IDENTIFIKASI EMOSI CITRA EKSPRESI WAJAH ORANG MENGGUNAKAN METODE KNN

FINAL PROJECT PPDM

I DEWA GEDE PARTHA WIJAYA	**NIM. 2108561093**
I MADE SUMA GUNAWAN	**NIM. 2108561108**
AKHMAD FAJRI MUBARAK	**NIM. 2108561113**

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

JIMBARAN

2023

BAB I
PENDAHULUAN

1.1 Latar Belakang

Identifikasi emosi dari gambar wajah manusia merupakan penelitian menarik dalam pengenalan pola dan kecerdasan buatan. Ekspresi wajah manusia dapat mengungkapkan berbagai emosi seperti senyum, kesedihan, kemarahan, dan kejutan. Pengenalan dan identifikasi emosi berdasarkan ekspresi wajah memiliki banyak aplikasi, termasuk dalam interaksi manusia dan mesin, pengawasan keamanan, serta pengembangan antarmuka pengguna responsif.

Metode K-Nearest Neighbors (KNN) digunakan dalam identifikasi emosi dari citra ekspresi wajah. KNN adalah metode klasifikasi yang memanfaatkan kesamaan atribut untuk mengelompokkan data ke dalam kategori yang sama. Dalam konteks identifikasi emosi, KNN digunakan untuk memprediksi emosi berdasarkan kemiripan fitur dengan data latih yang memiliki label emosi.

Penerapan metode KNN dalam sistem identifikasi emosi citra ekspresi wajah melibatkan langkah-langkah seperti pengumpulan dan praproses data, ekstraksi fitur wajah, pembentukan set data latih, pemilihan parameter k, pelatihan model, serta klasifikasi dan identifikasi emosi. Proses ini memungkinkan sistem untuk mengklasifikasikan citra wajah yang tidak dikenal dan mengidentifikasi emosi dominan dengan mencari tetangga terdekat dalam ruang atribut dan menggunakan mayoritas suara.

Metode KNN memiliki potensi sebagai alat efektif dalam identifikasi emosi citra ekspresi wajah manusia. Penggunaannya dapat memberikan manfaat dalam berbagai bidang, seperti pengenalan emosi dalam interaksi manusia dan mesin, pengawasan keamanan, serta pengembangan antarmuka pengguna yang responsif.

1.2 Rumusan Masalah

Adapun rumusan masalah dari laporan ini adalah:

1. Bagaimana membangun sistem untuk identifikasi emosi ekspresi wajah orang menggunakan knn.
2. Bagaimana tingkat kompleksitas dari sistem yang dibuat.

1.3 Tujuan

Adapun tujuan dari pembuatan sistem ini adalah:

1. Mengembangkan alat yang dapat mengidentifikasi emosi dari citra ekspresi wajah orang.
2. Meningkatkan pemahaman tentang pengenalan emosi melalui ekspresi wajah.
3. Meningkatkan kinerja dan akurasi identifikasi emosi.

1.4 Batasan Masalah

Batasan masalah pada sistem ini adalah:

1. Menginplementasikan metode KNN sebagai metode untuk membangun sistem identifikasi emosi citra ekspresi wajah orang.
2. Sistem ini hanya berfokus pada identifikasi dan klasisfikasi emosi yang terkandung dalam citra ekspresi wajah manusia.
3. Sistem ini dirancang untuk mengidentifikasi emosi pada wajah manusia.

1.5 Manfaat

Adapun manfaat yang akan dicapai pada laporan ini adalah sebagai berikut:

1. Bagi Keilmuan

   Laporan ini diharapkan dapat memberikan referensi kepada peneliti lain yang akan membangun sebuah sistem identifikasi emosi citra menggunakan KNN.
2. Bagi Mahasiswa

   Laporan ini diharapkan dapat memberikan referensi dalam membangun sebuah sistem identifikasi emosi citra menggunakan KNN.
3. Bagi Peneliti

   Laporan ini diharapkan dapat menambah pengetahuan untuk menginplementasikan metode knn dalam identifikasi emosi pada citra ekspresi wajah orang.

BAB II
ISI

2.1 Image Preprocessing

Image Preprocessing merupakan salah satu tahap yang digunakan dalam pengolahan dataset guna memastikan keberhasilan pelatihan mesin. Pada tahap ini, data gambar awal diolah sedemikian rupa sehingga dapat diubah menjadi sekelompok array yang dapat dipahami oleh mesin pembelajaran.

Pemrosesan awal gambar melibatkan serangkaian langkah yang diambil untuk mengatur ulang gambar sebelum digunakan dalam pelatihan model dan inferensi. Langkah-langkah ini meliputi, tetapi tidak terbatas pada, mengubah ukuran, mengubah orientasi, dan koreksi warna.

Penggunaan pemrosesan awal gambar juga memiliki manfaat dalam mengurangi waktu yang dibutuhkan untuk melatih model serta meningkatkan kecepatan inferensi model. Jika ukuran gambar masukan sangat besar, mengurangi ukuran gambar tersebut secara signifikan akan meningkatkan waktu pelatihan model tanpa mengorbankan kinerja model secara substansial. Sebagai contoh, ukuran gambar standar pada iPhone 11 adalah 3024 × 4032 piksel. Untuk model pembelajaran mesin yang digunakan oleh Apple dalam membuat topeng dan menerapkan Mode Potret, gambar tersebut diolah dengan ukuran separuhnya sebelum hasilnya diubah skala kembali ke ukuran penuh.

2.2 GLCM

Menurut penelitian Xie et al. (2010), metode GLCM adalah suatu pendekatan yang menganalisis piksel dalam citra dan mengidentifikasi tingkat keabuan yang sering terjadi. Metode ini juga digunakan untuk tabulasi frekuensi kombinasi nilai piksel yang muncul dalam citra tersebut. Dalam melakukan analisis citra berdasarkan distribusi statistik dari intensitas pikselnya, fitur tekstur dapat diekstraksi (Pullaperuma & Dharmaratne, 2013).

GLCM merupakan metode yang digunakan untuk mengekstraksi fitur berbasis statistik, dengan mengambil nilai piksel dari matriks yang memiliki nilai khusus dan membentuk pola sudut tertentu (Kasim & Harjoko, 2014; Xie et al.,

2010). Sudut yang digunakan dalam GLCM untuk nilai piksel citra adalah 0°, 45°, 90°, dan 135° (Eleyan & Demirel, 2011).

2.3 K-Nearest Neighbor

Algoritma K-Nearest Neighbor (KNN) adalah suatu metode klasifikasi yang mengukur jarak antara data uji dengan data pelatihan. KNN bekerja dengan mencari sejumlah k data pelatihan yang memiliki jarak terdekat dengan data uji, lalu mengklasifikasikan kelas berdasarkan mayoritas kelas dari data terdekat tersebut (Suyanto, 2018). Jarak antara data dapat dihitung menggunakan rumus jarak seperti rumus jarak Euclidean dan jarak Minkowski, baik dalam kasus letak yang dekat maupun jauh. Dalam mengukur jarak antar data, rumus jarak Euclidean sering digunakan karena memiliki tingkat akurasi yang tinggi. Berikut rumus jarak Euclidean:

$$d(x_i, y_i) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Keterangan :

d = Jarak antar data

xi= sampel data

yi= data uji

i = Variabel data

n = Dimensi data

2.4 Source Code

2.4.1 Import Library

Pada gambar diatas adalah code untuk mengimportkan library untuk mempermudah pembangunan program.

2.4.2 Tahap Preprocessing

```python
# preprocessing
angles = [0, np.pi/4, np.pi/2, 3*np.pi/4, np.pi]
metric_texture = ['dissimilarity', 'correlation', 'homogeneity', 'contrast', 'ASM', 'energy']

def glcm_matrix(image):
    glcm = graycomatrix(image, distances=[1], angles=angles, levels=256, symmetric=True, normed=True)
    features = []
    for i in metric_texture:
        feature = []
        for j in angles:
            feature.append(graycoprops(glcm, prop=i)[0, 0])
        features.extend(feature)
    return np.array(features)

X = []  # Features
y = []  # Labels
```

[2] ✓ 0.0s Python

Selanjutnya adalah tahap preprocessing menggunakan glcm yang berguna dalam pemrosesan citra untuk mengekstraksi fitur tekstur, mengurangi noise, melakukan segmentasi, dan mendeteksi perubahan.

2.4.3 Tahap Training

```python
image_path = "dataset training/"

def load_images_and_extract_features(directory, label, X, y):
    image_paths = os.listdir(directory)
    for img_path in image_paths:
        image = io.imread(os.path.join(directory, img_path))
        features = glcm_matrix(image)
        X.append(features)
        y.append(label)

load_images_and_extract_features(f"{image_path}happy/", 1, X, y)    # Load citra dengan ekspresi happy
load_images_and_extract_features(f"{image_path}sad/", 0, X, y)      # Load citra dengan ekspresi sad
```

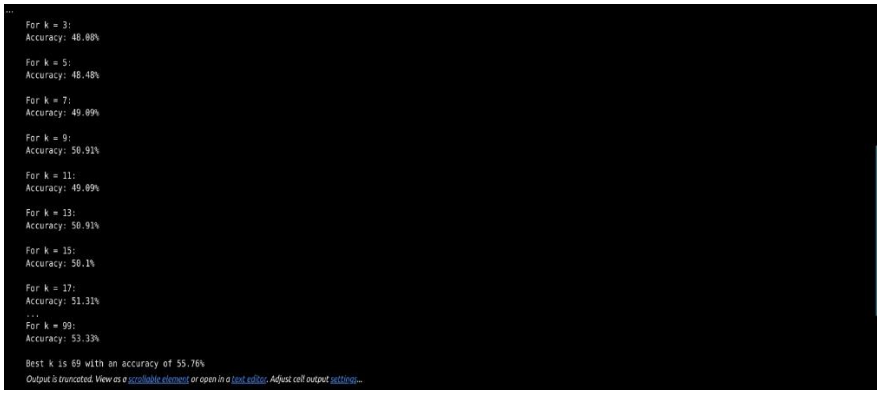
[3] ✓ 11m 4.3s Python

Setelah data diproses, langkah berikutnya adalah melatih model. Anda menggunakan model K-Nearest Neighbors (KNN), di mana Anda menyetel jumlah tetangga (n) dan melakukan fitting model dengan data latih.

2.4.4 Tahap Testing

```python
# Testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

def knn_predict_and_score(n):
    knn = KNeighborsClassifier(n_neighbors=n)              # Inisialisasi model KNN
    knn.fit(X_train, y_train)                              # Latih model
    accuracy_percentage = knn.score(X_test, y_test) * 100  # Evaluasi model
    return accuracy_percentage.round(2)

accuracy_scores = {}

# Coba nilai k dari 3 sampai 9 (hanya bilangan ganjil)
for k in range(3, 100, 2):
    accuracy = knn_predict_and_score(k)
    accuracy_scores[k] = accuracy
    print(f"\nFor k = {k}:")
    print(f"Accuracy: {accuracy}%")

# Temukan k yang memberikan akurasi tertinggi
best_k = max(accuracy_scores, key=accuracy_scores.get)
print(f"\nBest k is {best_k} with an accuracy of {accuracy_scores[best_k]}%")
```

[4] ✓ 15.1s Python



Sebelum model dilatih, data image dipisahkan menjadi 2, yaitu data training sebanyak 80% dan data testing sebanyak 20%. Data testing ada dengan tujuan untuk mengukur kinerjanya. Di sini, digunakan metode score dari model KNN untuk mengevaluasi akurasi model. Dilakukan juga percobaan dengan berbagai nilai k untuk melihat pengaruhnya terhadap kinerja model.

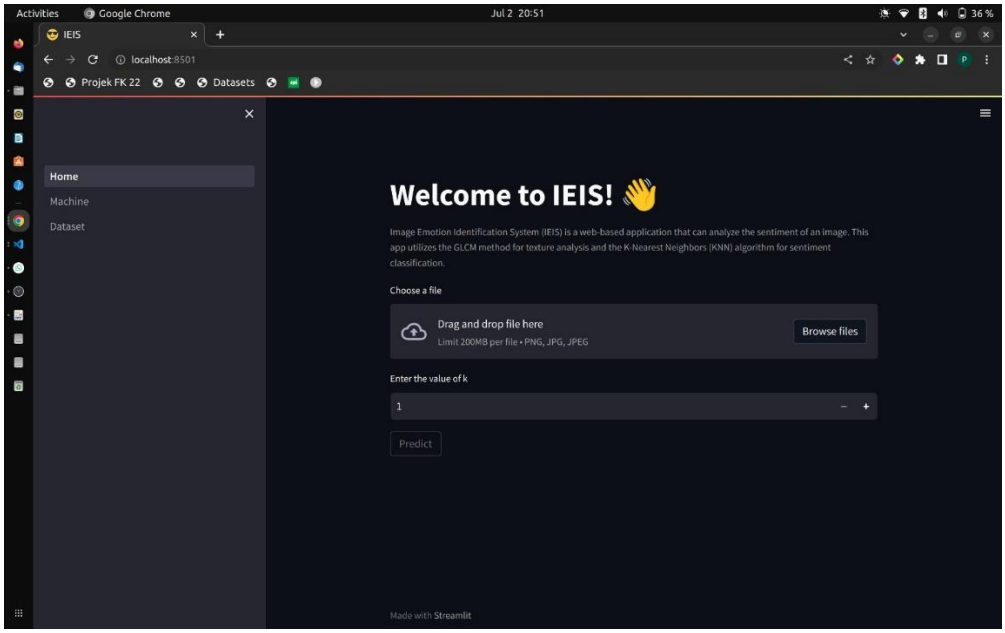
2.4.5 Tahap Visualisasi Deployment Website

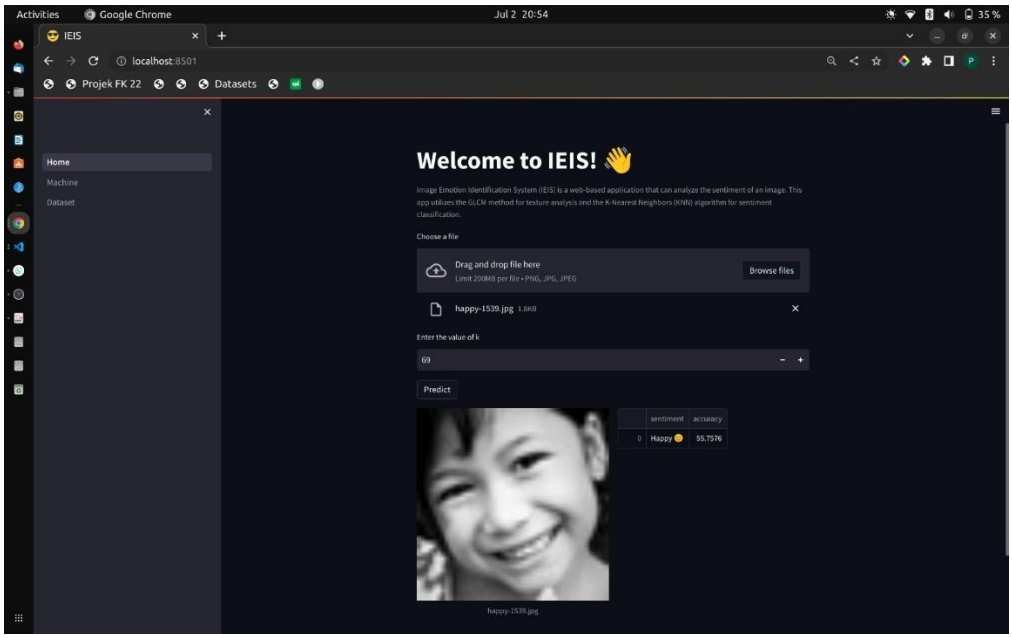
```python
import streamlit as st
import pandas as pd
from skimage import io, color, util, transform
from Backend import knn_predict, glcm_matrix

st.set_page_config(page_title="IEIS", page_icon="😎")
st.title("Welcome to IEIS! 👋")
st.caption("Image Emotion Identification System (IEIS) is a web-based application that can analyze the sentiment of an image. This app utilizes the GLCM method for texture analysis and the K-Nearest Neighbors (KNN) algorithm for sentiment classification.")

uploaded_file = st.file_uploader("Choose a file", type=['png', 'jpg', 'jpeg'])
k = st.number_input("Enter the value of k", min_value=1)

if uploaded_file is None:
    st.button("Predict", disabled=True)
else:
    if st.button("Predict"):
        image = io.imread(uploaded_file)

        if image.shape[0] > 48 or image.shape[1] > 48:
            image = transform.resize(image, (48, 48))

        if len(image.shape) == 3:  # Check if the image has color channels
            image = color.rgb2gray(image)

        image = util.img_as_ubyte(image)
        new_features = glcm_matrix(image)
        prediction, accuracy_percentage = knn_predict(new_features, k)

        col1, col2 = st.columns(2)
        with col1:
            st.image(image, caption=uploaded_file.name, use_column_width=True)

        with col2:
            sentiment = "Happy 😊" if prediction == 1 else "Sad 😔"
            data_df = pd.DataFrame({"sentiment": [sentiment], "accuracy": [accuracy_percentage]})
            st.dataframe(data_df)
```

Pada tahapan ini kami menggunakan streamlit untuk front-end sehingga memudahkan dalam membuat tampilan dari websitenya.

Sementara Backendnya adalah kode model KNN dari import, preprocessing, training, dan testing.

2.5 Implementasi





Dalam tahap implementasi, kami menggunakan Streamlit, sebuah kerangka kerja berbasis Python yang bersifat open-source dan diciptakan untuk mempermudah pembangunan aplikasi web. Pada situs web ini, kami hanya menerima masukan berupa file image dan keluarannya berupa hasil identifikasi emosi dari file gambar yang diunggah.

BAB III
PENUTUP

Dapat disimpulkan bahwa framework streamlit dapat digunakan untuk membangun sebuah website yang mampu mengidentifikasi emosi pada gambar. Pada bagian antarmuka pengguna, terdapat fitur untuk memasukkan gambar dan sebuah kolom untuk menampilkan hasil prediksi atau identifikasi emosi. Di sisi program, terdapat serangkaian fungsi dalam backend yang bertugas menganalisis tekstur menggunakan metode GLCM dan mengklasifikasikan data gambar dengan menggunakan algoritma KNN. Dengan demikian, pengguna dapat memeriksa gambar yang ingin diketahui jenis emosinya.

DAFTAR PUSTAKA

Hugo. (2022). Belajar Machine Learning - Image Processing. Diakses pada 28 Juni 2023 dari https://www.anbidev.com/machine-learning-image-processing/#:~:text=Image%20Preprocessing%20adalah%20salah%20satu,agar%20dapat%20dipelajari%20oleh%20mesin.

Nelson, J. (2020). What is Image Preprocessing and Augmentation? Diakses pada 28 Juni 2023 dari https://blog.roboflow.com/why-preprocess-augment/.

Ahmad, N., Alhamad, A. (2019). Penerapan Metode Glcm (Gray Level Co-Occurrence Matrix) Untuk Reduksi Ciri Pada Pengenalan Ekspresi Wajah. Jurnal Nasional cosPhi, Vol. 3, No. 2.