

FINAL PROJECT

MATA KULIAH PENGANTAR PEMROSESAN DATA MULTIMEDIA



DISUSUN OLEH:

Ananda Putra	2108561001
Ni Putu Suci Paramita	2108561011
Hammam Akmal Prathama	2108561016

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

2023

BAB I

PENDAHULUAN

1.1 Latar Belakang

Image preprocessing adalah serangkaian teknik yang digunakan untuk mempersiapkan data gambar sebelum diolah oleh algoritma machine learning [6]. Tahap preprocessing sangat penting dalam proses training model, karena dapat membantu meningkatkan akurasi dan efisiensi algoritma. Salah satu algoritma yang sering digunakan dalam pengenalan pola pada data gambar adalah algoritma k-Nearest Neighbor (K-NN).

Algoritma K-NN merupakan salah satu algoritma machine learning yang sederhana dan populer dalam klasifikasi data. Algoritma ini bekerja dengan cara mencari k data terdekat dalam ruang fitur berdasarkan jarak Euclidean atau jarak lainnya, lalu melakukan mayoritas pemungutan suara untuk menentukan label kelas dari data yang sedang diprediksi. Dalam konteks pengenalan pola pada data gambar, K-NN dapat digunakan untuk mengklasifikasikan gambar berdasarkan fitur-fitur yang dihasilkan dari preprocessing [8].

Tahap preprocessing dalam penggunaan algoritma K-NN pada data gambar melibatkan beberapa langkah penting. Berikut adalah beberapa langkah umum dalam image preprocessing sebelum menggunakan algoritma K-NN:

1. **Resolusi dan Ukuran:** Gambar dalam dataset dapat memiliki resolusi dan ukuran yang berbeda-beda. Langkah pertama adalah menyamakan ukuran dan resolusi gambar agar memiliki dimensi yang seragam. Hal ini dapat dilakukan dengan melakukan resizing, cropping, atau padding pada gambar.
2. **Normalisasi:** Normalisasi gambar adalah proses mengubah intensitas piksel agar memenuhi rentang yang diinginkan. Misalnya, dapat dilakukan normalisasi dengan mengubah rentang intensitas piksel menjadi antara 0 hingga 1 atau -1 hingga 1. Hal ini dilakukan untuk memudahkan perhitungan jarak antar piksel pada algoritma K-NN.
3. **Pemrosesan Warna:** Jika gambar berwarna, maka dapat dilakukan pemrosesan warna untuk mengubah gambar menjadi skala abu-abu atau mengambil saluran warna tertentu yang lebih relevan untuk klasifikasi.

Pemrosesan warna juga dapat melibatkan ekstraksi fitur warna seperti histogram warna atau metode ekstraksi ciri lainnya.

4. Ekstraksi Fitur: Ekstraksi fitur adalah langkah penting dalam preprocessing gambar untuk klasifikasi menggunakan algoritma K-NN. Fitur-fitur yang relevan diekstraksi dari gambar untuk mewakili karakteristik unik yang membedakan kelas. Contoh metode ekstraksi fitur yang umum digunakan adalah metode ekstraksi ciri tekstur, seperti Local Binary Patterns (LBP) atau metode ekstraksi ciri bentuk seperti Histogram of Oriented Gradients (HOG) serta Gray-Level Co-occurrence Matrix (GLCM).
5. Reduksi Dimensi: Dalam beberapa kasus, gambar memiliki dimensi fitur yang besar. Reduksi dimensi dapat dilakukan untuk mengurangi kompleksitas perhitungan dan mempercepat proses training.

Setelah langkah-langkah preprocessing di atas selesai, data gambar siap digunakan untuk training algoritma K-NN. Dalam tahap training, K-NN akan menghitung jarak antara fitur-fitur gambar yang telah diekstraksi dengan fitur-fitur gambar pada dataset latihan. Kemudian, algoritma akan menentukan kelas yang paling sering muncul dari k data terdekat dan menggunakannya sebagai prediksi untuk gambar yang sedang diproses.

Dengan melakukan langkah-langkah preprocessing yang tepat, algoritma K-NN dapat memberikan hasil klasifikasi yang lebih baik pada data gambar. Preprocessing membantu dalam mengurangi keberisikan, menormalisasi data, dan mengekstraksi fitur yang relevan, sehingga meningkatkan kemampuan algoritma untuk memahami dan mengklasifikasikan gambar dengan lebih baik.

1.2 Problem Komputasi

1. Feature Extraction dengan metode berbasis tekstur menggunakan metode GLCM (Gray-Level Co-occurrence Matrix) untuk menentukan 6 nilai fitur dari data image: dissimilarity, correlation, homogeneity, contrast, ASM, energy, dengan 5 sudut (0, 45, 90, 135, dan 180). Total ada 30 nilai fitur (6 x 5) untuk setiap data image.

2. Training dilaksanakan untuk menghasilkan model klasifikasi yang terbaik. Untuk metode KNN training dilakukan dengan mencoba beberapa nilai k yang ganjil (contoh $k=3$, atau 5, atau 7, atau 9).
3. Model yang dihasilkan di deploy ke sistem aplikasi berbasis web dengan fitur utama adalah user menginput satu atau beberapa data dan outputnya adalah hasil sentimen atau identifikasi emosi.

1.3 Tujuan

Berikut tujuan dari penugasan final project

1. Untuk mengidentifikasi emosi dari sebuah citra ekspresi wajah dengan 2 sentimen yaitu happy dan sad.
2. Untuk mendapatkan hasil dari tahap image preprocessing dengan ekstraksi fitur menggunakan metode GLCM (Gray-Level Co-occurrence Matrix).
3. Untuk metode KNN training dilakukan dengan mencoba beberapa nilai k yang ganjil (contoh $k=3$, atau 5, atau 7, atau 9).

1.4 Manfaat

Adapun manfaat yang didapatkan dari penugasan final project ini

1. Mendapatkan hasil identifikasi dari sebuah gambar (sad atau happy).
2. Untuk mendapatkan hasil dari tahap image preprocessing dengan ekstraksi fitur menggunakan metode GLCM (Gray-Level Co-occurrence Matrix).
3. Mendapatkan hasil dari akurasi training machine learning dengan algoritma K-NN.

BAB II

ISI

1.1 Image Processing

Image processing adalah metode pengolahan atau manipulasi gambar dua dimensi [2]. Image processing bertujuan untuk memperbaiki kesalahan data sinyal citra yang terjadi akibat transmisi dan deteksi sinyal, serta memperbaiki tampilan citra sehingga dapat diinterpretasikan oleh sistem visual manusia melalui manipulasi dan lebih mudah. Citra grayscale merupakan gambar yang menggunakan pada tingkatan warna abu – abu [1]. Warna Abu-abu adalah satu-satunya warna dalam ruang RGB di mana komponen merah, hijau, dan biru memiliki intensitas yang sama. Untuk gambar grayscale, hanya menentukan nilai intensitas untuk setiap piksel sebagai satu nilai, sedangkan untuk gambar berwarna, Anda menentukan tiga nilai intensitas untuk setiap piksel. Intensitas gambar skala abu-abu disimpan sebagai bilangan bulat 8-bit dan menawarkan 256 kemungkinan dari level 0 hingga 255 (0 untuk hitam dan 255 untuk putih, di antaranya adalah level abu-abu).

2.2 GLCM

Grayscale Co-Occurrence Matrix (GLCM) adalah metode analisis tekstur/ekstraksi fitur. GLCM adalah matriks yang menggambarkan frekuensi pasangan dua piksel dengan intensitas tertentu pada jarak dan arah tertentu dalam gambar [7]. Prinsip dasar GLCM adalah menghitung frekuensi pasangan nilai intensitas piksel yang terjadi pada jarak dan arah tertentu pada citra grayscale. Dengan membangun matriks koeksistensi, kita dapat menganalisis hubungan distribusi dan kejadian dari sepasang nilai intensitas yang memberikan informasi tentang tekstur citra.

2.3 KNN

K-Nearest Neighbor (K-NN) merupakan salah satu metode klasifikasi yang tersedia dalam penambangan data dan termasuk dalam kelompok pembelajaran berbasis instan. KNN dilakukan dengan mencari k objek pada data training yang paling dekat (mirip) dengan objek pada data test [4]. KNN merupakan algoritma supervised learning dimana hasil dari query

instance yang baru diklasifikan berdasarkan mayoritas dari kategori pada algoritma KNN. Dimana kelas yang paling banyak muncul yang nantinya akan menjadi kelas hasil dari klasifikasi [5].

2.4 Implementasi Program

Tahapan Processing Image dengan metode berbasis tekstur menggunakan metode GLCM untuk menentukan 6 nilai fitur dari data image: dissimilarity, correlation, homogeneity, contrast, ASM, energy, dengan 5 sudut (0, 45, 90, 135, dan 180). Total ada 30 nilai fitur (6 x 5) untuk setiap data image. Kami menggunakan parameter nilai k yang ganjil (k=3, atau 5, atau 7, atau 9).

1. Import Library

```
import os
import glob
import numpy as np
from skimage.feature import graycomatrix, graycoprops
import cv2
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
import joblib
import streamlit as st
```

2. Mendefinisikan beberapa variable data agar lebih mudah di proses

```
# Global variables
happy_dir = 'happy'
sad_dir = 'sad'
dataset_file = 'dataset.npy'
model_file = 'model.joblib'
```

3. Menghitung fitur GLCM pada sebuah gambar

```
# Calculate GLCM features
def calculate_glcml_features(img):
    angles = [0, np.pi/4, np.pi/2, 3*np.pi/4, np.pi]
```

```

        properties = ['dissimilarity', 'correlation', 'homogeneity',
        'contrast', 'ASM', 'energy']

        glcm = graycomatrix(img, [1], angles, 256, symmetric=True,
        normed=True)

        features = []

        for prop in properties:
            feature = graycoprops(glcm, prop).ravel()
            features.append(feature)

            st.write(f'{prop}: {feature}')

    return np.concatenate(features)

```

4. Untuk melatih dataset gambar atau label yang digunakan

```

# Train the model
def train_model(k):
    st.write("Training Model...")

    # Load image files and labels
    happy_files = glob.glob(os.path.join(happy_dir, '*'))
    sad_files = glob.glob(os.path.join(sad_dir, '*'))
    files = happy_files + sad_files
    labels = [0] * len(happy_files) + [1] * len(sad_files) # 0 for
    happy, 1 for sad

```

5. Memproses gambar dan menghasilkan fitur GLCM serta melatih model KNN dan menyimpan model serta dataset yang telah diproses.

```

# Preprocess images and extract GLCM features
features = [calculate_glcm_features(cv2.resize(cv2.imread(file,
0), (48, 48))) for file in files]

# Ensure features and labels are of the same size

```

```

features, labels = np.array(features), np.array(labels)
assert features.shape[0] == labels.shape[0], "Mismatch in
features and labels sizes"

# Train KNN classifier
X_train, X_test, y_train, y_test = train_test_split(features,
labels, test_size=0.2, random_state=42)
model = KNeighborsClassifier(n_neighbors=k)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = metrics.accuracy_score(y_test, y_pred)
precision = metrics.precision_score(y_test, y_pred)
recall = metrics.recall_score(y_test, y_pred)
f1_score = metrics.f1_score(y_test, y_pred)

st.write(f"Model accuracy: {accuracy*100:.2f}%")
st.write(f"Precision: {precision:.2f}")
st.write(f"Recall: {recall:.2f}")
st.write(f"F1-Score: {f1_score:.2f}")

# Save the model
joblib.dump(model, model_file)
st.write("Model saved!")

# Save the dataset
dataset = np.column_stack((features, labels))
np.save(dataset_file, dataset)
st.write("Dataset saved!")

```

6. Untuk menampilkan Halaman Utama

```

# Streamlit UI - Home Page
def home():
    st.title('Happy or Sad Image Classifier')

    # Check if model and dataset files exist

```



```

model_exists = os.path.exists(model_file)
dataset_exists = os.path.exists(dataset_file)

# Add "Train Model" button
if st.button("Train Model"):
    st.session_state.page = 'Train'

# Image classification
file = st.file_uploader("Upload an image")
col1, col2 = st.columns(2)
if file is not None:
    # Preprocess and extract GLCM features
    file_bytes = np.asarray(bytearray(file.read()), dtype=np.uint8)
    img = cv2.imdecode(file_bytes,
cv2.IMREAD_GRAYSCALE)
    img = cv2.resize(img, (48, 48))
    features = calculate_glcm_features(img)

    col1.image(img, caption="Uploaded Image",
use_column_width=False)

    if model_exists:
        # Load the model and predict
        model = joblib.load(model_file)
        prediction = model.predict([features])[0]
        if prediction == 0:
            col2.write("The image is happy!")
        else:
            col2.write("The image is sad!")

# Calculate metrics
if dataset_exists:
    dataset = np.load(dataset_file)
    X_test, y_test = dataset[:, :-1], dataset[:, -1]
    y_pred = model.predict(X_test)

```

```

accuracy = metrics.accuracy_score(y_test, y_pred)
precision = metrics.precision_score(y_test, y_pred)
recall = metrics.recall_score(y_test, y_pred)
f1_score = metrics.f1_score(y_test, y_pred)
st.write(f'Accuracy: {accuracy*100:.2f}%')
st.write(f'Precision: {precision:.2f}')
st.write(f'Recall: {recall:.2f}')
st.write(f'F1-Score: {f1_score:.2f}')

```

7. Untuk menampilkan halaman proses pada nilai K yang digunakan

```

# Streamlit UI - Train Page
def train():
    st.title('Train Model')

    # Add "Back" button
    if st.button("Back"):
        st.session_state.page = 'Home'

    # Select K value
    k = st.selectbox("Select K value", [1, 3, 5, 7, 9])

    # Train the model
    if st.button("Train", key="train_button"):
        train_model(k)

# Streamlit App
def main():
    # Initialize Streamlit session state
    if 'page' not in st.session_state:
        st.session_state.page = 'Home'

    # Page navigation
    if st.session_state.page == 'Home':
        home()

```

```

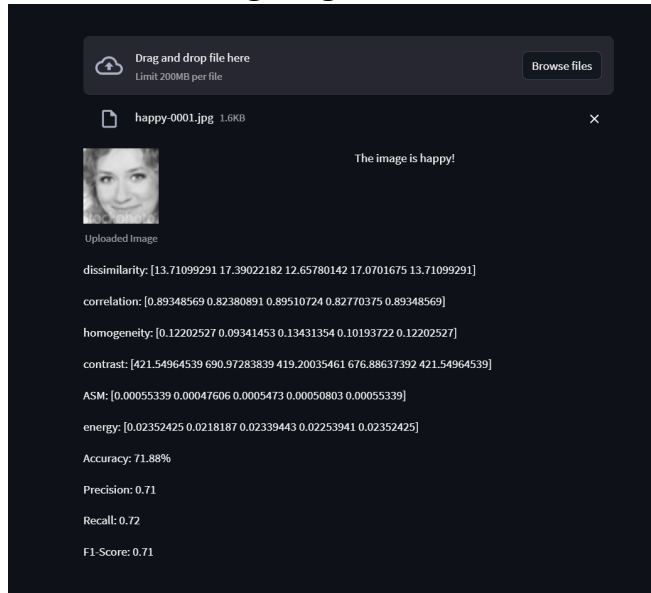
elif st.session_state.page == 'Train':

    train()

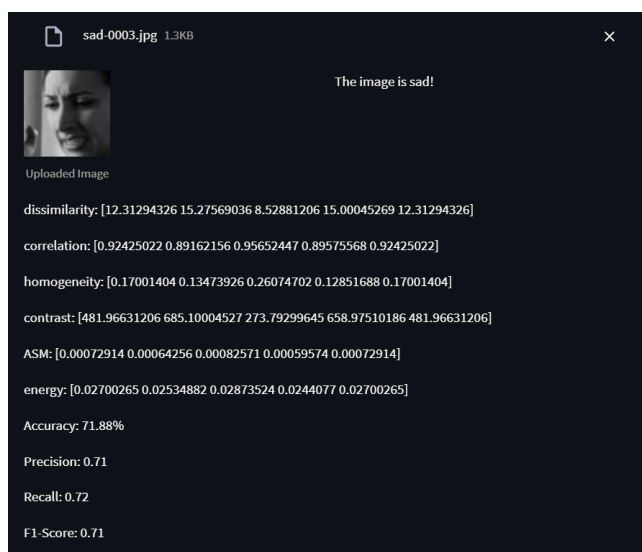
if __name__ == "__main__":
    main()

```

2.5 Hasil Running Program



Gambar 1 Menggunakan Nilai Parameter $K = 3$



Gambar 2 Menggunakan nilai parameter $K = 3$

BAB III

PENUTUP

Dengan melakukan langkah-langkah preprocessing menggunakan, algoritma K-NN dapat memberikan hasil klasifikasi yang lebih baik pada data gambar. Preprocessing membantu dalam mengurangi keberisikan, menormalisasi data, dan mengekstraksi fitur yang relevan, sehingga meningkatkan kemampuan algoritma untuk memahami dan mengklasifikasikan gambar dengan lebih baik. Dari 3 nilai parameter yang digunakan, hasil yang didapatkan adalah nilai parameter $K = 3$ memiliki akurasi yang paling tinggi sekitar 71.88%.

DAFTAR PUSTAKA

- [1] d. Sutayo, *Teori Pengolahan Citra Digital*. Andi. Yogyakarta, 2009.
- [2] A. Usman, *Pengolahan Citra Digital & Teknik Pemrogramannya*, 2005.
- [3] S. Indra, *Simulasi Pengaturan Lalu lintas Berdasarkan Data Image Processing Kepadatan Kendaraan Berbasis Mikrokontroler Atmega 16* , 2014.
- [4] X. &. K. Wu, *The Top Ten Algorithms in Data Mining*, 2009.
- [5] B. Avelita, *Klasifikasi_K-Nearest_Neighbor*, 2016.
- [6] AnbiDev, *Belahar Machine Learning - Image Processing* , 2020.
- [7] A. W. W. A. S. Restu Widodo, *Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (Citrus reticulata Blanco) untuk Klasifikasi Mutu*, vol. 2, 2018.
- [8] Trivusi, *Yuk Kenali Apa itu Algoritma K-Nearest Neighbors (KNN)* , 2022.