



Recognition of Letter Characters in Handwritten Images Using Convolutional Neural Network and K-Means Clustering Algorithm

Pengenalan Karakter Huruf pada Gambar Tulisan Tangan Menggunakan Algoritma Convolutional Neural Network dan K-Means Clustering

Natan Enggal Swasono¹, Agustinus Rudatyo Himamunanto^{*2}, Haeni Budiati³

^{1,2,3}Program Studi Informatika, Fakultas Sains dan Komputer
Universitas Kristen Immanuel, Indonesia

E-Mail: natanswasono@gmail.com¹,
rudatyo@ukrimuniversity.ac.id^{*2}, heni@ukrimuniversity.ac.id³

Received Jun 8th 2024; Revised Sept 25th 2024; Accepted Oct 12th 2024
Corresponding Author: Agustinus Rudatyo Himamunanto

Abstract

Handwritten character recognition is a major challenge in various digital applications due to variations in handwriting styles, character sizes, orientation, and image quality. This study aims to compare two primary methods for handwritten character recognition: Convolutional Neural Network (CNN) and K-Means Clustering. CNN is a supervised learning method used to extract important features from images, while K-Means Clustering is an unsupervised learning method that groups characters based on feature similarity. The results show that the CNN model achieved an accuracy of 90% in character recognition, demonstrating excellent performance in recognizing complex patterns and variations in handwriting. In contrast, K-Means Clustering only achieved an accuracy of 11%, indicating that this method is less effective for handwritten character recognition tasks. Based on these findings, it is concluded that CNN significantly outperforms K-Means Clustering in addressing the challenges of handwriting variation.

Keyword: Character Recognition, Convolutional Neural Network, K-Means Clustering, Handwriting, Digital Technology

Abstrak

Pengenalan karakter tulisan tangan adalah tantangan utama dalam berbagai aplikasi digital, disebabkan oleh variasi gaya tulisan, ukuran karakter, orientasi, dan kualitas gambar. Penelitian ini bertujuan membandingkan dua metode utama untuk pengenalan karakter tulisan tangan: Convolutional Neural Network (CNN) dan K-Means Clustering. CNN merupakan metode pembelajaran terarah yang digunakan untuk mengekstrak fitur penting dari gambar, sementara K-Means Clustering merupakan metode pembelajaran tanpa pengawasan yang mengelompokkan karakter berdasarkan kesamaan fitur. Hasil penelitian menunjukkan bahwa model CNN berhasil mencapai akurasi 90% dalam pengenalan karakter, menandakan kinerja yang sangat baik dalam mengenali pola kompleks dan variasi tulisan tangan. Sebaliknya, K-Means Clustering hanya mencapai akurasi 11%, menunjukkan bahwa metode ini kurang efektif untuk tugas pengenalan karakter tulisan tangan. Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa CNN jauh lebih unggul dibandingkan dengan K-Means Clustering dalam mengatasi tantangan variasi tulisan tangan.

Kata Kunci: Convolutional Neural Network, K-Means Clustering, Pengenalan Karakter, Teknologi Digital, Tulisan Tangan

1. PENDAHULUAN

Pengenalan karakter huruf pada gambar tulisan tangan adalah sebuah permasalahan yang relevan dan signifikan dalam berbagai aplikasi di era digital saat ini. Pengenalan karakter huruf adalah sebuah proses untuk mengenali tulisan tangan seseorang yang didapatkan dari sebuah citra digital kemudian sistem akan mengenali tulisan tersebut menggunakan metode klasifikasi.[1] Penerapan teknologi pengenalan karakter huruf ini dapat digunakan dalam berbagai konteks, termasuk pengenalan otomatis tulisan tangan, pengenalan tanda tangan digital, digitalisasi dokumen bersejarah, dan banyak lagi.

Tantangan utama dalam pengenalan karakter huruf pada gambar tulisan tangan adalah variasi yang tinggi dalam gaya tulisan tangan, ukuran karakter, orientasi, dan kejelasan gambar. Orang yang berbeda dapat menulis karakter yang sama dengan cara yang sangat berbeda, dan faktor-faktor seperti goresan tinta, kualitas kertas, dan penggunaan alat tulis yang berbeda juga berkontribusi pada kompleksitas masalah ini.

Untuk mengatasi tantangan-tantangan dalam pengenalan karakter huruf pada gambar tulisan tangan, teknologi yang digunakan harus mampu mengekstrak fitur-fitur yang relevan dan mengklasifikasikan karakter dengan akurasi tinggi. Dalam penelitian ini, dipilih dua metode, yaitu Convolutional Neural Network (CNN) dan K-Means Clustering, untuk dibandingkan kinerjanya. [2] CNN dipilih karena kemampuannya yang terbukti unggul dalam pengenalan pola pada gambar, khususnya dalam menangani variasi visual yang kompleks seperti perbedaan gaya tulisan, ukuran, dan orientasi karakter huruf. CNN juga memiliki kemampuan untuk mengekstrak fitur-fitur penting dan membangun representasi yang kuat, sehingga sangat cocok untuk tugas pengenalan karakter. Di sisi lain, K-Means Clustering dipilih karena kesederhanaannya serta efisiensinya dalam pemrosesan data tanpa label. K-Means Clustering mampu mengelompokkan karakter berdasarkan kesamaan fitur, meskipun tanpa pelatihan yang diawasi. Pemilihan kedua metode ini bertujuan untuk mengevaluasi pendekatan yang diawasi (CNN) dan tidak diawasi (K-Means) dalam konteks pengenalan karakter huruf pada gambar tulisan tangan.

Penelitian ini bertujuan untuk mengembangkan dan membandingkan kinerja dua pendekatan berbeda dalam pengenalan karakter huruf pada gambar tulisan tangan. CNN digunakan sebagai metode klasifikasi yang diawasi, di mana jaringan neural secara langsung belajar dari data yang diberi label untuk mengklasifikasikan karakter huruf secara akurat. Di sisi lain, K-Means digunakan sebagai metode clustering yang tidak diawasi, yang bertujuan untuk mengelompokkan karakter berdasarkan kemiripan fitur tanpa memerlukan label data selama pelatihan. Perbandingan ini tidak hanya untuk melihat tingkat akurasi klasifikasi kedua metode, namun juga untuk memahami batasan dari masing-masing pendekatan dalam pengenalan pola pada data tulisan tangan. Dengan demikian, meskipun CNN dan K-Means beroperasi dengan prinsip yang berbeda, analisis perbandingan ini diharapkan dapat memberikan wawasan tentang pendekatan yang lebih efektif dan efisien dalam menangani variasi karakter huruf pada gambar tulisan tangan.

Pengimplementasian metode CNN dan K-Means Clustering sudah banyak diteliti untuk pengenalan gambar maupun pengenalan karakter huruf pada tulisan tangan. Penelitian oleh [3] yang menggunakan metode CNN untuk identifikasi karakter Hiragana dapat menghasilkan akurasi ketepatan gambar sebesar 82% dan tidak perlu memiliki kumpulan data yang besar untuk mendapatkan hasil yang tinggi.

Penelitian yang dilakukan oleh [4] yang berjudul “Ekstraksi Ciri Bentuk Pada Aksara Jawa Kawi Menggunakan Metode L^*A^*B dan K-Means Clustering”. Penelitian ini membuat sistem ekstraksi bentuk pada aksara Jawa kawi yang ngan memproses selanjutnya dapat digunakan untuk mengklasifikasi citra aksara Jawa kawi agar dapat digunakan untuk proses pembacaan aksara Jawa kawi. Pengujian dilakukan deses 6 citra aksara menggunakan metode K-Means Clustering yang akan menghasilkan nilai segmentasi selanjutnya diambil nilai ciri bentuk meliputi Area, Perimeter, Metric dan Eccentricity selanjutnya dapat diproses menggunakan metode Jaringan Syaraf Tiruan untuk klasifikasinya.

Menurut [5] dalam penelitian “Handwritten Character Recognition Using Convolutional Neural Network”, ditemukan bahwa akurasi yang diperoleh dari 200 gambar latih sebesar 65,32% meningkat secara bertahap seiring dengan bertambahnya gambar latih. Akurasinya mencapai 92,91% dengan 1000 gambar latih, hal ini menunjukkan bahwa jumlah gambar latih menentukan akurasi dari CNN. Kemudian dari penelitian oleh [6] yang berjudul “Convolutional Neural Networks for Handwritten Javanese Character Recognition”, mengatakan bahwa akurasi model CNN lebih baik daripada akurasi model Multi Layer Perceptron (MLP) untuk tugas pengenalan karakter Jawa tulisan tangan. Namun model CNN membutuhkan waktu pelatihan yang lebih lama.

Dengan demikian, penelitian ini tidak hanya akan mengembangkan sistem pengenalan karakter huruf, tetapi juga menyediakan analisis komprehensif mengenai kinerja CNN dan K-Means Clustering, yang akan memiliki dampak positif dalam berbagai aplikasi, termasuk pengolahan dokumen, keamanan, dan banyak lagi.

2. METODE PENELITIAN

Untuk membuat sistem program ada beberapa tahapan dan metode dalam penelitian ini. Metodologi yang diuraikan di sini mencakup langkah-langkah yang diperlukan dalam pengumpulan data, mengembangkan program metode CNN dan program dengan K-Means Clustering. Diharapkan bahwa metodologi ini akan memberikan kerangka kerja yang kokoh dan terperinci untuk mencapai tujuan penelitian dengan efektif dan efisien.

2.1. Tahapan Proses

Dalam penelitian ini, tahapan proses dimulai dengan pemilihan dataset yang sesuai untuk pengenalan karakter huruf pada gambar tulisan tangan. Pemilihan dataset yang tepat merupakan kunci untuk memastikan ketersediaan data tentang variasi gaya tulisan tangan yang cukup. Untuk menguji dan melatih model pengenalan karakter huruf tulisan tangan, penelitian ini menggunakan dataset English Handwritten Characters

yang dibuat oleh Dhruvil Dave. Selanjutnya, proses preprocessing data dilakukan untuk membersihkan dan menormalkan gambar-gambar dari dataset, termasuk resizing, normalisasi intensitas piksel, dan segmentasi karakter huruf [7]. Langkah preprocessing yang teliti diperlukan untuk mempersiapkan data yang sesuai untuk pelatihan model CNN dan K-Means Clustering.

Setelah data siap, model CNN akan dilatih menggunakan dataset yang telah diproses untuk mengekstrak fitur-fitur penting dari gambar tulisan tangan. Proses dari pelatihan ini melibatkan operasi konvolusi, ReLU, dan Pooling. Operasi konvolusi pada gambar I dengan filter F menghasilkan output fitur G , yang dirumuskan pada persamaan 1.

$$G[i, j] = (I * F)[i, j] = \sum_m \sum_n I[i + m, j + n] \cdot F[m, n] \quad (1)$$

Kemudian, fungsi aktivasi ReLU digunakan untuk memperkenalkan non-linearitas, dengan menggunakan persamaan 2.

$$f(x) = \max(0, x) \quad (2)$$

Setelahnya, lapisan pooling diterapkan untuk mengurangi dimensi fitur map, contohnya max pooling dengan menggunakan persamaan 3.

$$G[i, j] = \max_{m, n}(I[i + m, j + n]) \quad (3)$$

Proses dari pelatihan ini memerlukan iterasi yang cermat untuk mengoptimalkan kinerja model. Selanjutnya, menguji model CNN dan mengevaluasi hasil model berdasarkan tingkat akurasi. Setelah model CNN sudah siap, masuk ke metode K-Means Clustering. Dimulai dengan menerapkan Principal Component Analysis (PCA) untuk reduksi dimensi. Pertama data dikurangkan dengan rata-ratanya (mean centering), ditunjukkan pada persamaan 4.

$$X_{centered} = X - \underline{X} \quad (4)$$

Kemudian, matriks kovarian dihitung:

$$\Sigma = \frac{1}{n-1} X_{centered}^T X_{centered} \quad (5)$$

Selanjutnya, dekomposisi eigen dilakukan pada matriks kovarian:

$$\Sigma_v = \lambda_v \quad (6)$$

Data kemudian diproyeksikan ke ruang baru yang dibentuk oleh eigenvectors:

$$Z = X_{centered} V \quad (7)$$

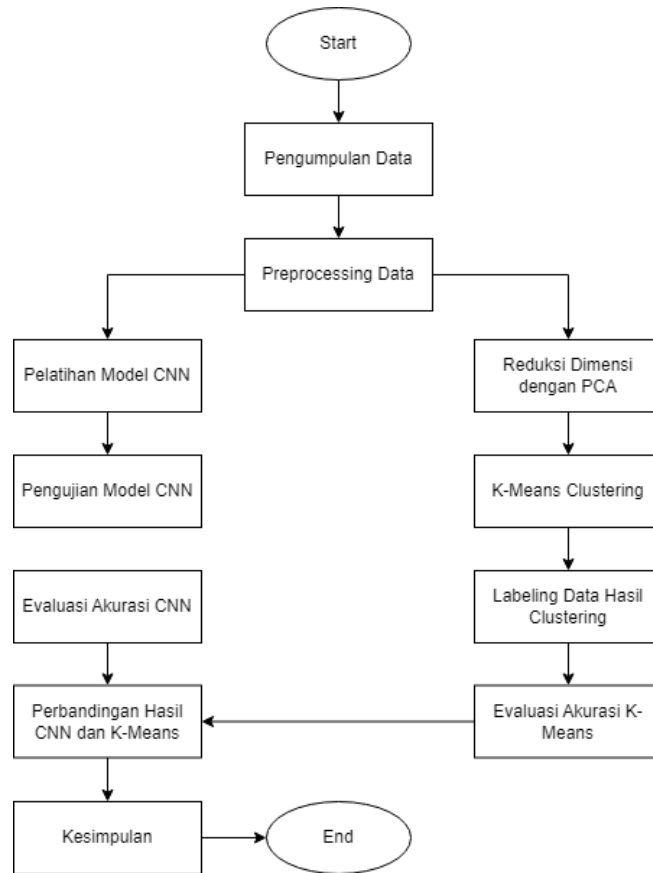
Setelah reduksi dimensi, K-Means Clustering diterapkan pada data yang telah direduksi. K-Means dimulai dengan inialisasi centroid, kemudian setiap data ditetapkan ke centroid terdekat menggunakan jarak Euclidean, menggunakan persamaan 8.

$$d(x, c) = \sqrt{\sum_{i=1}^n (x_i - c_i)^2} \quad (8)$$

Posisi centroid diperbarui sebagai rata-rata dari semua titik data dalam cluster tersebut, ditunjukkan pada persamaan 9.

$$c_j = \frac{1}{C_j} \sum_{x \in C_j} x \quad (9)$$

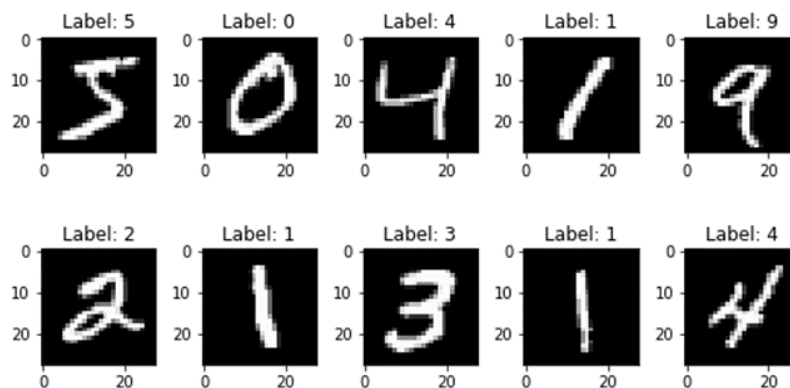
Langkah ini diulang hingga konvergensi tercapai. Setelahnya, data hasil clustering diberi label berdasarkan mayoritas label asli dalam cluster tersebut, dan akurasi K-Means dihitung dengan membandingkan label asli dan label prediksi. Terakhir, evaluasi dan perbandingan hasil akurasi CNN dan K-Means Clustering dilakukan. Evaluasi kelebihan dan kekurangan masing-masing metode berdasarkan hasil pengujian. Metodologi penelitian ditunjukkan pada gambar 1.



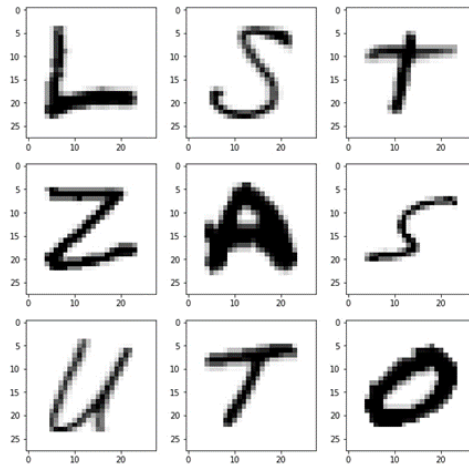
Gambar 1. Flowchart rancangan proses penelitian

2.2. Dataset Gambar Tulisan Tangan

Model data ini terdiri dari gambar tulisan tangan mencakup karakter huruf dalam banyak gaya tulisan. Dataset ini adalah komponen utama dari model data ini. Dataset mencakup karakter tulisan 26 huruf besar (A-Z) dan 26 huruf kecil (a-z), bersama dengan angka (0-9) dan berbagai tanda baca. Setiap gambar dalam dataset harus diberi label yang sesuai dengan karakter huruf yang terdapat dalam gambar. Label ini diperlukan untuk melatih model CNN.



Gambar 2. Contoh dataset gambar tulisan tangan angka



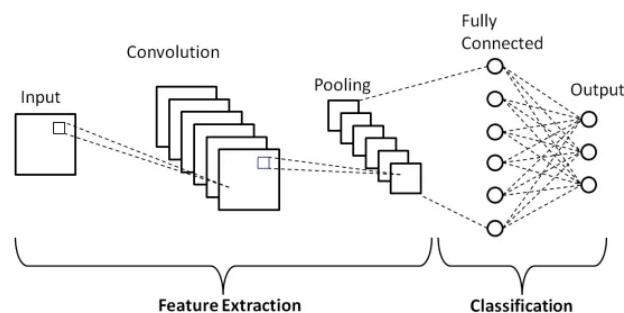
Gambar 3. Contoh dataset gambar tulisan tangan karakter huruf

Dataset akan dibagi menjadi dua bagian, data pelatihan (training data) dan data pengujian (testing data). [8] Training data kan digunakan dalam pelatihan model CNN, sedangkan testing data akan digunakan untuk menguji kinerja dari model.

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan yang dirancang khusus untuk pengolahan gambar dan data yang berstruktur grid, seperti gambar dan video [9]. CNN merupakan model baru di bidang pengenalan objek. Dikhususkan untuk input data yang bertipe spatial, CNN memiliki layer khusus, yaitu layer konvolusi dan layer pooling yang memungkinkan proses pembelajaran fitur secara hierarki dari data.

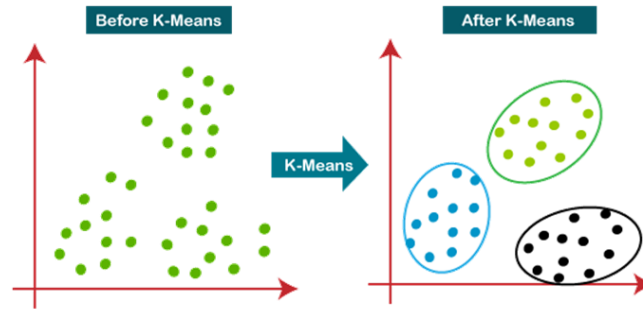
Menurut [10] CNN is an efficient feature extractor, atau CNN adalah ekstraktor fitur yang efisien. Karena CNN mempunyai kemampuan untuk secara otomatis mengidentifikasi fitur-fitur dalam gambar, sehingga CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image yang membuatnya sangat efektif dalam tugas pengenalan gambar, klasifikasi, deteksi objek, segmentasi, dan banyak lagi [11]. CNN adalah alat yang sangat berguna dalam pengolahan gambar dan pengenalan pola, dan telah memungkinkan kemajuan signifikan dalam berbagai bidang. Arsitektur seperti AlexNet, VGG, ResNet, dan Inception adalah beberapa contoh arsitektur CNN yang telah berhasil dalam berbagai tugas pengenalan gambar.



Gambar 4. Contoh CNN

2.4. K-Means Clustering

K-Means Clustering adalah sebuah algoritma yang digunakan untuk mengelompokkan data ke dalam kelompok-kelompok (cluster) yang mirip berdasarkan kemiripan karakteristik. Menurut penelitian [12] tujuan dari clustering adalah untuk mengklasifikasikan data, dengan cara menentukan pengelompokan dalam satu set data yang tidak diketahui. K-Means merupakan algoritma untuk cluster n objek berdasarkan atribut menjadi k partisi, dimana $k < n$. [13] Dalam konteks pengenalan gambar tulisan tangan, K-Means Clustering dapat digunakan untuk pemisahan karakter huruf yang berdekatan dalam gambar. Algoritma K-means menggunakan proses secara berulang-ulang untuk mendapatkan basis data cluster [14].

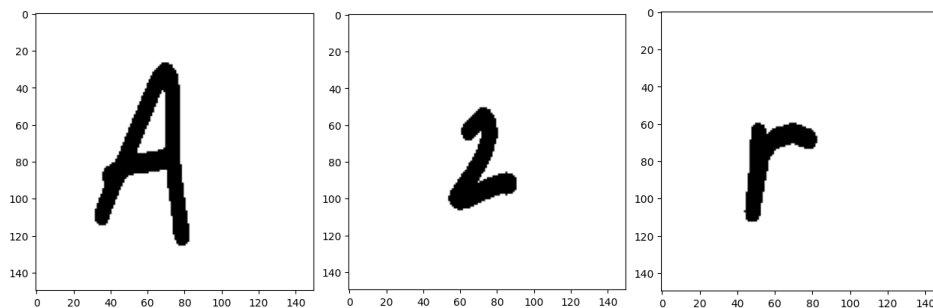


Gambar 5. Contoh K-Means Clustering

3. HASIL DAN PEMBAHASAN

Pada tahap ini akan membahas tentang hasil dan pembahasan dari pengembangan model machine learning yang telah dilakukan, dari proses dan serangkaian pengujian untuk mengevaluasi kinerjanya secara obyektif. Pengujian dilakukan menggunakan data yang terpisah dari data yang digunakan dalam pelatihan model, dengan tujuan untuk mengukur kemampuan model dalam melakukan generalisasi pada data baru yang belum pernah dilihat sebelumnya. Melalui tahap ini, diharapkan dapat memberikan pemahaman yang lebih dalam tentang kinerja model yang dikembangkan, serta memberikan wawasan yang berharga untuk pengembangan dan peningkatan model di masa depan.

Tahapan pengumpulan dataset karakter huruf tulisan tangan memainkan peran penting dalam menentukan kualitas dan generalisasi model. Dataset diperoleh dari internet di kaggle.com. Dimana terkumpul 3410 data gambar dengan 62 class A-Z, a-z, 0-9, dan tiap karakter terdapat 55 gambar tulisan tangan. Setelah dataset terkumpul, kemudian dataset dibagi menjadi data pelatihan dan data validasi dengan 70% untuk data pelatihan dan 30% untuk data validasi. Setelah pembagian dataset selesai, dilakukan tahap preprocessing. Dimana data yang dimiliki perlu diubah dengan ukuran yang sama lebih mudah. Ukuran gambar diubah menjadi 150 x 150 pixels. Setelah diubah ukuran gambarnya selanjutnya dilakukan labeling sesuai dengan karakter menjadi 62 class. Setelah melewati tahapan-tahapan sebelumnya data siap digunakan dalam pelatihan model.



Gambar 6. Contoh Data Gambar Setelah di Resize

3.1. Implementasi Convolutional Neural Network (CNN)

Proses selanjutnya adalah proses untuk mendefinisikan arsitektur model CNN menggunakan Sequential API dari TensorFlow Keras. Tujuannya adalah untuk membangun sebuah model CNN yang mampu melakukan klasifikasi gambar berdasarkan fitur-fitur yang diekstraksi dari gambar input.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(150, 150, 3),
padding='same'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(256, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(62, activation='softmax')
])
```

Pada tahap ini Convolutional Layers bertujuan untuk mengekstraksi fitur-fitur penting dari gambar input. Pada kode tersebut, digunakan beberapa layer konvolusi dengan jumlah filter yang bertambah secara bertahap (16, 32, 64, 128, 256). Setiap filter akan mempelajari pola-pola yang berbeda dari gambar input. MaxPooling Layers bertujuan untuk mengurangi dimensi dari fitur-fitur yang telah diekstraksi oleh layer konvolusi, sehingga mengurangi kompleksitas model dan mempercepat proses training.

Flatten Layer ditambahkan untuk mengubah output dari layer konvolusi menjadi vektor satu dimensi, sehingga dapat menjadi input untuk layer-layer Dense selanjutnya. Ditambahkan juga Dense Layers dengan 512 unit atau fully connected dan fungsi aktivasi ReLU yang bertujuan untuk melakukan klasifikasi berdasarkan fitur-fitur yang telah diekstraksi sebelumnya. Pada kode tersebut, terdapat dua layer dense terakhir. Layer dense terakhir memiliki jumlah unit yang sama dengan jumlah kelas yang akan diprediksi, dengan fungsi aktivasi softmax untuk menghasilkan distribusi probabilitas untuk setiap kelas.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(150, 150, 3),
padding='same'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(256, (3,3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(62, activation='softmax')
])
```

Setelah melalui serangkaian tahapan sebelumnya, model CNN yang telah dirancang kemudian diuji untuk melihat tingkat keakuratannya dengan menggunakan adam optimizer. Dengan melakukan 10 iterasi epoch, hasil evaluasi menunjukkan tingkat akurasi dan loss yang dapat memberikan gambaran tentang performa model.

```
accuracy = history.history['accuracy'][-1] * 100
print(f'Akurasi model: {accuracy:.2f}%')
```

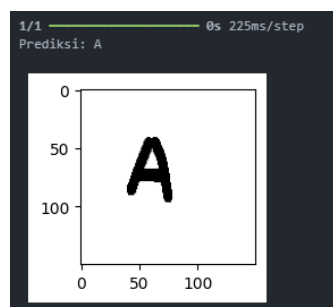
✓ 1.8s

Akurasi model: 90.62%

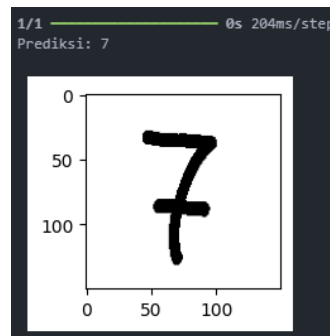
Gambar 7. Akurasi Model CNN

Hasilnya menunjukkan bahwa model berhasil mencapai tingkat akurasi 90.62% pada data pelatihan. Meskipun masih terdapat ruang untuk peningkatan, tingkat akurasi ini menunjukkan bahwa model sudah cukup baik dalam mempelajari pola-pola yang ada dalam data pelatihan. Hasil tersebut menunjukkan bahwa model mampu secara umum untuk menggeneralisasi pola-pola yang dipelajari dari data pelatihan ke data baru yang belum pernah dilihat sebelumnya.

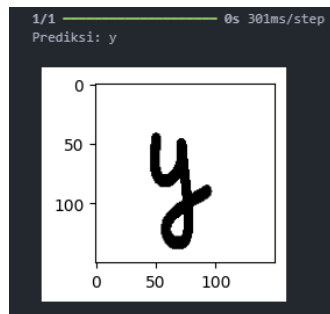
Langkah berikutnya dalam proses ini adalah menguji model CNN yang telah dikembangkan dengan mengunggah gambar tulisan tangan ke dalam sistem untuk melihat hasil prediksi. Data masukan diambil dari file yang sebelumnya telah disiapkan dan dilatih. Hasil output akan menampilkan gambar beserta prediksinya, hasil sesuai seperti gambar 8.



Gambar 8. Hasil Prediksi Unggah Gambar ke Sistem CNN



Gambar 9. Hasil Prediksi Unggah Gambar ke Sistem CNN



Gambar 10. Hasil Prediksi Unggah Gambar ke Sistem CNN

3.2. *K-Means Clustering*

Proses ini bertujuan untuk mengimplementasikan metode K-Means Clustering sebagai metode alternatif untuk pengenalan karakter huruf pada gambar tulisan tangan. K-Means Clustering adalah metode pembelajaran tanpa pengawasan yang mengelompokkan data ke dalam sejumlah k cluster berdasarkan kesamaan fitur. Berikut adalah langkah-langkah yang dilakukan dalam penerapan K-Means Clustering:

Proses selanjutnya adalah reduksi dimensi dengan PCA, dimana PCA bertujuan untuk mengurangi dimensi fitur dan menghilangkan noise. Principal Component Analysis (PCA) mengurangi fitur menjadi sejumlah komponen utama yang paling signifikan, mempercepat proses clustering.

```
pca = PCA(n_components=50)
data_pca = pca.fit_transform(data_scaled)
```

Kode di atas membuat objek PCA (Principal Component Analysis) yang akan mereduksi data ke 50 komponen utama. PCA adalah teknik untuk mengurangi dimensi data sambil mempertahankan sebanyak mungkin variasi di dalam data. Kemudian menerapkan PCA pada data yang sudah diskalakan. `fit_transform` pertama-tama menyesuaikan PCA pada data (menghitung komponen utama) dan kemudian mengubah data ke ruang komponen utama yang baru. Hasilnya adalah data yang direduksi menjadi 50 dimensi.

Selanjutnya adalah penerapan clustering dengan K-Means, fitur yang telah direduksi dimensi kemudian dikelompokkan menggunakan K-Means Clustering. Jumlah cluster k ditentukan berdasarkan jumlah label unik dalam dataset, yaitu 62.

```
k = len(np.unique(labels))
kmeans = KMeans(n_clusters=k, random_state=42)
cluster_labels = kmeans.fit_predict(features_pca)
```

Kode diatas membuat objek K-Means dari scikit-learn dengan parameter `n_clusters=k`, di mana k adalah jumlah klaster yang telah ditentukan sebelumnya. Kemudian `random_state=42` adalah parameter `random_state` diatur untuk memastikan bahwa hasil clustering dapat direproduksi. Nilai 42 adalah seed untuk generator angka acak yang digunakan dalam K-Means. Bisa menggunakan angka lain, tetapi nilai 42 sering digunakan sebagai konvensi dalam penelitian sebelumnya.

Kemudian metode `'fit_predict'` melakukan dua hal, yaitu `'fit'`, melatih (menyesuaikan) model K-Means pada data `features_pca`. Ini berarti K-Means akan menemukan k klaster dalam data. Lalu `'predict'`, memprediksi klaster untuk setiap data poin dalam `features_pca`. Hasilnya adalah array `cluster_labels` yang berisi nomor klaster (0 hingga k-1) untuk setiap data poin. Langkah selanjutnya adalah Mapping Cluster ke Label, setiap cluster diberi label berdasarkan mayoritas label asli dalam cluster tersebut. Ini dilakukan untuk memetakan hasil cluster ke label yang sebenarnya.


```
cluster_to_labels = map_clusters_to_labels(cluster_labels, labels)
predicted_labels = np.array([cluster_to_labels[cluster] for cluster in cluster_labels])
```

Kode tersebut memetakan hasil clustering K-Means ke label asli dari dataset. Pertama, `map_clusters_to_labels` membuat dictionary `cluster_to_labels` yang memetakan setiap cluster ke label mayoritas dalam cluster tersebut berdasarkan `cluster_labels` dan `labels`. Kemudian, menggunakan dictionary ini, kode `predicted_labels = np.array([cluster_to_labels[cluster] for cluster in cluster_labels])` mengonversi label cluster yang dihasilkan oleh K-Means menjadi label asli yang diprediksi untuk setiap data point, memungkinkan evaluasi akurasi hasil clustering terhadap label asli. Untuk proses selanjutnya adalah mengukur seberapa mirip titik data dalam cluster dibandingkan dengan titik data di cluster lain. Nilai Silhouette Score yang lebih tinggi menunjukkan pemisahan yang lebih baik antar cluster.

```
silhouette_avg = silhouette_score(features_pca, kmeans.labels_)
print("Silhouette Score:", silhouette_avg )
```

Silhouette Score: 0.046357278

Gambar 11. Silhouette Score

Kode di atas menghitung Silhouette Score menggunakan fitur yang telah direduksi dimensinya (`features_pca`) dan label cluster yang dihasilkan oleh K-Means (`kmeans.labels_`). Silhouette Score mengukur seberapa baik setiap titik dalam satu cluster mirip dengan titik-titik dalam cluster yang sama dibandingkan dengan titik-titik di cluster yang berbeda. Nilai Silhouette Score berkisar antara -1 dan 1, dengan nilai yang lebih tinggi menunjukkan bahwa titik data lebih dekat ke cluster mereka dan jauh dari cluster lainnya.

Hasil yang didapatkan adalah Silhouette Score: 0.046357278, yang menunjukkan bahwa kualitas clustering yang dihasilkan oleh K-Means pada dataset ini rendah, karena nilai Silhouette Score mendekati 0. Ini menunjukkan bahwa cluster yang dihasilkan tidak cukup terpisah dengan baik atau bahwa data dalam cluster tidak dikelompokkan dengan baik.

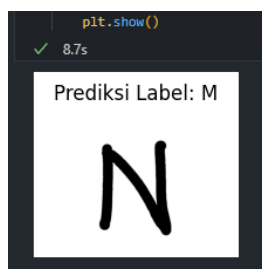
```
# Evaluate clustering accuracy
predicted_labels = np.array([cluster_to_labels[cluster] for cluster in cluster_labels])
accuracy = accuracy_score(labels_numeric, predicted_labels)
print(f"Akurasi K-Means clustering: {accuracy * 100:.2f}%")
```

Akurasi K-Means clustering: 11.03%

Gambar 12. Tingkat Akurasi K-Means Clustering

Selanjutnya adalah menghitung akurasi dengan membandingkan label asli dengan label prediksi. Akurasi ini dicetak dalam persentase. Hasil yang diperoleh adalah Akurasi K-Means clustering: 11.03%, yang menunjukkan bahwa model K-Means clustering hanya berhasil mengelompokkan sekitar 11.03% dari data dengan benar. Ini adalah hasil yang rendah, yang menunjukkan bahwa metode clustering K-Means tidak berhasil dengan baik dalam mengenali dan mengelompokkan karakter huruf dalam dataset ini.

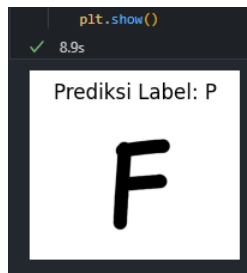
Langkah berikutnya dalam proses ini adalah menguji model K-Means Clustering yang telah dikembangkan dengan mengunggah gambar tulisan tangan ke dalam sistem untuk melihat hasil prediksi. Data masukan diambil dari file yang sebelumnya telah disiapkan. Hasil output akan menampilkan gambar beserta prediksinya, hasil sesuai seperti gambar 13-16.



Gambar 13. Hasil Prediksi Unggah Gambar ke K-Means Clustering



Gambar 14. Hasil Prediksi Unggah Gambar ke K-Means Clustering



Gambar 15. Hasil Prediksi Unggah Gambar ke K-Means Clustering



Gambar 16. Hasil Prediksi Unggah Gambar ke K-Means Clustering

Dari contoh hasil prediksi dari gambar yang diupload ke sistem dapat dilihat bahwa sebagian besar hasil prediksi menunjukkan bahwa model K-Means sering kali tidak mampu memberikan prediksi yang akurat. Terlebih lagi model kesulitan membedakan bentuk huruf yang hampir mirip misalnya, gambar 'N' diprediksi sebagai 'M', gambar '6' diprediksi sebagai 'G', dan gambar 'F' diprediksi sebagai 'P'.

4. KESIMPULAN

Dalam penelitian ini, telah dibandingkan dua metode yang berbeda untuk pengenalan karakter huruf pada gambar tulisan tangan: Convolutional Neural Network (CNN) sebagai metode pembelajaran terawasi dan K-Means Clustering sebagai metode pembelajaran tanpa pengawasan. Berdasarkan hasil penelitian, dapat disimpulkan bahwa model CNN menunjukkan kinerja yang sangat baik dalam pengenalan karakter huruf, dengan tingkat akurasi mencapai 90.62%. Hal ini menunjukkan kemampuan CNN dalam mengekstraksi fitur-fitur kompleks dari gambar dan melakukan klasifikasi dengan akurasi tinggi. Di sisi lain, model K-Means Clustering hanya mencapai tingkat akurasi sebesar 11.03%, yang menunjukkan bahwa metode ini kurang efektif untuk tugas pengenalan karakter huruf pada gambar tulisan tangan.

Secara keseluruhan, dapat disimpulkan bahwa CNN memiliki kinerja yang jauh lebih unggul dibandingkan dengan K-Means Clustering untuk tugas pengenalan karakter huruf. CNN terbukti mampu menangkap fitur-fitur kompleks dan informasi spasial dari gambar tulisan tangan dengan lebih baik, yang membuatnya sangat efektif untuk tugas ini. Di sisi lain, K-Means Clustering tidak mampu menangkap pola-pola kompleks dalam data, sehingga kurang cocok untuk digunakan dalam pengenalan karakter huruf pada gambar tulisan tangan.

REFERENSI

- [1] N. K. Qudsi, R. A. Asmara, and A. R. Syulistyo, "Identifikasi Citra Tulisan Tangan Digital Menggunakan Convolutional Neural Network (CNN)," *Semin. Inform. Apl. Polinema*, pp. 48–53, 2019.
- [2] A. Willyanto, D. Alamsyah, and H. Irsyad, "Identifikasi Tulisan Tangan Aksara Jepang Hiragana Menggunakan Metode CNN Arsitektur VGG-16," *J. Algoritm.*, vol. 2, no. 1, pp. 1–11, 2021.
- [3] C. Umam and L. Budi Handoko, "Convolutional Neural Network (CNN) Untuk Identifikasi Karakter Hiragana," *Pros. Semin. Nas. Lppm Ump*, vol. 0, no. 0, pp. 527–533, 2020, [Online]. Available: <https://seminaslppm.ump.ac.id/index.php/seminaslppm/article/view/199>
- [4] A. I. Maulana, C. I. Isniawan, and M. I. Y. Mustofa, "Identifikasi Kepribadian Dari Tulisan Tangan Menggunakan Euclidean Distance," vol. 2, pp. 177–182, 2023.
- [5] I. Khandokar, M. Hasan, F. Ernawan, S. Islam, and M. N. Kabir, "Handwritten character recognition using convolutional neural network," *J. Phys. Conf. Ser.*, vol. 1918, no. 4, 2021, doi: 10.1088/1742-6596/1918/4/042152.
- [6] C. K. Dewa, A. L. Fadhilah, and A. Afiahayati, "Convolutional Neural Networks for Handwritten Javanese Character Recognition," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 12, no. 1, p. 83, 2018, doi: 10.22146/ijccs.31144.
- [7] A. Bwatiramba and S. Venkataraman, "Handwritten Character Recognition Using Deep Learning (Convolutional Neural Network)," *Comput. Eng. Intell. Syst.*, no. March, 2023, doi: 10.7176/ceis/14-1-05.
- [8] J. Briliantio, N. Santosa, G. Ardian, and L. Hakim, "Penerapan Convolutional Neural Network untuk Handwriting Recognition pada Aplikasi Belajar Aritmatika Dasar Berbasis Web," *J. Tek. Inform. Unika St. Thomas*, vol. 05, no. 02, pp. 137–146, 2020.
- [9] S. Prihatiningsih, N. S. M. F. Andriani, and N. Nugraha, "Analisa Performa Pengenalan Tulisan Tangan Angka Berdasarkan Jumlah Iterasi Menggunakan Metode Convolutional Neural Network," *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 1, pp. 58–66, 2019, doi: 10.35760/tr.2019.v24i1.1934.
- [10] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, "Handwritten Character Recognition from Images using CNN-ECOC," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 2403–2409, 2020, doi: 10.1016/j.procs.2020.03.293.

-
- [11] M. T. Stefanus Christian Adi Pradhana, Untari Novia Wisesty S.T.,M.T., Febryanthi Sthevanie S.T., “Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network,” *e-Proceeding Eng.*, vol. 7, no. 1, pp. 2558–2567, 2020.
- [12] N. Wakhidah *et al.*, “Penerapan Data Mining K-Means Clustering Untuk,” *Transmisi*, vol. 4, no. 4, pp. 241–249, 2021.
- [13] N. Wakhidah, “Clustering Menggunakan K-Means Algorithm (K-Means Algorithm Clustering),” *Fak. Teknol. Inf.*, vol. 21, no. 1, pp. 70–80, 2014.
- [14] E. Fernando Ade Pratama, K. Khairil, and J. Jumadi, “Implementasi Metode K-Means Clustering Pada Segmentasi Citra Digital,” *J. Media Infotama*, vol. 18, no. 2, pp. 291–301, 2022.
- [15] R. A. C, S. Dev Nalluri, S. Manoj Thota, and A. Thota Asst Professor, “International Journal of Innovative Research in Computer and Communication Engineering Handwritten Digit Recognition Using CNN,” vol. 11, no. 1, pp. 8849–8858, 2023, doi: 10.15680/IJIRCCE.2023.11010012.
- [16] P. Guruprasad and D. J. Majumdar, “Optimal Clustering Technique for Handwritten Nandinagari Character Recognition,” *Int. J. Comput. Appl. Technol. Res.*, vol. 6, no. 5, pp. 213–223, 2017, doi: 10.7753/ijcatr0605.1001.
- [17] Y. Sugianela and N. Suciati, “Javanese Document Image Recognition Using Multiclass Support Vector Machine,” *CommIT J.*, vol. 13, no. 1, pp. 25–30, 2019, doi: 10.21512/commit.v13i1.5330.
- [18] A. Agustina, S. Suwarno, and U. Proboyekti, “Pengenalan Aksara Jawamenggunakan Learning Vector Quantization (Lvq),” *Inform. J. Teknol. Komput. dan Inform.*, no. 1, 2011.
- [19] K. P. Sinaga and M. Yang, “Unsupervised K-Means Clustering Algorithm,” vol. 8, 2020, doi: 10.1109/ACCESS.2020.2988796.
- [20] M. H. Faishal, M. D. Sulistiyo, A. F. Ihsan, A. Info, F. R-cnn, and J. Script, “Javanese Script Letter Detection Using Faster R-CNN,” vol. 6, no. 2, pp. 243–251, 2023.