



PERANCANGAN APLIKASI PENGENALAN AKSARA JAWA DIGITAL MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DAN COMPUTER VISION

Alvin Jonathan^{1)*}, Ito Wasito²⁾

¹Universitas Pradita, Tangerang, Indonesia

Email: alvin.jonathan@student.pradita.ac.id

Abstrak

Aksara Jawa merupakan salah satu aksara tertua yang terkenal dan sering digunakan oleh masyarakat pulau Jawa untuk penulisan sehari-harinya. Namun pada zaman sekarang penggunaan aksara Jawa sudah jarang digunakan oleh masyarakat dan ketertarikan para generasi muda untuk mempelajari budaya Jawa yang berupa aksara ini sangat minim. Pada penelitian ini akan membahas tentang penggunaan teknologi *Deep Learning* dan *Computer Vision* untuk pembelajaran aksara Jawa dan sebagai sarana dalam pelestarian budaya Jawa. Penulis menggunakan metode studi literatur dengan mencari dan membaca dari penelitian-penelitian terdahulu untuk menemukan keterbaruan dalam penelitian. Pada penelitian ini akan menggunakan teknologi *Deep Learning*, *Computer Vision*, dan algoritma *Deep Learning* yaitu *Convolutional Neural Network* untuk mengklasifikasikan gambar aksara Jawa tulisan tangan digital. Tujuan dari dilakukannya penelitian ini yaitu untuk menguji akurasi dari penggunaan teknologi *Deep Learning* dan *Computer Vision* dalam pengenalan dan klasifikasi gambar aksara Jawa tulisan tangan digital berdasarkan kelompoknya. Dari penelitian yang telah dilakukan didapatkan akurasi yang cukup memuaskan dari penggunaan algoritma *Convolutional Neural Network* sebesar 84% dalam pengenalan gambar aksara Jawa tulisan tangan digital.

Kata kunci: aksara jawa; computer vision; convolutional neural network; deep learning; tulisan tangan digital.

DESIGN OF DIGITAL JAVANESE SCRIPT RECOGNITION APPLICATION USING CONVOLUTIONAL NEURAL NETWORK AND COMPUTER VISION

Abstract

The Javanese script is one of the oldest well-known scripts and is often used by the people of the island of Java for their daily writing. However, nowadays the use of Javanese script is rarely used by the community and the interest of the younger generation in learning Javanese culture in the form of this script is very minimal. This research will discuss the use of Deep Learning and Computer Vision technology for learning Javanese script and as a means of preserving Javanese culture. The author uses the literature study method by searching and reading from previous studies to find updates in research. This research will use Deep Learning technology, Computer Vision, and Deep Learning algorithms, namely the Convolutional Neural Network to classify images of digital handwritten Javanese script. The purpose of this research is to test the accuracy of the use of Deep Learning and Computer Vision technologies in the recognition and classification of digital handwritten Javanese script images based on their groups. From the research that has been done, it is obtained that the accuracy is quite satisfactory from the use of the Convolutional Neural Network algorithm of 84% in recognizing images of digital handwritten Javanese characters.

Keywords: *javanese script; computer vision; convolutional neural network; deep learning; digital handwriting.*

Submitted: 24 Mei 2023

Reviewed: 24 Mei 2023

Accepted: 12 Juli 2023

Published: 13 Agustus 2023

PENDAHULUAN

Aksara merupakan sebuah sistem penulisan suatu bahasa yang memiliki bentuk visual, aksara biasanya ditulis diatas sebuah media seperti kertas, batu, kayu, kain, dan media tulis lainnya. Terdapat berbagai jenis aksara di dunia seperti aksara Latin, aksara Arab, dan aksara lainnya. Salah satu negara kepulauan terbesar di dunia adalah Indonesia, sehingga di Indonesia terdapat budaya yang beraneka ragam seperti budaya berupa aksara, di Indonesia terdapat berbagai macam aksara seperti aksara Jawa, aksara Makassar, aksara Batak, aksara Sasak, dan aksara lainnya. Namun pada penelitian ini akan berfokus pada aksara Jawa, aksara Jawa merupakan sebuah warisan budaya dari suku Jawa yang digunakan sebagai sarana penulisan pada zaman dahulu. Suku Jawa merupakan suku dengan populasi terbesar di Indonesia, berdasarkan keterangan dari Badan Pusat Statistik pada tahun 2010 total populasi suku Jawa berjumlah 95.217.022 jiwa (Putri, 2023). Aksara Jawa juga memiliki beberapa sebutan yaitu Hanacaraka, Carakan, dan Dentawyanjana.

Aksara Jawa sudah digunakan untuk penulisan sastra dan untuk penulisan sehari-harinya oleh masyarakat Jawa pada abad ke-17. Aksara Jawa memiliki bentuk yang unik dan bervariasi, namun pada zaman sekarang penggunaan aksara Jawa di lingkungan masyarakat sudah sangat jarang digunakan tetapi masih dilestarikan oleh beberapa pihak. Dikarenakan penggunaan aksara Jawa sudah jarang digunakan maka pentingnya untuk dilakukan pelestarian agar aksara tersebut tidak hilang dan terlupakan, khususnya bagi generasi muda di zaman sekarang perlunya kesadaran untuk melestarikan dan mempelajari aksara Jawa dikarenakan pada generasi muda Indonesia sekarang kebanyakan tidak bisa membaca maupun menulis aksara Jawa. Menurut Adi Prasetyo Tedjakusuma, aksara Jawa pada zaman sekarang sudah tidak banyak dikenali oleh para generasi muda Indonesia (Ilham & Rochmawati, 2020).

Menurut Arpeni Rahmawati, salah satu faktor yang membuat aksara Jawa dilupakan karena aksara Jawa masih sangat minim diajarkan di sekolah (Bahasa, 2021). Dengan melihat pentingnya budaya, penulis melakukan perancangan aplikasi terhadap pengenalan aksara Jawa. Menurut Sri Sultan Hamengkubuwono X, literasi aksara Jawa harus berjalan seiring dengan adanya pemanfaatan teknologi informasi dan komunikasi (Wicaksono, 2021), sehingga dengan diciptakannya aplikasi ini dapat membantu dalam upaya pelestarian aksara Jawa di Indonesia khususnya bagi generasi muda dan juga dapat digunakan sebagai alat atau sarana pembelajaran.

Dengan berkembangnya teknologi khususnya dalam bidang *Artificial Intelligence*, *Artificial Intelligence* sudah ada sejak 1950-an yang merupakan kemampuan sebuah komputer dalam memecahkan sebuah permasalahan atau tugas yang biasanya dikerjakan oleh manusia. Teknologi *Artificial Intelligence* juga memiliki kemampuan dalam melakukan klasifikasi citra kedalam kelompok tertentu. Perancangan aplikasi pengenalan aksara Jawa telah melewati banyak tahapan penelitian untuk mencari metode penelitian yang tepat dalam pengenalan dan klasifikasi terhadap aksara Jawa, dengan adanya penggunaan metode yang tepat dapat meningkatkan nilai akurasi dalam pengenalan dan klasifikasi aksara Jawa dengan tingkat *error* yang rendah. Terdapat beberapa penelitian dalam pengenalan karakter dengan menggunakan metode *Neural Network* menunjukkan hasil yang baik dalam pengenalan data berupa gambar.

Penelitian ini menggunakan algoritma *Convolutional Neural Network* (CNN) yang merupakan salah satu algoritma *Neural Network* yang terdapat pada *Deep Learning*. *Deep Learning* merupakan algoritma pada bidang ilmu pembelajaran mesin yang didalamnya terdiri dari algoritma pemodelan dengan abstraksi yang tinggi. Sedangkan *Convolutional Neural Network* merupakan jaringan saraf tiruan yang dapat digunakan untuk menganalisis citra visual dengan tingkat kesulitan yang cukup sulit dan dapat mengekstraksi fitur topologis dari gambar. *Convolutional Neural Network* dipilih karena dapat digunakan untuk melakukan *image processing* dengan baik dan pengenalan terhadap jenis *data* yang cukup kompleks seperti aksara Jawa yang memiliki bentuk yang beragam dan bervariasi. Dalam beberapa penelitian terdahulu

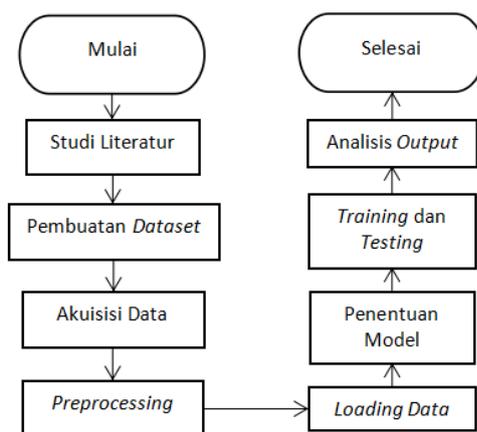
Convolutional Neural Network menunjukkan hasil yang baik dalam hal akurasi pengenalan karakter aksara Jawa.

Berdasarkan studi literatur yang telah dilakukan penulis pada penelitian terdahulu masih menggunakan *dataset* aksara Jawa tulisan tangan murni sedangkan pada penelitian ini penulis akan menggunakan *dataset* aksara Jawa tulisan tangan yang dibuat secara digital oleh penulis dengan menggunakan aplikasi gambar *Paint* sebagai keterbaruan dalam penelitian tentang pengenalan karakter aksara Jawa. Serta penulis menambahkan fitur *scan* dengan memanfaatkan kemampuan *Computer Vision* dalam mendeteksi gambar, namun deteksi gambar masih ditujukan hanya untuk gambar stasis.

Penulis berharap dalam penelitian ini yaitu mendapatkan hasil yang memuaskan dalam hal akurasi dan penggunaan algoritma *Convolutional Neural Network* agar dapat mengenali gambar aksara Jawa dengan baik. Selain itu penulis juga berharap dengan pemanfaatan *Computer Vision* dapat melakukan *scan* dan pengenalan gambar dengan baik.

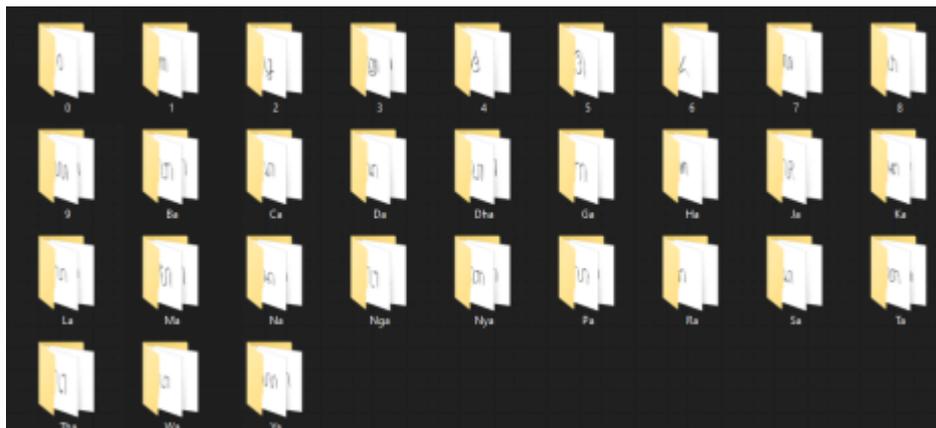
METODE

Pada penelitian ini penulis mencoba untuk merancang pengenalan gambar karakter aksara Jawa tulisan tangan digital. Target dari diadakannya penelitian ini yaitu generasi muda Indonesia dan juga orang dewasa agar mereka dapat lebih mengenal, mempelajari, dan melestarikan aksara Jawa agar tidak punah.



Gambar 1. Bagan Alur Penelitian

Penulis memulai penelitian dengan tahapan pertama berupa studi literatur yang dilakukan oleh penulis dengan mencari dan membaca sumber referensi tertulis dari jurnal, buku, dan *internet*. Setelah melalui tahapan studi literatur, penulis mulai melakukan tahap kedua yaitu pembuatan *dataset* gambar aksara Jawa tulisan tangan secara digital dengan menggunakan aplikasi gambar *Paint*. *Dataset* yang dibuat oleh penulis terdiri dari aksara Jawa Hanacaraka dan aksara Jawa angka dengan total 30 kelas karakter, karakter aksara yang dibuat oleh penulis yaitu Ha, Na, Ca, Ra, Ka, Da, Ta, Sa, Wa, La, Pa, Dha, Ja, Ya, Nya, Ma, Ga, Ba, Tha, Nga, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.



Gambar 2. *Dataset* Aksara Jawa Tulisan Tangan Digital

Dari total 30 kelas karakter aksara Jawa yang penulis buat masing-masing kelas terdiri dari 35 gambar sehingga total keseluruhan gambar yang dibuat oleh penulis berjumlah 1.050 gambar dengan dimensi ukuran 150 pixel x 150 pixel. Kemudian gambar yang telah dibuat dipisah menjadi 2 kelompok yaitu *data train* dan *data test* dengan perbandingan pembagian 8 banding 2, sehingga terdapat 840 gambar *train* dan 210 gambar *test*.

Tabel 1. Pembagian *Data Train* dan *Data Test*

Kelas	Total Data Train	Total Data Test	Kelas	Total Data Train	Total Data Test
0	28	7	Da	28	7
1	28	7	Ta	28	7
2	28	7	Sa	28	7
3	28	7	Wa	28	7
4	28	7	La	28	7
5	28	7	Pa	28	7
6	28	7	Dha	28	7
7	28	7	Ja	28	7
8	28	7	Ya	28	7
9	28	7	Nya	28	7
Ha	28	7	Ma	28	7
Na	28	7	Ga	28	7
Ca	28	7	Ba	28	7
Ra	28	7	Tha	28	7
Ka	28	7	Nga	28	7

Tahap ketiga yaitu memasukkan *folder* gambar aksara Jawa ini ke dalam *Google Drive* dan kemudian dihubungkan dengan *Google Collab* agar *data* yang ada di *folder* bisa terbaca di *Google Collab*, lalu dilakukan *import libraries* yang dibutuhkan untuk melakukan *image processing* pada gambar. Penulis menggunakan *Google Collab* karena sumber daya yang tersedia pada layanan *cloud Google Collab* sangat baik sehingga penulis dapat melakukan penelitian dengan proses yang lebih cepat.

```
#Data directory
data_dir = "/content/drive/MyDrive/Aksara-Jawa"
```

Gambar 3. Menghubungkan *Google Drive* dengan *Google Collab*

```
import tensorflow as tf
import os
import cv2
import imghdr
from matplotlib import pyplot as plt

import numpy as np

from sklearn.model_selection import train_test_split

from keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, MaxPool2D, Dense, Flatten, Dropout

from tensorflow.keras.metrics import Precision, Recall, Accuracy

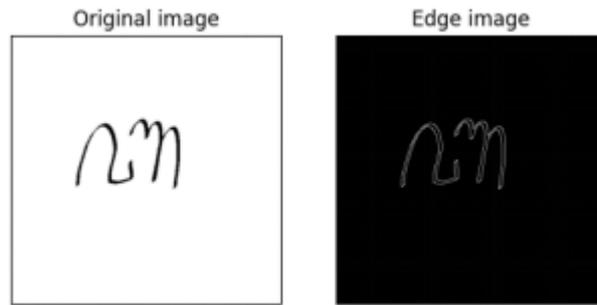
import numpy as np
import os
from sklearn.metrics import confusion_matrix
import seaborn as sn; sn.set(font_scale=1.4)
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
import cv2
import tensorflow as tf
from tqdm import tqdm

from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode

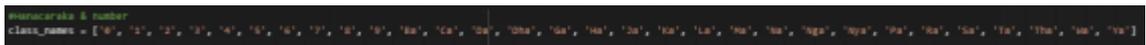
from IPython.display import Image
```

Gambar 4. Melakukan *import libraries*

Pada tahapan keempat penulis melakukan *pre-processing* yang terdiri dari proses segmentasi pada gambar, penulis melakukan pengubahan gambar menjadi bentuk *grayscale* atau pengubahan warna menjadi hitam-putih untuk mempermudah pemrosesan citra. Lalu penulis juga memasukkan nama *label* sesuai dengan 30 kelas karakter yang terdapat pada *dataset*.



Gambar 5. Pengubahan gambar ke bentuk *grayscale*



Gambar 6. Pemberian *label* sesuai dengan nama kelas *dataset*

Tahapan kelima berupa penulis melakukan *load data* dari *Google Drive* untuk gambar dan *label* pada *folder* pelatihan (*train*) dan *folder* pengujian (*test*).

Loading /content/drive/MyDrive/Aksara-Jawa/train				Loading /content/drive/MyDrive/Aksara-Jawa/test			
100%		28/28	[00:08<00:00, 3.31it/s]	100%		7/7	[00:01<00:00, 3.89it/s]
100%		28/28	[00:08<00:00, 3.49it/s]	100%		7/7	[00:02<00:00, 3.49it/s]
100%		28/28	[00:08<00:00, 3.34it/s]	100%		7/7	[00:01<00:00, 4.16it/s]
100%		28/28	[00:07<00:00, 3.57it/s]	100%		7/7	[00:02<00:00, 3.50it/s]
100%		28/28	[00:07<00:00, 3.64it/s]	100%		7/7	[00:01<00:00, 3.53it/s]
100%		28/28	[00:08<00:00, 3.36it/s]	100%		7/7	[00:01<00:00, 4.18it/s]
100%		28/28	[00:07<00:00, 3.70it/s]	100%		7/7	[00:01<00:00, 4.14it/s]
100%		28/28	[00:06<00:00, 4.05it/s]	100%		7/7	[00:02<00:00, 3.26it/s]
100%		28/28	[00:07<00:00, 3.79it/s]	100%		7/7	[00:01<00:00, 3.95it/s]
100%		28/28	[00:08<00:00, 3.49it/s]	100%		7/7	[00:01<00:00, 3.93it/s]
100%		28/28	[00:07<00:00, 3.86it/s]	100%		7/7	[00:02<00:00, 3.50it/s]
100%		28/28	[00:07<00:00, 3.68it/s]	100%		7/7	[00:01<00:00, 3.74it/s]
100%		28/28	[00:08<00:00, 3.37it/s]	100%		7/7	[00:01<00:00, 3.75it/s]
100%		28/28	[00:07<00:00, 3.79it/s]	100%		7/7	[00:01<00:00, 4.13it/s]
100%		28/28	[00:07<00:00, 3.52it/s]	100%		7/7	[00:01<00:00, 3.86it/s]
100%		28/28	[00:08<00:00, 3.42it/s]	100%		7/7	[00:01<00:00, 3.68it/s]
100%		28/28	[00:07<00:00, 3.71it/s]	100%		7/7	[00:02<00:00, 3.30it/s]
100%		28/28	[00:07<00:00, 3.62it/s]	100%		7/7	[00:01<00:00, 3.73it/s]
100%		28/28	[00:07<00:00, 3.60it/s]	100%		7/7	[00:01<00:00, 3.95it/s]
100%		28/28	[00:07<00:00, 3.98it/s]	100%		7/7	[00:01<00:00, 3.89it/s]
100%		28/28	[00:07<00:00, 3.87it/s]	100%		7/7	[00:01<00:00, 3.60it/s]
100%		28/28	[00:07<00:00, 3.78it/s]	100%		7/7	[00:01<00:00, 3.98it/s]
100%		28/28	[00:07<00:00, 3.72it/s]	100%		7/7	[00:01<00:00, 4.18it/s]
100%		28/28	[00:08<00:00, 3.48it/s]	100%		7/7	[00:01<00:00, 3.95it/s]
100%		28/28	[00:07<00:00, 3.75it/s]	100%		7/7	[00:01<00:00, 3.50it/s]
100%		28/28	[00:07<00:00, 3.79it/s]	100%		7/7	[00:01<00:00, 4.12it/s]
100%		28/28	[00:07<00:00, 3.93it/s]	100%		7/7	[00:02<00:00, 3.46it/s]
100%		28/28	[00:07<00:00, 3.92it/s]	100%		7/7	[00:01<00:00, 3.85it/s]
100%		28/28	[00:07<00:00, 3.99it/s]	100%		7/7	[00:01<00:00, 4.05it/s]
100%		28/28	[00:07<00:00, 3.82it/s]	100%		7/7	[00:01<00:00, 4.13it/s]

Gambar 7. *Load data train dan test*

Tahapan keenam berupa penentuan model CNN yang tepat untuk pengenalan dan klasifikasi gambar aksara Jawa tulisan tangan digital, dengan penggunaan model yang tepat dapat membantu dalam perolehan dari segi akurasi *training* dan *testing*. Model yang digunakan dalam penelitian ini mengalami beberapa kali modifikasi hingga penulis mendapatkan model arsitektur yang terbaik.

```

model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu', input_shape = (150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(64, (3, 3), activation = 'relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3, 3), activation = 'relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(256, (3, 3), activation = 'relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Flatten(),

    tf.keras.layers.Dense(256, activation=tf.nn.relu),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(30, activation=tf.nn.softmax)
])
    
```

Gambar 8. Model arsitektur terbaik yang digunakan oleh penulis

Tahapan ketujuh berupa tahapan *training* dan *testing*, pada tahapan ini dilakukan pelatihan dan evaluasi dari model yang digunakan dengan menggunakan *data training* dan *data testing* yang telah dibagi sebelumnya. Tahapan terakhir berupa analisis terhadap *output* yang berupa perolehan nilai akurasi dan nilai *loss* dari segi *training* dan *testing*.

HASIL DAN PEMBAHASAN

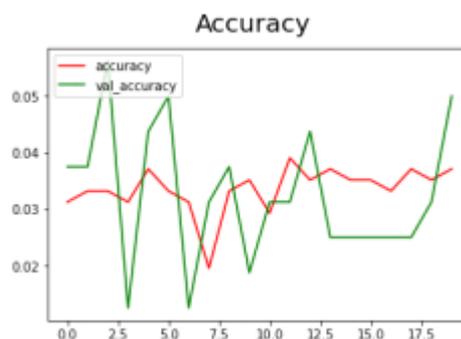
Berdasarkan penelitian yang telah dilakukan oleh penulis, penulis mencoba beberapa macam model untuk klasifikasi terhadap aksara Jawa berdasarkan kelompoknya dengan menggunakan algoritma *Convolutional Neural Network* sehingga adanya proses *train* dan *test* yang berulang hingga mendapatkan akurasi yang memuaskan. Dari percobaan penggunaan model yang berbeda-beda membuat hasil akurasi yang didapatkan juga berbeda dalam pengenalan gambar aksara Jawa. Pada percobaan pertama, *dataset* yang digunakan penulis hanya memiliki jumlah total 750 gambar untuk 30 karakter aksara Jawa serta menggunakan 20 *epoch* dan dengan 16 *cycle* untuk train data. Dari percobaan pertama itu, penulis mendapatkan akurasi yang sangat kecil dari segi *train* dan *test data*.

```

Epoch 1/20
15/16 [====] - 45s 3s/step - loss: -3895.1191 - accuracy: 0.0312 - val_loss: -52954.2578 - val_accuracy: 0.0375
Epoch 2/20
15/16 [====] - 44s 3s/step - loss: -393937.4062 - accuracy: 0.0312 - val_loss: -5271329.3290 - val_accuracy: 0.0375
Epoch 3/20
15/16 [====] - 41s 3s/step - loss: -5131344.0000 - accuracy: 0.0312 - val_loss: -11671509.0000 - val_accuracy: 0.0562
Epoch 4/20
15/16 [====] - 46s 3s/step - loss: -36795832.0000 - accuracy: 0.0312 - val_loss: -94252016.0000 - val_accuracy: 0.0125
Epoch 5/20
15/16 [====] - 41s 2s/step - loss: -182109096.0000 - accuracy: 0.0371 - val_loss: -364802496.0000 - val_accuracy: 0.0437
Epoch 6/20
15/16 [====] - 43s 3s/step - loss: -697287808.0000 - accuracy: 0.0312 - val_loss: -1848004832.0000 - val_accuracy: 0.0500
Epoch 7/20
15/16 [====] - 41s 3s/step - loss: -2846030736.0000 - accuracy: 0.0312 - val_loss: -3366633600.0000 - val_accuracy: 0.0125
Epoch 8/20
15/16 [====] - 39s 2s/step - loss: -5438412896.0000 - accuracy: 0.0195 - val_loss: -8674239008.0000 - val_accuracy: 0.0312
Epoch 9/20
15/16 [====] - 39s 2s/step - loss: -12109665984.0000 - accuracy: 0.0332 - val_loss: -15815650384.0000 - val_accuracy: 0.0375
Epoch 10/20
15/16 [====] - 41s 2s/step - loss: -28170893088.0000 - accuracy: 0.0392 - val_loss: -7345388192.0000 - val_accuracy: 0.0188
Epoch 11/20
15/16 [====] - 44s 3s/step - loss: -49269195520.0000 - accuracy: 0.0293 - val_loss: -6971337536.0000 - val_accuracy: 0.0312
Epoch 12/20
15/16 [====] - 66s 3s/step - loss: -89811998624.0000 - accuracy: 0.0393 - val_loss: -128861107200.0000 - val_accuracy: 0.0312
Epoch 13/20
15/16 [====] - 45s 3s/step - loss: -151566811136.0000 - accuracy: 0.0352 - val_loss: -197244046880.0000 - val_accuracy: 0.0437
Epoch 14/20
15/16 [====] - 46s 3s/step - loss: -243039888512.0000 - accuracy: 0.0371 - val_loss: -309956617728.0000 - val_accuracy: 0.0290
Epoch 15/20
15/16 [====] - 43s 2s/step - loss: -398533896432.0000 - accuracy: 0.0352 - val_loss: -46754679328.0000 - val_accuracy: 0.0250
Epoch 16/20
15/16 [====] - 94s 3s/step - loss: -598148132032.0000 - accuracy: 0.0392 - val_loss: -717948298880.0000 - val_accuracy: 0.0290
Epoch 17/20
15/16 [====] - 42s 3s/step - loss: -856795488032.0000 - accuracy: 0.0332 - val_loss: -908448116736.0000 - val_accuracy: 0.0250
Epoch 18/20
15/16 [====] - 47s 3s/step - loss: -1252664818032.0000 - accuracy: 0.0371 - val_loss: -1482824493956.0000 - val_accuracy: 0.0290
Epoch 19/20
15/16 [====] - 45s 3s/step - loss: -1696451126272.0000 - accuracy: 0.0352 - val_loss: -2068020886576.0000 - val_accuracy: 0.0312
Epoch 20/20
15/16 [====] - 44s 3s/step - loss: -2322617679872.0000 - accuracy: 0.0371 - val_loss: -2832869468864.0000 - val_accuracy: 0.0500

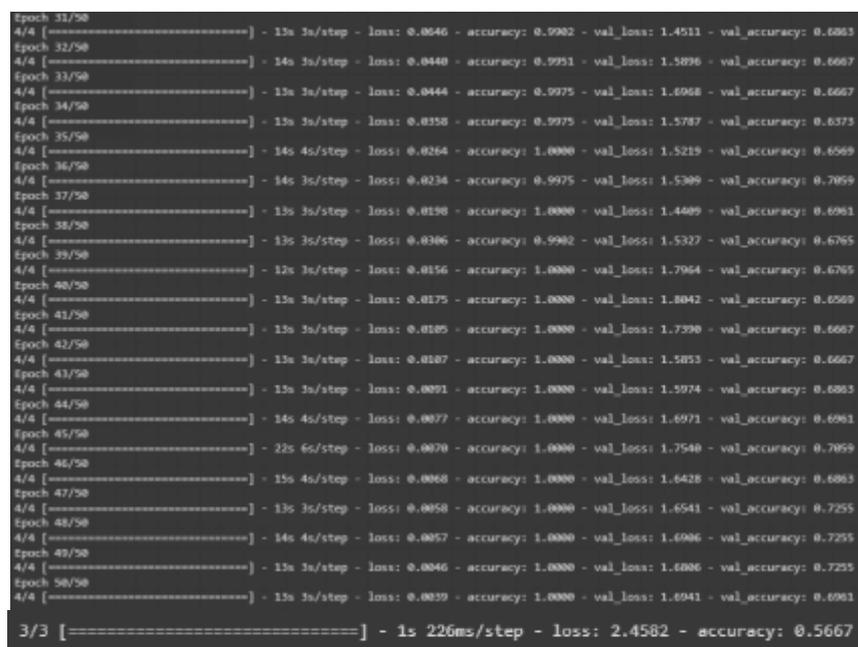
Precision: 0.9230769276618958, Recall: 1.0, Accuracy: 0.025641025975346565
    
```

Gambar 9. Hasil akurasi *train* (atas) dan *test* (bawah) pada percobaan pertama

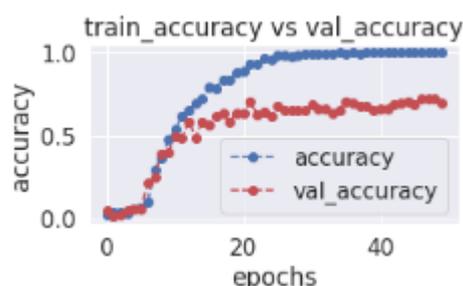


Gambar 10. Grafik akurasi *train* dan *validation accuracy* pada percobaan pertama

Dari hasil percobaan sebelumnya, penulis merasa akurasi yang didapat belum mencapai target yang diinginkan sehingga penulis mencoba membuat model baru dengan arsitektur yang berbeda. Pada percobaan kedua ini penulis menggunakan 50 *epoch* dan 4 *cycle*, pada percobaan kedua ini mengalami kenaikan akurasi yang cukup signifikan dengan adanya perubahan model yang dilakukan oleh penulis.



Gambar 11. Hasil akurasi *train* (atas) dan *test* (bawah) pada percobaan kedua



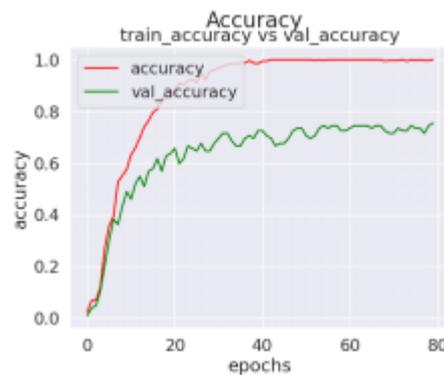
Gambar 12. Grafik akurasi *train* dan *validation accuracy* pada percobaan kedua

Setelah percobaan kedua penulis membuat percobaan ketiga dengan membuat dan menggunakan model arsitektur yang berbeda lagi dengan tujuan agar dapat meningkatkan akurasi menjadi lebih baik lagi. Pada percobaan ketiga ini penulis menaikkan jumlah *epoch* yaitu

menggunakan 80 *epoch* dan dengan 4 *cycle*. Dari percobaan ketiga ini terjadi kenaikan dalam segi akurasi pada *train* dan *test*.

```
Epoch 61/80  
4/4 [-----] - 8s 2s/step - loss: 0.0033 - accuracy: 1.0000 - val_loss: 1.7404 - val_accuracy: 0.7353  
Epoch 62/80  
4/4 [-----] - 9s 2s/step - loss: 0.0030 - accuracy: 1.0000 - val_loss: 1.7573 - val_accuracy: 0.7451  
Epoch 63/80  
4/4 [-----] - 7s 2s/step - loss: 0.0028 - accuracy: 1.0000 - val_loss: 1.7737 - val_accuracy: 0.7451  
Epoch 64/80  
4/4 [-----] - 9s 2s/step - loss: 0.0027 - accuracy: 1.0000 - val_loss: 1.7935 - val_accuracy: 0.7451  
Epoch 65/80  
4/4 [-----] - 8s 2s/step - loss: 0.0026 - accuracy: 1.0000 - val_loss: 1.8051 - val_accuracy: 0.7451  
Epoch 66/80  
4/4 [-----] - 9s 2s/step - loss: 0.0025 - accuracy: 1.0000 - val_loss: 1.8054 - val_accuracy: 0.7353  
Epoch 67/80  
4/4 [-----] - 9s 2s/step - loss: 0.0033 - accuracy: 1.0000 - val_loss: 1.7981 - val_accuracy: 0.7451  
Epoch 68/80  
4/4 [-----] - 7s 2s/step - loss: 0.0029 - accuracy: 1.0000 - val_loss: 1.8452 - val_accuracy: 0.7451  
Epoch 69/80  
4/4 [-----] - 9s 2s/step - loss: 0.0091 - accuracy: 0.9951 - val_loss: 1.7769 - val_accuracy: 0.7255  
Epoch 70/80  
4/4 [-----] - 8s 2s/step - loss: 0.0066 - accuracy: 0.9975 - val_loss: 1.7637 - val_accuracy: 0.7157  
Epoch 71/80  
4/4 [-----] - 9s 2s/step - loss: 0.0082 - accuracy: 0.9975 - val_loss: 1.7272 - val_accuracy: 0.7157  
Epoch 72/80  
4/4 [-----] - 12s 3s/step - loss: 0.0100 - accuracy: 0.9975 - val_loss: 1.7518 - val_accuracy: 0.7353  
Epoch 73/80  
4/4 [-----] - 15s 3s/step - loss: 0.0039 - accuracy: 1.0000 - val_loss: 1.7811 - val_accuracy: 0.7255  
Epoch 74/80  
4/4 [-----] - 7s 2s/step - loss: 0.0229 - accuracy: 0.9951 - val_loss: 1.7283 - val_accuracy: 0.7255  
Epoch 75/80  
4/4 [-----] - 10s 2s/step - loss: 0.0040 - accuracy: 1.0000 - val_loss: 1.7472 - val_accuracy: 0.7451  
Epoch 76/80  
4/4 [-----] - 9s 2s/step - loss: 0.0122 - accuracy: 0.9975 - val_loss: 1.7505 - val_accuracy: 0.7353  
Epoch 77/80  
4/4 [-----] - 8s 2s/step - loss: 0.0065 - accuracy: 1.0000 - val_loss: 1.7203 - val_accuracy: 0.7353  
Epoch 78/80  
4/4 [-----] - 9s 2s/step - loss: 0.0073 - accuracy: 0.9975 - val_loss: 1.6882 - val_accuracy: 0.7157  
Epoch 79/80  
4/4 [-----] - 7s 2s/step - loss: 0.0052 - accuracy: 0.9975 - val_loss: 1.6498 - val_accuracy: 0.7451  
Epoch 80/80  
4/4 [-----] - 10s 2s/step - loss: 0.0027 - accuracy: 1.0000 - val_loss: 1.6636 - val_accuracy: 0.7549  
3/3 [-----] - 0s 132ms/step - loss: 2.5647 - accuracy: 0.7111
```

Gambar 13. Hasil akurasi *train* (atas) dan *test* (bawah) pada percobaan ketiga



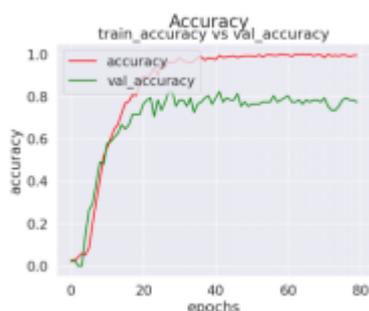
Gambar 14. Grafik akurasi *train* dan *validation accuracy* pada percobaan ketiga

Pada percobaan keempat penulis mencoba untuk melakukan perubahan model arsitektur lagi agar dapat menaikkan akurasi menjadi lebih tinggi dan mengalami peningkatan sebesar 2 persen untuk akurasi *testing*.

```

Epoch 62/80
4/4 [=====] - 26s 6s/step - loss: 0.0158 - accuracy: 0.9902 - val_loss: 1.0866 - val_accuracy: 0.7745
Epoch 63/80
4/4 [=====] - 24s 6s/step - loss: 0.0026 - accuracy: 1.0000 - val_loss: 1.1478 - val_accuracy: 0.7745
Epoch 64/80
4/4 [=====] - 25s 6s/step - loss: 0.0077 - accuracy: 0.9975 - val_loss: 1.1858 - val_accuracy: 0.7843
Epoch 65/80
4/4 [=====] - 26s 6s/step - loss: 0.0038 - accuracy: 1.0000 - val_loss: 1.1700 - val_accuracy: 0.7647
Epoch 66/80
4/4 [=====] - 25s 6s/step - loss: 0.0050 - accuracy: 0.9975 - val_loss: 1.1598 - val_accuracy: 0.7843
Epoch 67/80
4/4 [=====] - 23s 6s/step - loss: 0.0053 - accuracy: 1.0000 - val_loss: 1.2150 - val_accuracy: 0.7745
Epoch 68/80
4/4 [=====] - 26s 6s/step - loss: 0.0161 - accuracy: 0.9926 - val_loss: 1.2552 - val_accuracy: 0.7843
Epoch 69/80
4/4 [=====] - 25s 6s/step - loss: 0.0128 - accuracy: 0.9975 - val_loss: 1.3852 - val_accuracy: 0.7843
Epoch 70/80
4/4 [=====] - 24s 6s/step - loss: 0.0106 - accuracy: 0.9975 - val_loss: 1.3074 - val_accuracy: 0.7745
Epoch 71/80
4/4 [=====] - 26s 6s/step - loss: 0.0533 - accuracy: 0.9853 - val_loss: 1.1495 - val_accuracy: 0.7941
Epoch 72/80
4/4 [=====] - 26s 6s/step - loss: 0.0080 - accuracy: 1.0000 - val_loss: 1.4741 - val_accuracy: 0.7549
Epoch 73/80
4/4 [=====] - 24s 6s/step - loss: 0.0231 - accuracy: 0.9926 - val_loss: 1.6317 - val_accuracy: 0.7353
Epoch 74/80
4/4 [=====] - 24s 6s/step - loss: 0.0263 - accuracy: 0.9951 - val_loss: 1.6932 - val_accuracy: 0.7353
Epoch 75/80
4/4 [=====] - 26s 6s/step - loss: 0.0176 - accuracy: 0.9926 - val_loss: 1.4037 - val_accuracy: 0.7549
Epoch 76/80
4/4 [=====] - 25s 6s/step - loss: 0.0128 - accuracy: 0.9975 - val_loss: 1.1702 - val_accuracy: 0.7843
Epoch 77/80
4/4 [=====] - 24s 6s/step - loss: 0.0215 - accuracy: 0.9951 - val_loss: 1.0577 - val_accuracy: 0.7941
Epoch 78/80
4/4 [=====] - 26s 6s/step - loss: 0.0221 - accuracy: 0.9902 - val_loss: 1.2615 - val_accuracy: 0.7843
Epoch 79/80
4/4 [=====] - 26s 6s/step - loss: 0.0142 - accuracy: 0.9951 - val_loss: 1.3091 - val_accuracy: 0.7843
Epoch 80/80
4/4 [=====] - 24s 6s/step - loss: 0.0318 - accuracy: 0.9951 - val_loss: 1.1756 - val_accuracy: 0.7745
3/3 [=====] - 1s 381ms/step - loss: 1.7982 - accuracy: 0.7333
    
```

Gambar 15. Hasil akurasi *train* (atas) dan *test* (bawah) pada percobaan keempat



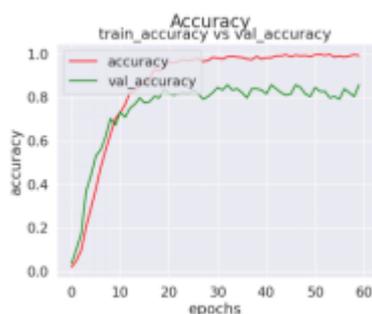
Gambar 16. Grafik akurasi *train* dan *validation accuracy* pada percobaan keempat

Pada percobaan kelima penulis melakukan penambahan gambar pada *dataset* aksara Jawa dengan total keseluruhan menjadi 1.050 gambar dengan masing-masing karakter terdiri dari 35 gambar, kemudian *data* tersebut dibagi menjadi *data train* dan *data test*. Untuk model arsitektur yang digunakan pada percobaan kelima ini menggunakan model arsitektur yang sama dengan percobaan keempat tetapi terdapat perubahan dalam jumlah *epoch* yang digunakan menjadi 60 *epoch* dan 6 *cycle*. Dari hasil percobaan kelima ini mendapatkan kenaikan akurasi yang cukup memuaskan pada bagian *test*.

```

Epoch 42/60
6/6 [-----] - 44s 7s/step - loss: 0.0653 - accuracy: 0.9821 - val_loss: 0.7392 - val_accuracy: 0.8571
Epoch 43/60
6/6 [-----] - 44s 7s/step - loss: 0.0348 - accuracy: 0.9911 - val_loss: 0.7211 - val_accuracy: 0.8393
Epoch 44/60
6/6 [-----] - 46s 8s/step - loss: 0.0255 - accuracy: 0.9911 - val_loss: 0.7685 - val_accuracy: 0.8214
Epoch 45/60
6/6 [-----] - 47s 8s/step - loss: 0.0158 - accuracy: 0.9955 - val_loss: 0.9511 - val_accuracy: 0.8695
Epoch 46/60
6/6 [-----] - 47s 8s/step - loss: 0.0295 - accuracy: 0.9881 - val_loss: 0.9744 - val_accuracy: 0.8155
Epoch 47/60
6/6 [-----] - 44s 7s/step - loss: 0.0186 - accuracy: 0.9940 - val_loss: 0.9140 - val_accuracy: 0.8571
Epoch 48/60
6/6 [-----] - 44s 7s/step - loss: 0.0370 - accuracy: 0.9911 - val_loss: 0.7944 - val_accuracy: 0.8452
Epoch 49/60
6/6 [-----] - 44s 7s/step - loss: 0.0238 - accuracy: 0.9926 - val_loss: 0.9400 - val_accuracy: 0.8155
Epoch 50/60
6/6 [-----] - 50s 9s/step - loss: 0.0467 - accuracy: 0.9881 - val_loss: 0.9366 - val_accuracy: 0.8036
Epoch 51/60
6/6 [-----] - 44s 7s/step - loss: 0.0186 - accuracy: 0.9955 - val_loss: 0.8728 - val_accuracy: 0.8452
Epoch 52/60
6/6 [-----] - 44s 7s/step - loss: 0.0138 - accuracy: 0.9970 - val_loss: 0.8887 - val_accuracy: 0.8274
Epoch 53/60
6/6 [-----] - 44s 7s/step - loss: 0.0228 - accuracy: 0.9926 - val_loss: 1.0019 - val_accuracy: 0.8274
Epoch 54/60
6/6 [-----] - 44s 7s/step - loss: 0.0151 - accuracy: 0.9970 - val_loss: 1.1042 - val_accuracy: 0.7976
Epoch 55/60
6/6 [-----] - 45s 8s/step - loss: 0.0311 - accuracy: 0.9851 - val_loss: 1.1092 - val_accuracy: 0.8036
Epoch 56/60
6/6 [-----] - 46s 8s/step - loss: 0.0297 - accuracy: 0.9911 - val_loss: 1.1894 - val_accuracy: 0.7917
Epoch 57/60
6/6 [-----] - 46s 8s/step - loss: 0.0458 - accuracy: 0.9851 - val_loss: 0.8830 - val_accuracy: 0.8393
Epoch 58/60
6/6 [-----] - 44s 7s/step - loss: 0.0373 - accuracy: 0.9911 - val_loss: 0.8928 - val_accuracy: 0.8214
Epoch 59/60
6/6 [-----] - 45s 8s/step - loss: 0.0173 - accuracy: 0.9955 - val_loss: 0.9794 - val_accuracy: 0.8036
Epoch 60/60
6/6 [-----] - 44s 8s/step - loss: 0.0228 - accuracy: 0.9926 - val_loss: 1.0636 - val_accuracy: 0.8571
7/7 [-----] - 3s 447ms/step - loss: 0.8910 - accuracy: 0.8476
    
```

Gambar 17. Hasil akurasi *train* (atas) dan *test* (bawah) pada percobaan kelima

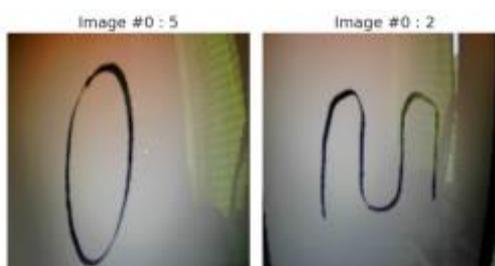


Gambar 18. Grafik akurasi *train* dan *validation accuracy* pada percobaan kelima

Setelah percobaan kelima, penulis memulai penggunaan *Computer Vision* dalam melakukan klasifikasi gambar aksara Jawa dengan *output* berupa klasifikasi gambar sesuai dengan nama kelompok aksara yang tepat. Penulis menggunakan *webcam* sebagai *input device* untuk memfoto gambar aksara Jawa tulisan tangan digital, pada percobaan *Computer Vision* ini penulis menggunakan pengujian 5 karakter aksara yaitu 0, 4, 5, 9, dan Pa. Dari 5 kali percobaan yang penulis lakukan mendapatkan hasil 3 kali berhasil pada karakter 4, 5, dan 9 serta 2 kali gagal pada karakter 0 dan Pa, kesalahan klasifikasi yang terjadi disebabkan dari *input* gambar yang kurang bagus dan kurang kontras sehingga menyebabkan kesalahan dalam klasifikasi gambar.



Gambar 19. Percobaan yang berhasil pada pengenalan karakter 4, 5, dan 9 (kiri ke kanan)



Gambar 20. Percobaan yang gagal pada pengenalan karakter 0 dan Pa (kiri ke kanan)

Setelah percobaan dengan *input* gambar berupa aksara Jawa tulisan tangan digital penulis melakukan percobaan dengan melakukan *input* gambar berupa aksara Jawa tulisan tangan murni. Dalam percobaan dengan menggunakan gambar aksara Jawa tulisan tangan murni masih mendapatkan hasil yang kurang memuaskan karena belum dapat melakukan klasifikasi dengan baik. Klasifikasi yang gagal pada pengenalan aksara Jawa tulisan tangan murni dikarenakan kurangnya sampel gambar pada *dataset* yang dibuat oleh penulis untuk aksara Jawa tulisan tangan murni.

KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, penulis menyimpulkan bahwa hasil yang didapatkan dari penggunaan algoritma *Convolutional Neural Network* memberikan hasil yang cukup baik dan memuaskan dalam pengenalan gambar aksara Jawa tulisan tangan digital dengan akurasi yang didapatkan dari *testing* sebesar 84% dan tingkat *error* sebesar 16% sesuai dengan tujuan penelitian yang menjawab permasalahan berupa pelestarian aksara Jawa dengan memanfaatkan teknologi sebagai sarana pembelajaran dan pelestarian budaya bagi para generasi muda dan juga bagi orang dewasa. Penggunaan *Computer Vision* pada penelitian ini juga memiliki hasil yang cukup baik untuk pengenalan gambar aksara Jawa tulisan tangan digital akan tetapi masih memiliki kekurangan pada pengenalan gambar aksara Jawa tulisan tangan murni dan juga masih terdapat beberapa klasifikasi yang salah dalam pengenalan gambar aksara Jawa tulisan tangan digital dikarenakan *input* gambar yang kurang bagus serta keterbatasan dalam segi *hardware* oleh penulis. Dari penelitian ini penulis berharap dapat membantu penelitian tentang topik aksara Jawa kedepannya serta pengembangan dari penggunaan *Computer Vision* agar selain dapat mengenali gambar aksara tulisan tangan digital juga dapat mengenali gambar aksara Jawa tulisan tangan murni dengan lebih baik.

Saran untuk penelitian di masa depan dari penulis yaitu pada penelitian berikutnya peneliti dapat menggunakan *hardware* yang lebih baik terutama untuk bagian *input device* contohnya dengan penggunaan *webcam* yang lebih bagus dan beresolusi tinggi, serta penambahan gambar aksara Jawa tulisan tangan murni serta gambar aksara Jawa tulisan tangan digital pada *dataset* agar lebih bervariasi sehingga ketika proses klasifikasi dapat mengenali 2 jenis gambar aksara

Jawa tulisan tangan murni dan gambar aksara Jawa tulisan tangan digital dengan akurasi yang lebih tinggi dan lebih baik.

DAFTAR PUSTAKA

- Afakh, M. L., Risnumawan, A., Anggraeni, M. E., Tamara, M. N., & Ningrum, E. S. (2017). Aksara Jawa Text Detection in Scene Images using Convolutional Neural Network. *Proceedings - International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2017, 2017-Janua*, 77-82. <https://doi.org/10.1109/KCIC.2017.8228567>
- Anggraeny, F. T., Via, Y. V., & Mumpuni, R. (2023). Image preprocessing analysis in handwritten Javanese character recognition. *Bulletin of Electrical Engineering and Informatics*, 12(2), 860-867. <https://doi.org/10.11591/eei.v12i2.4172>
- Bahasa. (2021). *Aksara Jawa Menolak Punah*. <https://budaya.jogjaprovo.go.id/berita/detail/753-aksara-jawa-menolak-punah>
- Chollet, F. (2021). *Deep Learning with Python* (2nd ed.). Manning.
- Dewa, C. K., Fadhilah, A. L., & Afiahayati, A. (2018). Convolutional Neural Networks for Handwritten Javanese Character Recognition. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 12(1), 83-94. <https://doi.org/10.22146/ijccs.31144>
- Diqi, M., & Muhdalifah, M. F. (2020). Design and Building Javanese Script Classification in The State Museum of Sonobudoyo Yogyakarta. *International Journal of Informatics and Computation*, 1(2), 35-45. <https://doi.org/10.35842/ijicom.v1i2.18>
- Diyasa, I. G. S. M., & Romadhon, R. (2023). Klasifikasi Karakter Tulisan Aksara Jawa Menggunakan Algoritma Convolutional Neural Network. *Seminar Keinsinyuran*, 3(1), 927-936. <https://doi.org/10.22219/skpsppi.v3i1.7720>
- G., A. R., Tandra, A., Susanto, I., Harefa, J., & Chowanda, A. (2019). Implementation of Optical character Recognition using Tesseract with the Javanese Script Target in Android Application. *Procedia Computer Science*, 157, 499-505. <https://doi.org/10.1016/j.procs.2019.09.006>
- Hanindria, I. S., & Hendry, H. (2022). Pengklasifikasian Aksara Jawa Metode Convolutional Neural Network. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 9(3), 2727-2737.
- Harjoseputro, Y. (2018). *Convolutional Neural Network (CNN) Untuk Pengklasifikasian Aksara Jawa* [Universitas Atma Jaya]. <http://e-journal.uajy.ac.id/15485/>
- Harjoseputro, Y. (2020). A Classification Javanese Letters Model using a Convolutional Neural Network with KERAS Framework. *International Journal of Advanced Computer Science and Applications*, 11(10), 106-111. <https://doi.org/10.14569/IJACSA.2020.0111014>
- Ilham, F., & Rochmawati, N. (2020). Transliterasi Aksara Jawa Tulisan Tangan ke Tulisan Latin Menggunakan CNN. *Journal of Informatics and Computer Science (JINACS)*, 1(04), 200-208. <https://doi.org/10.26740/jinacs.v1n04.p200-208>
- Lorentius, C. A., Adipranata, R., & Tjondrowiguno, A. (2020). Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network. *E-Proceeding of Engineering*, 7(1), 2558-2567.

- Putra, A. S. (2020). *Identifikasi Aksara Jawa pada Naskah Kuno dengan Metode CNN*. Universitas Islam Indonesia.
- Putri, A. A. (2023). *Suku Dengan Populasi Terbanyak di Indonesia*. <https://data.goodstats.id/statistic/adelandilaa/suku-dengan-populasi-terbanyak-di-indonesia-AJmNV#:~:text=Suku Jawa menjadi suku bangsa,Jawa Timur%2C dan DI Yogyakarta>.
- Rismiyati, Khadijah, & Nurhadiyah, A. (2017). Deep Learning for Handwritten Javanese Character Recognition. *Proceedings - 2017 1st International Conference on Informatics and Computational Sciences, ICICoS 2017*, 59-63. <https://doi.org/10.1109/ICICOS.2017.8276338>
- Riyandi, A., & 'Uyun, S. (2022). Improvement of Handwriting Javascript Image Quality and Segmentation With Closing Morphology and Adaptive Thresholding Methods. *Telematika*, 19(3), 311-322. <https://doi.org/10.31315/telematika.v19i3.7564>
- Romawati, E. F., Widaningrum, I., & Astuti, I. P. (2020). Rancang Bangun Aplikasi Mobile Pengenalan Huruf Jawa (Aksara Jawa) Berbasis Android. *Jurnal CoSciTech (Computer Science and Information Technology)*, 1(2), 93-100. <https://doi.org/10.37859/coscitech.v1i2.2185>
- Saphira, N. (2021). *Pengenalan Karakter Optik (OCR) Aksara Jawa dengan Convolutional Neural Network*. Universitas Kristen Duta Wacana.
- Sudana, O., Gunaya, I. W., & Putra, I. K. G. D. (2020). Handwriting identification using deep convolutional neural network method. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(4), 1934-1941. <https://doi.org/10.12928/TELKOMNIKA.V18I4.14864>
- Wicaksono, P. (2021). *Hari Aksara Internasional, Sultan: Aksara Jawa Harus Bisa Hidup di Ranah Digital*. <https://tekno.tempo.co/amp/1503949/hari-aksara-internasional-sultan-aksara-jawa-harus-bisa-hidup-di-ranah-digital>
- Wijaya, E. P. N. A. (2020). Klasifikasi Akasara Jawa Dengan CNN. *Jurnal Teknika*, 12(2), 61-64. <https://doi.org/10.30736/jt.v13i2.479>
- Zin, T. T., Thant, S., Pwint, M. Z., & Ogino, T. (2021). Handwritten Character Recognition on Android for Basic Education Using Convolutional Neural Network. *Journal of Physics: Conference Series*, 1918(4). <https://doi.org/10.1088/1742-6596/1918/4/042152>

How to cite:

Jonathan, A., & Wasito, I. (2023). Perancangan Aplikasi Pengenalan Aksara Jawa Digital Menggunakan Convolutional Neural Network dan Computer Vision. *DECODE: Jurnal Pendidikan Teknologi Informasi*, 3(2), 364-377. <http://dx.doi.org/10.51454/decode.v3i2.209>