

TUGAS PROYEK PENGENALAN POLA
PENGUNAAN METODE PENGENALAN POLA

Dosen Pengampu:

Imam Much Ibnu Subroto, ST., M.Sc., Ph.D



Disusun oleh:

KELOMPOK 7

Minan Abdillah	(32602100065)
Moch. Chaidar Chanif	(32602100066)
Moch. Isna Ardian Cahyana	(32602100067)
M. Sirojudin Mahdi Faza	(32602100069)

PRODI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG SEMARANG
2024

A. Pendahuluan

Dataset Aksara Jawa merupakan kumpulan gambar yang mewakili aksara-aksara Jawa. Dataset ini penting untuk melatih model machine learning yang dapat mengklasifikasikan aksara Jawa. Preprocessing data merupakan langkah penting sebelum melatih model, untuk memastikan data berkualitas baik dan siap untuk digunakan.

Berikut sumber Dataset dan Hasil Pengerjaannya :

GitHub :

<https://github.com/minanabdillah/Tugas-Pengenalan-Pola-HANACARAKA.git>

Google Colab:

<https://www.kaggle.com/datasets/hannanhunafa/javanese-script-aksara-jawa-augmented>

Link Gdrive:

[https://colab.research.google.com/drive/1ehnwJEyo9uWiNx4kTIVHFHSRW7QfT4VG?usp=drive link](https://colab.research.google.com/drive/1ehnwJEyo9uWiNx4kTIVHFHSRW7QfT4VG?usp=drive_link)

B. Langkah-langkah Preprocessing Data


1. Import Library yang dibutuhkan

```
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Conv2D, MaxPooling2D, MaxPool2D, Flatten, Dense, Dropout, Input, UpSampling2D, concatenate
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.callbacks import Callback, ReduceLROnPlateau
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.regularizers import l2
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

Import library yang diperlukan seperti numpy, tensorflow, dan matplotlib yang akan digunakan nantinya.

2. Load Data yang sebelumnya dalam bentuk zip

```
from google.colab import files
uploaded = files.upload()
```

 **Pilih File** | images.zip

- images.zip(application/zip) - 138611303 bytes, last modified: 6/6/2024 - 100% done

Saving images.zip to images.zip

Gambar diatas adalah load data melalui google colab files.

3. Unzip Data yang sudah di Load

```
!unzip images.zip

inflating: val/ya/ya_original_ya73.mine3.b.png_24a9b18b-f12c-4433-bf78-1b6ef5b2d151.png
inflating: val/ya/ya_original_ya73.mine3.b.png_2719f17d-fafe-4c36-bf89-4724c71fa513.png
inflating: val/ya/ya_original_ya73.mine3.b.png_34dcc3ae-60ae-443e-be6f-660354241248.png
inflating: val/ya/ya_original_ya73.mine3.b.png_a5a089fc-b7b9-4d5e-9fa2-68e04a5c3d2b.png
inflating: val/ya/ya_original_ya73.mine3.png_208cdc5d-6909-4c08-9aca-94a31d19e91b.png
inflating: val/ya/ya_original_ya73.mine3.png_4c7721f2-2ae9-4317-95e5-32ff584959b1.png
inflating: val/ya/ya_original_ya73.mine3.png_8c0be3e1-ba54-4638-9142-be5bd30d406c.png
inflating: val/ya/ya_original_ya73.mine3.png_e31a58fd-a949-4a31-83c5-16479117caf7.png
inflating: val/ya/ya_original_ya73.mine3.png_f4940887-8078-476e-981b-a33374855bf2.png
inflating: val/ya/ya_original_ya73.mine3.png_f53b2b9b-720f-4f29-822d-45c973ec1f1a.png
inflating: val/ya/ya_original_ya73.mine3.r10.png_00845043-a797-4c54-a469-505eae0d236.png
inflating: val/ya/ya_original_ya73.mine3.r10.png_343aee18-3624-4f6c-adf9-4166d87d85ed.png
inflating: val/ya/ya_original_ya73.mine3.r10.png_48f7935c-d93d-493e-b8d2-4fba63c98244.png
inflating: val/ya/ya_original_ya73.mine3.r10.png_5ed7f956-8456-4e4d-9997-8e8d192fbf38.png
inflating: val/ya/ya_original_ya73.mine3.r10.png_b10b0322-3226-468c-b305-0aba2a5a69bc.png
inflating: val/ya/ya_original_ya73.mine3.r10.png_d30d54ff-bd7a-41d0-aa5b-c80fc97b9274.png
inflating: val/ya/ya_original_ya73.mine3.r5.png_244a5aaf-e03b-46f5-a6ab-b9a9f1587708.png
inflating: val/ya/ya_original_ya73.mine3.r5.png_5dc84dfe-72e6-4697-baa1-dbc384321d23.png
inflating: val/ya/ya_original_ya73.mine3.r5.png_66c67466-c6f2-4550-8492-a49a4bf8a9db.png
inflating: val/ya/ya_original_ya73.mine3.r5.png_7965b8fe-4566-4e1d-809f-8ca683dadad0.png
inflating: val/ya/ya_original_ya73.mine3.r5.png_851f0773-a30d-40ch-hr7h-8h1258178h5d.png
```

Gambar diatas melakukan unzip data yang sudah di upload sebelumnya

4. Direktori untuk Data yang sudah di unzip

```
train_dir = '/content/train'
val_dir = '/content/val'
```

Gambar diatas memuat jalur file di dalam variabel bernama train_dir dan val_dir

5. Rescale Data

```
# Data augmentation validation data
validation_datagen = ImageDataGenerator(
    rescale=1./255
)
```

rescale=1./255: Normalisasi gambar dengan mengubah skala piksel dari 0-255 menjadi 0-1. Rescale dilakukan pada data train dan validation

6. Preprocessing

```
# Generator data training
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(64, 64),
    color_mode='grayscale',
    batch_size=32,
    class_mode='categorical',
    shuffle=True
)

validation_generator = validation_datagen.flow_from_directory(
    val_dir,
    target_size=(64, 64),
    color_mode='grayscale',
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)
```

A. Training Data

- 1) `train_dir`: Direktori yang berisi data training.
- 2) `target_size=(64, 64)`: Mengubah ukuran gambar menjadi 64x64 piksel.
- 3) `color_mode='grayscale'`: Mengubah gambar menjadi hitam putih.
- 4) `batch_size=32`: Menentukan ukuran batch, yaitu jumlah gambar yang akan diproses dalam satu iterasi.
- 5) `class_mode='categorical'`: Menentukan bahwa ini adalah masalah klasifikasi kategori.
- 6) `shuffle=True`: Mengacak urutan gambar dalam setiap epoch untuk memastikan model tidak menghafal urutan gambar.

B. Validation Data

- 1) `val_dir`: Direktori yang berisi data validasi.
- 2) `target_size=(64, 64)`: Mengubah ukuran gambar menjadi 64x64 piksel.
- 3) `color_mode='grayscale'`: Mengubah gambar menjadi hitam putih.
- 4) `batch_size=32`: Menentukan ukuran batch, yaitu jumlah gambar yang akan diproses dalam satu iterasi.
- 5) `class_mode='categorical'`: Menentukan bahwa ini adalah masalah klasifikasi kategori.
- 6) `shuffle=False`: Tidak mengacak urutan gambar dalam validasi untuk menjaga konsistensi evaluasi.

7. Labeling

```
[8] labels = train_generator.class_indices
    print(labels)

{'ba': 0, 'ca': 1, 'da': 2, 'dha': 3, 'ga': 4, 'ha': 5, 'ja': 6, 'ka': 7, 'la': 8, 'ma': 9, 'na': 10, 'nga': 11, 'nya': 12, 'pa': 13, 'ra': 14, 'sa': 15, 'ta': 16, 'tha': 17, 'wa': 18, 'ya': 19}
```

Labeling data menjadi 0 – 19 (20 label) menurut sub direktori

8. Plot Data Train

```
import os


def plot_gambar_asli_train(data_dir, labels):
    num_classes = len(labels)
    fig, axes = plt.subplots(1, num_classes, figsize=(20, 20))
    axes = axes.flatten()

    for idx, (class_name, class_index) in enumerate(labels.items()):
        image_dir = os.path.join(data_dir, class_name)
        image_path = os.path.join(image_dir, os.listdir(image_dir)[5])
        image = plt.imread(image_path)

        axes[idx].imshow(image)
        axes[idx].set_title(class_name)
        axes[idx].axis('off')

    plt.tight_layout()
    plt.show()

plot_gambar_asli_train(train_dir, labels)
```



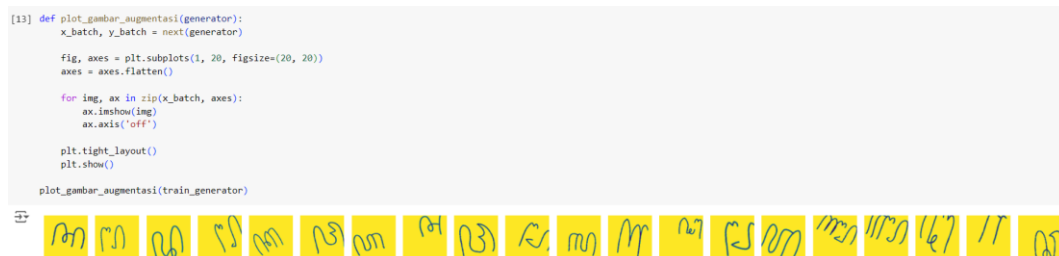
Kode diatas adalah plot gambar asli data train, setiap data diambil di index ke 5 atau gambar ke 6.

9. Plot Data Val



Kode diatas adalah plot gambar asli data val, setiap data diambil di index ke 5 atau gambar ke 6.

10. Plot Gambar Augmentasi



Kode diatas adalah plot gambar augmentasi, memunculkan 20 gambar

11. Callback

```
class CustomCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
        if logs.get('accuracy') >= 0.97 and logs.get('val_accuracy') >= 0.96:
            print("\nTraining stopped as accuracy and val_accuracy have reached 97% and 96%")
            self.model.stop_training = True
```

Penggunaan callback dimaksudkan untuk menghentikan pelatihan Ketika sudah mencapai target, kode diatas menggunakan custom callback dengan pengaturan pelatihan berhenti Ketika mencapai 97% pada akurasi training dan 96% pada validation akurasi.

12. CNN Model

```
def create_cnn(input_shape=(64, 64, 1), num_classes=20):
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(128, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Dropout(0.5))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(num_classes, activation='softmax'))

    return model

cnn = create_cnn()

cnn.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

cnn.summary()
```

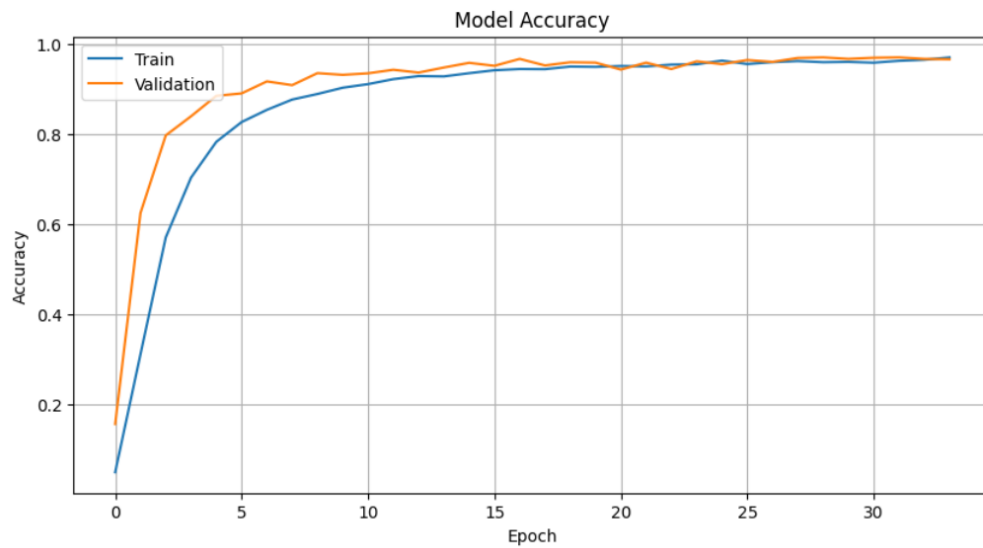
Model Convolutional Neural Network (CNN) yang Anda buat terdiri dari total 8 hidden layers, yang meliputi 3 layer konvolusi (Convolutional), 2 layer dense, dan 3 layer dropout. Arsitektur model dimulai dengan tiga lapisan konvolusi berturut-turut yang diikuti oleh lapisan pooling untuk pengurangan dimensi, kemudian diikuti oleh lapisan dropout untuk mengurangi overfitting. Setelah itu, model mengubah output menjadi satu dimensi dengan lapisan flatten, dilanjutkan dengan dua lapisan dense dengan aktivasi 'relu' dan lapisan dropout di antara mereka. Lapisan output menggunakan fungsi aktivasi 'softmax' untuk menghasilkan probabilitas dari 20 kelas yang berbeda. Model ini kemudian dikompilasi menggunakan optimizer 'adam' dan fungsi loss 'categorical_crossentropy', dengan metrik akurasi untuk memantau kinerja selama pelatihan.

13. Training Model

```
Epoch 34/50
263/263 [=====] - ETA: 0s - loss: 0.0948 - accuracy: 0.9707
Training stopped as accuracy and val_accuracy have reached 97% and 96%
263/263 [=====] - 71s 271ms/step - loss: 0.0948 - accuracy: 0.9707 - val_loss: 0.1728 - val_accuracy: 0.9667
```

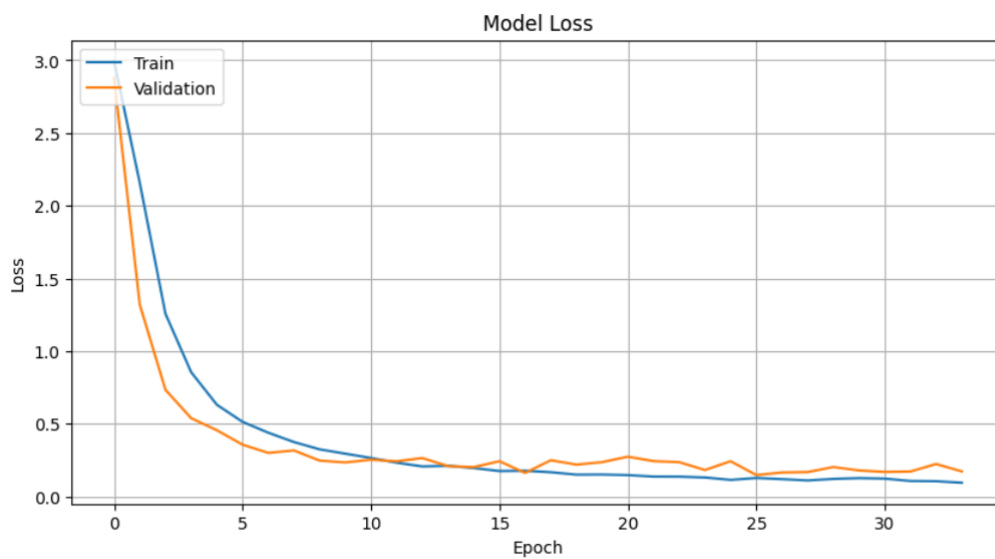
Model berhenti melatih pada epoch ke 34, karena telah mencapai 97% pada training akurasi dan 96% pada validation akurasi.

14. Plot akurasi model



Dari akurasi model dapat dilihat bahwa tidak terjadi overfitting maupun underfitting.

15. Plot loss model



Dari plot dapat dilihat bahwa loss semakin turun seiring berjalannya waktu menunjukkan performa yang cukup baik pada model.

16. Test pada huruf NA

```
na_original_na88.mine3.png_2d84f529-7a9a-4293-ad9a-4858553b41ef (4).png
na
1/1 [=====] - 8s 47ms/step
Predicted: na
```



```
na_original_na88.mine3.png_51bd5da7-418d-4b43-a31e-885d5c99b840 (4).png
na
1/1 [=====] - 8s 39ms/step
Predicted: da
```



```
na_original_na88.mine3.png_81ceF463-26da-40b3-a570-15f74c5a430f (4).png
da
1/1 [=====] - 8s 34ms/step
Predicted: na
```



```
na_original_na88.mine3.r5.png_27c7d08e-f1a0-4491-9abb-74a5bf215f13.png
na
1/1 [=====] - 8s 28ms/step
Predicted: na
```



Pada test dapat dilihat bahwa model dapat mendeteksi gambar dengan baik dan hanya terdapat 1 gambar yang salah prediksi karena akurasi model memang tidak 100%, jadi pasti ada beberapa yang salah deteksi

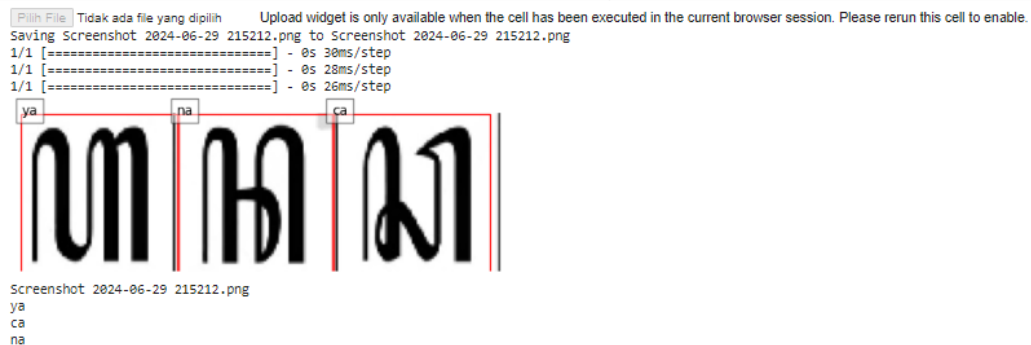
17. Test pada tulisan tangan

```
Pilih File Tidak ada file yang dipilih Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving 4693439601225045062_base64_38.png to 4693439601225045062_base64_38.png
1/1 [=====] - 0s 23ms/step
Predicted: nya
```



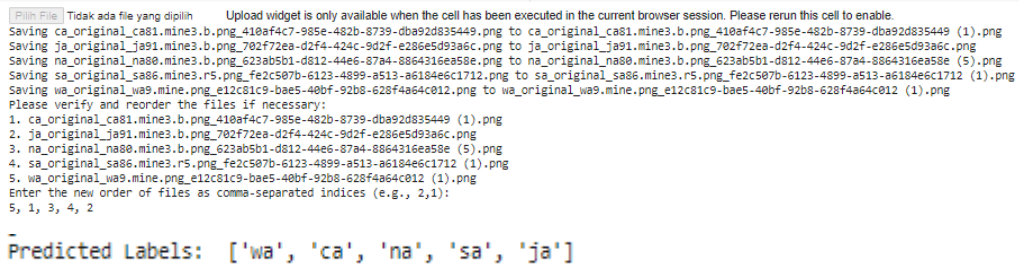
Pada tulisan tangan tersebut model mampu mendeteksi huruf dengan baik yaitu huruf NYA.

18. Test pada multiple image



Dapat dilihat dari test bahwa gambar dapat mendeteksi 2 huruf dengan benar yaitu NA dan CA, sedangkan pada huruf HA model salah prediksi menjadi YA mungkin karena kedua huruf tersebut hampir mirip.

19. Test pada gambar yang berbeda



Test dilakukan pada gambar 5 huruf yang berbeda yaitu huruf WA, CA, NA, SA, JA. Dari test tersebut model mampu memprediksi kelima gambar huruf dengan tepat.