using Convolutional Neural Network Algorithm (CNN). *ILKOM Jurnal Ilmiah*, 14(1), 10-16. <https://doi.org/10.33096/ilkom.v14i1.989.10-16>.
- Rahmawati, S. N., Hidayat, E. W., & Mubarak, H. (2021). Implementasi Deep Learning Pada Pengenalan Aksara Sunda Menggunakan Metode Convolutional Neural Network. *INSERT: Information System and Emerging Technology Journal*, 2(1), 46-58. <https://doi.org/10.23887/insert.v2i1.37405>.
- Riansyah, R. R., Nurhasanah, Y. I., & Dewi, I. A. (2017). Sistem Pengenalan Aksara Sunda Menggunakan Metode Modified Direction Feature dan Learning Vector Quantization. *Jurnal Teknik Informatika dan Sistem Informasi*, 3(1), 17-30. <https://doi.org/10.28932/jutisi.v3i1.563>.
- Salsabila, H., Rachmawati, E., & Sthevanie, F. (2019). Sundanese Aksara Recognition Using Histogram of Oriented Gradients. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 253-258. <https://doi.org/10.1109/ISRITI48646.2019.9034589>.
- Septiadi, A. (2018). *Perbandingan algoritma template matching dengan canny untuk pengenalan aksara Sunda* (Doctoral dissertation, UIN Sunan Gunung Djati Bandung). <https://digilib.uinsgd.ac.id/14220/>
- Setyansyah, R., Siregar, Y. S., & Khairani, M. (2019). Noise Removal Pada Citra Digital Dengan Menggunakan Metode Active Contour. *ALGORITMA: Jurnal Ilmu Komputer Dan Informatika*, 5(1), 978-979.
- Shukla, N., & Fricklas, K. (2018). *Machine Learning with Tensor Flow*. Greenwich: Manning.
- Soleymani, A. (2022). *Your validation loss is lower than your training loss? This is why!* Towards Data Science. <https://towardsdatascience.com/what-your-validation-loss-is-lower-than-your-training-loss-this-is-why-5e92e0b1747e>.
- Sugiri, D., Hikmayanti, H., & Suharso, A. (2019). Rancang Bangun Aplikasi Kamus Sunda-Indonesia Dengan Metode Binary Search Berbasis Android. *Techno Xplore: Jurnal Ilmu Komputer Dan Teknologi Informasi*, 4(1), 1-14. <https://doi.org/10.36805/technoxplore.v4i1.537>.
- Taufiqurahman, M. (2019). Rekognisi Karakter Aksara Sunda Tulisan Tangan Menggunakan Ekstraksi Ciri HOG dan Metode Klasifikasi SVM. *Universitas Gadjah Mada*. <http://etd.repository.ugm.ac.id/penelitian/detail/183149>.
- Ungkawa, U., Fauzi, R., & Rohaeti, N. (2022). Penentuan Ukuran Citra Minimal Sistem Konversi Aksara Sunda dengan Metode Template Matching Correlation. *Multimedia Artificial Intelligent Networking Database Journal*, 7(2), 177-187. <https://doi.org/10.26760/mindjournal.v7i2.177-187>.
- Usman, S. (2020). *Terancam Punah dan Jalan Panjang Digitalisasi Aksara Daerah di Indonesia*. Merdeka.Com. <https://www.merdeka.com/teknologi/terancam-punah-dan-jalan-panjang-digitalisasi-aksara-daerah-di-indonesia.html>.
- Vaughan, J. (2018). *What Is Tensorflow?* Data Management Strategies Tech Target. <https://www.techtargget.com/searchdatamanagement/definition/TensorFlow>.

How to cite:

Nathanael, D., & Wasito, I. (2023). Model Klasifikasi Convolutional Neural Network Pada Sistem Penerjemah Audio Aksara Sunda. *DECODE: Jurnal Pendidikan Teknologi Informasi, 3(2)*, 322-340. <http://dx.doi.org/10.51454/decode.v3i2.217>