

KLASIFIKASI MICRO-EXPRESSION MENGGUNAKAN K-NEAREST NEIGHBORS MENGGUNAKAN FITUR CAS DAN HOG

Nikko R. P. Wardoyo¹, Joan Santoso¹, dan Esther I. Setiawan¹

¹Magister Teknologi Informasi, Institut Sains dan Teknologi Terpadu Surabaya Jl.Ngagel jaya tengah 73-77, Surabaya 60284, Indonesia

Corresponding author: Nikko Riestian Putra Wardoyo (nikko.r21@mhs.istts.ac.id).

ABSTRACT Micro-Expressions are expressions that appear in a short time, only lasting a fraction of a second. This may be a result of communication activities between humans during social interactions. Facial micro-expression reactions occur naturally and immediately, leaving little room for manipulation. However, because Micro-Expressions are transient and have low intensity, their recognition and recognition is difficult and relies heavily on the experience of experts. Due to their specificity and intrinsic complexity, the classification of Micro-Expressions using 2 extractions namely CAS and HOG is interesting but challenging, and recently- has recently become an active area of research. context-aware saliency (CAS) which aims to detect image regions that represent the scene. The goal is to detect the dominant object. Since this descriptor is extracted on a regular grid, it is powerful for face detection with changes in occlusion, pose, and lighting. The K-Nearest Neighbors (K-NN) method is used for Micro-Expression classification based on HOG features from saliency images. The dataset used in this research was 45 students from Ma'arif NU Prambon Vocational School, majoring in Multimedia, and added a dataset from affecnet. The results obtained from a total dataset of 4116 images which were divided into 6 Micro-Expressions, namely anger, disgust, fear, happy, sad and surprise, obtained accuracy results above 80% from a comparison of a dataset of 4116 divided into 2 with a percentage of 70% training and 30 % testing data.

KEYWORDS - CAS, HOG, K-NN, Micro-Expression.

ABSTRAK *Micro-Expression* adalah ekspresi yang muncul dalam waktu singkat, hanya berlangsung sepersekian detik. Hal ini mungkin merupakan akibat dari aktivitas komunikasi antar manusia selama interaksi sosial. Reaksi ekspresi mikro wajah terjadi secara alami dan segera, sehingga hanya menyisakan sedikit ruang untuk manipulasi. Namun, karena *Micro-Expression* bersifat sementara dan memiliki intensitas rendah, pengenalan dan pengenalannya sulit dan sangat bergantung pada pengalaman para ahli. Karena kekhususan dan kompleksitas intrinsiknya, klasifikasi *Micro-Expression* menggunakan 2 ekstraksi yaitu CAS dan HOG menarik tetapi menantang, dan baru-baru ini menjadi area penelitian yang aktif. *context-aware saliency* (CAS) yang bertujuan untuk mendeteksi wilayah gambar yang mewakili pemandangan. Tujuannya adalah untuk mendeteksi objek dominan. *Histogram Oriented Gradient* (HOG) Bertujuan sebagai deskriptor yang efektif untuk pengenalan dan deteksi objek. Metode *K-Nearest Neighbors* (K-NN) digunakan untuk klasifikasi *Micro-Expression* berdasarkan fitur HOG dari citra *saliency*. Dataset yang digunakan pada penelitian ini dari data sampel siswa SMK Ma'arif NU Prambon jurusan Multimedia sebanyak 45 siswa dan ditambahkan dataset dari affecnet. Hasil yang didapatkan dari total dataset sebanyak 4116 citra yang dibagi menjadi 6 *Micro-Expression* yaitu anger, disgust, fear, happy, sad dan surprise, mendapatkan hasil akurasi diatas 80% dari perbandingan dataset sejumlah 4116 terbagi menjadi 2 dengan persentase 70% training dan 30% data testing.

KATA KUNCI CAS, HOG, K-NN, *Micro-Expression*.

I. PENDAHULUAN

Emosi adalah bagian yang tidak terpisahkan dari kehidupan manusia, dan muncul secara sukarela atau tidak sengaja melalui ekspresi wajah ketika orang berkomunikasi satu sama lain secara tatap muka. Sebagai bentuk komunikasi nonverbal yang khas, ekspresi wajah memainkan peran penting dalam analisis emosi manusia [1]–[5], dan dengan demikian telah dipelajari secara luas di berbagai domain. Emosi juga adalah perasaan atau reaksi terhadap rangsangan yang mendorong orang untuk bertindak [6]. Emosi dapat dipahami dari ekspresi mikro seseorang, dan ekspresi mikro seseorang bersifat universal [7].

Ekspresi wajah adalah cara paling langsung untuk mengekspresikan emosi manusia, dan juga merupakan cara yang sangat penting untuk memahami niat manusia dalam komputasi yang berpusat pada manusia. Ekspresi wajah dibagi menjadi ekspresi makro dan ekspresi mikro (*micro-expression*). Dari perspektif temporal, ekspresi makro berlangsung antara 0,75 detik dan 2 detik, sedangkan ekspresi mikro bertahan antara 0,04 detik dan 0,2 detik [2]. Dalam istilah spasial, gerakan otot wajah ekspresi mikro lebih sedikit dibandingkan dengan ekspresi makro. Selain itu, ekspresi mikro bersifat spontan, artinya seseorang tidak tahu kapan dia menunjukkan ekspresi mikro. Informasi mikro yang penting mudah diabaikan terutama ketika orang menggunakan mata telanjang untuk mengamati. Alat Pelatihan Ekspresi Mikro telah dikembangkan untuk membantu pengenalan. Bahkan untuk profesional dengan pelatihan khusus, hasilnya masih belum ideal. Oleh karena itu, dibutuhkan teknologi komputer untuk membantu *micro-expression recognition* (MER) [3], [4].

Penelitian ini bertujuan untuk mendeteksi emosi saat proses pembelajaran dan untuk membandingkan ekstraksi fitur menggunakan CAS dengan HOG. Hal tersebut bermanfaat untuk membantu para pengajar dalam menentukan durasi dan metode pembelajaran yang tepat bagi siswa SMK.

Sejak 2011, sembilan kumpulan data ekspresi mikro representatif yang sudah dikembangkan: yaitu, USF-HD, kumpulan data Polikovskiy, York DDT, MEVIEW, SMIC, CASME, CASME II, SMM dan CAS(ME)2. Perlu dicatat bahwa ekspresi mikro yang berpose ini berbeda dari yang spontan. Selain itu, York DDT terdiri dari ekspresi mikro spontan dengan validitas ekologis yang tinggi. Namun demikian, mirip dengan deteksi kebohongan, data di York DDT dicampur dengan gerakan wajah non-emosional yang dihasilkan dari berbicara. Selain itu, semua kumpulan data (USF-HD, Polikovskiy, York DDT, MEVIEW, SMIC, CASME, CASME II, SMM dan CAS(ME)2) tidak tersedia untuk umum [8].

II. PENELITIAN SEBELUMNYA

Terdapat banyak metode yang dapat digunakan dalam pengklasifikasian *micro-expression* diantaranya yaitu *Convolution Neural Network*, K-NN, *Local Binary pattern*,

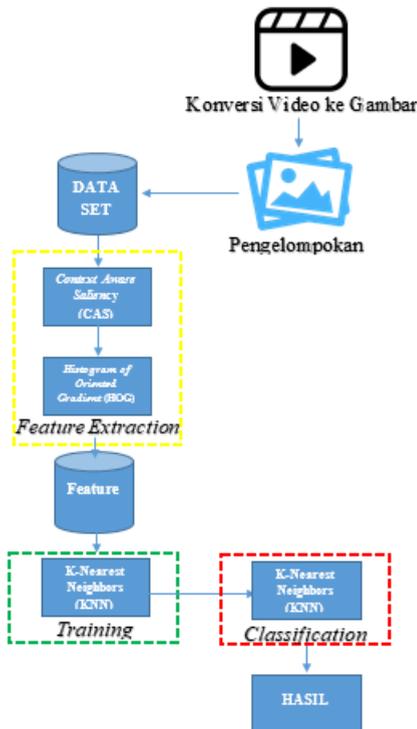
dan masih banyak lagi. Penelitian dengan memanfaatkan metode *Convolution Neural Network* pernah dilakukan oleh Sze-Teng Liong dkk pada tahun 2019 dalam jurnalnya yang berjudul “*Shallow Triple Stream Three-dimensional CNN (STSTNet) for Micro-expression Recognition*” [9]. Pada penelitian tersebut dataset yang digunakan adalah SMIC, CASME II, SMM. Makalah ini menyajikan CNN tiga dimensi dangkal (STSTNet) baru untuk mempelajari fitur panduan aliran optik untuk pengenalan ME. Representasi fitur yang ringkas dan diskriminatif dipelajari dari blok masukan yang terdiri dari tiga gambar aliran optik (yaitu aliran optik horizontal, aliran optik vertikal, dan lengkungan optik). Secara keseluruhan, pendekatan STSTNet yang diusulkan menunjukkan hasil identifikasi yang menjanjikan pada gabungan database ME sintetik yang baru, termasuk tiga database ME spontan, menghasilkan *Unweighted F1-score* sebesar 0,7353 dan *Unweighted Average Recall* sebesar 0,7605, mengungguli metode terancang saat ini [9].

Sedangkan penelitian menggunakan metode *K-Nearest Neighbor* (K-NN) Network pernah dilakukan oleh Ayu Wirdiani dkk pada tahun 2019 dalam jurnalnya yang berjudul “*Face Identification Based on K-Nearest Neighbor*” [10]. Pada penelitian tersebut dataset yang digunakan berasal dari pengambilan gambar manual. Penelitian dataset ini berisi 790 wajah dari 158 orang yang diambil dari beberapa sudut. Dataset tersebut belum melalui proses segmentasi sehingga masih terdapat background yang cukup signifikan di luar objek wajah. Dataset telah dipisahkan ke dalam setiap folder yang berisi nama pemilik individu wajah. Pada tahap pengujian, gambar yang diinput juga akan mengalami proses seperti pada tahap pelatihan, namun gambar yang digunakan merupakan gambar yang belum pernah diuji untuk pelatihan untuk mengukur keakuratan persamaan antara gambar latih dan gambar uji. Berdasarkan hasil percobaan dengan perubahan parameter diperoleh hasil yang berbeda-beda untuk setiap parameter. Pengujian k=1 memperoleh hasil akurasi sebesar 81%, pengujian k=2 memperoleh hasil akurasi sebesar 53%, pengujian k=3 memperoleh hasil akurasi sebesar 47%. Dari hasil uji coba dapat disimpulkan bahwa nilai k sangat mempengaruhi tingkat akurasi sistem. Nilai parameter k dan akurasi berbanding terbalik yaitu semakin besar nilai k maka semakin kecil keakuratan sistem identifikasi [10].

III. PERANCANGAN SISTEM

Alur sistem dijelaskan pada gambar 1, penulis menggunakan dataset siswa SMK Ma’arif Prambon sebanyak 45 siswa dan dengan dataset affecnet. Dataset dibagi menjadi data uji dan data latih. Data latih dan uji menggunakan 4116 citra pada setiap jenis *micro-expression*, Untuk data latih menggunakan 70% sedangkan data uji menggunakan 30% citra pada keseluruhan total dataset pada setiap jenis *micro expression* dengan rincian *anger* sejumlah 1065, *disgust* sejumlah 253, *fear* sejumlah 180, *happy* sejumlah 1220, *sad* sejumlah 895 dan *surprise* sejumlah 503.

Data latih kemudian dilakukan ekstraksi fitur menggunakan CAS dan setelah itu disimpan. Setelah diekstraksi menggunakan fitur CAS tersebut kemudian diekstraksi lagi menggunakan HOG. Fitur akhir hasil ekstraksi disimpan pada database. Proses selanjutnya adalah memisahkan dataset menjadi data latih dan data uji. Metode klasifikasi yang digunakan yaitu K-NN. K-NN digunakan untuk menghitung jarak antara fitur data uji dengan fitur data latih. K-NN digunakan untuk menghitung jarak antara fitur data uji dan fitur data latih. K-NN menghasilkan urutan dari jarak terkecil hingga terbesar. Jarak terkecil menunjukkan kesamaan antara data uji dan data latih.



Gambar 1. Gambaran umum sistem

A. PERANCANGAN PROSES

Proses pertama dilakukan perekaman wajah menggunakan Webcam Logitech 720P selama siswa mengikuti keseluruhan tampilan monitor dengan media visual. Siswa akan diwawancarai di studio podcast SMK Ma'arif NU Prambon dengan pendekatan *constructionist* dan *romantic* maksimal 20 menit dan ada juga yang diarahkan untuk menonton video pendek yang dapat menunjukkan ekspresi tersembunyi dari masing-masing siswa. Wawancara siswa akan direkam melalui webcam menggunakan aplikasi OBS. Hasil dari perekaman dapat dilihat pada gambar 2, dari gambar tersebut dapat dilihat bahwa proses perekaman video minimum resolusi 720 dengan 30 fps dan proses perekaman separuh badan.

Proses pada poin (1) kemudian akan dilakukan pemotongan per frame untuk mendapatkan input gambar. Pada gambar 3 adalah proses pemotongan yang dilakukan seleksi gambar secara otomatis dan manual. Input gambar

terdiri dari area wajah untuk kebutuhan pengenalan ekspresi wajah dan setengah badan yang terekam kamera untuk keperluan pengenalan ekspresi mikro



Gambar 2. Proses pengambilan Video

Setelah melalui konversi ke gambar, maka dilakukan proses pelabelan oleh guru untuk menentukan klasifikasi *micro expression*. Klasifikasi *micro-expression* terdiri dari 6 kelas yaitu bahagia, sedih, marah, takut, jijik, dan terkejut.



Gambar 3. Proses Cropping wajah

B. FEATURE EXTRACTION

B.1. CONTEXT AWARE SALIENCY

Pada proses feature extraction akan diterapkan metode *Context Aware Saliency* (CAS). CAS mengikuti empat prinsip dasar penglihatan visual manusia, yang didukung oleh bukti psikologis [11]:

1. Pertimbangan lokal tingkat rendah, termasuk faktor seperti kontras dan warna.
2. Pertimbangan global, yang menekan ciri-ciri yang sering muncul, dengan tetap mempertahankan ciri-ciri yang menyimpang dari norma.
3. Rule organisasi visual, yang menyatakan bahwa bentuk visual dapat memiliki satu atau beberapa pusat gravitasi di mana bentuk tersebut diatur.
4. Faktor tingkat tinggi, seperti wajah manusia [11].

B.1.1 LOCAL-GLOBAL SINGLE-SCALE SALIENCY

Ada dua tantangan untuk mendefinisikan *Saliency*. Yang pertama adalah mendefinisikan keunikan baik secara lokal maupun global. Opsi kedua adalah memasukkan informasi lokasi titik koordinat.

Menurut prinsip 1 dan 2, suatu piksel dianggap menonjol jika tampilannya unik. Namun, daripada berfokus pada piksel yang terisolasi, kita perlu fokus pada patch di sekitarnya yang memberikan konteks langsung.

Secara khusus, misalkan $d_{color}(p_i, p_j)$ adalah *Euclidean distance* antara patch tervektorisasi p_i dan p_j dalam ruang warna CIE L^*a^*b yang dinormalisasi ke

rentang [0;1]. Piksel i dianggap menonjol jika tinggi $d_{color}(p_i, p_j)$ tingginya $\forall j$.

Misalkan $d_{position}(p_i, p_j)$ adalah *Euclidean distance* antara posisi patch p_i dan p_j , yang dinormalisasi dengan dimensi gambar yang lebih besar. Berdasarkan observasi tersebut dapat didefinisikan ukuran ketidaksamaan antara sepasang patch sebagai:

$$d(p_i, p_j) = \frac{d_{color}(p_i, p_j)}{1 + c \cdot d_{position}(p_i, p_j)} \quad (1)$$

Dalam praktiknya, untuk mengevaluasi keunikan suatu *patch*, tidak perlu menggabungkan perbedaannya dengan semua *patch* gambar lainnya. Cukup untuk mempertimbangkan K *patch* yang paling mirip (jika *patch* yang paling mirip sangat berbeda dari p_i , maka jelas semua *patch* gambar sangat berbeda dari p_i). Oleh karena itu, untuk setiap *patch* p_i , dapat mencari K *patch* yang paling mirip $\{q_k\}_{k=1}^K$ dalam gambar, sesuai dengan (1). Sebuah piksel i menonjol ketika $d(p_i, q_k)$ tingginya $\forall k \in [1, k]$. Nilai *single-scale saliency* pada piksel i dalam skala r didefinisikan sebagai ($K = 64$):

$$S_i^r = 1 - \exp\left\{-\frac{1}{K} \sum_{k=1}^K d(p_i^r, q_k^r)\right\}, \quad (2)$$

B.1.2 MULTI-SCALE SALIENCY ENHANCEMENT

Piksel latar belakang (*patches*) cenderung memiliki *patches* serupa pada berbagai skala, misalnya, di wilayah sebagian besar yang homogen atau buram. Ini berbeda dengan piksel yang lebih menonjol yang dapat memiliki *patches* serupa pada beberapa skala tetapi tidak semuanya. Oleh karena itu, untuk menggabungkan beberapa skala agar semakin mengurangi arti-penting dari piksel latar belakang/*background*, meningkatkan kontras untuk wilayah yang lebih menonjol dan tidak menonjol.

Untuk *patch* p_i dengan skala r , dianggap sebagai kandidat tetangga semua *patches* pada gambar yang skalanya adalah $R_q = \{r, 1/2 r, 1/4 r\}$. Di antara semua *patch* ini, K *patch* yang paling mirip menurut (1) digunakan untuk menghitung *saliency*. Oleh karena itu, (2) dapat ditulis ulang menjadi [11]–[13]:

$$S_i^r = 1 - \exp\left\{-\frac{1}{K} \sum_{k=1}^K d(p_i^r, q_k^r)\right\}, \quad (3)$$

Selain itu, mewakili setiap piksel dengan kumpulan tambalan gambar multiskala yang berpusat padanya. Misalkan $R = \{r_1, \dots, r_M\}$ menunjukkan kumpulan ukuran *patches* yang harus dipertimbangkan untuk piksel i . Arti-penting pada piksel i diambil sebagai rata-rata arti-pentingnya pada skala yang berbeda [11]–[13]:

$$\bar{S}_i = \frac{1}{M} \sum_{r \in R} S_i^r, \quad (4)$$

B.1.3 INCLUDING THE IMMEDIATE CONTEXT

Menurut hukum Gestalt, bentuk visual dapat memiliki satu atau beberapa pusat gravitasi di mana bentuk diatur (prinsip (3)). Hal ini menunjukkan bahwa area yang dekat dengan fokus perhatian harus lebih banyak dieksplorasi secara signifikan daripada area yang jauh. Ketika wilayah di sekitar fokus menyampaikan konteksnya. Untuk mensimulasikan efek kontekstual visual ini terdapat dalam dua langkah. Pertama, area terlokalisasi yang paling banyak dikunjungi diekstraksi dari peta arti-penting yang dihasilkan oleh (4) [11]. Sebuah piksel dianggap ada jika nilai arti-pentingnya melebihi ambang tertentu ($S_i > 0.8$ dalam contoh yang ditunjukkan dalam paper ini). Kemudian, setiap piksel di luar area yang ada diberi bobot sesuai dengan jarak Euclidean-nya ke piksel terdekat. misalkan $d_{foci}(i)$ adalah posisi *Euclidean distance* antara piksel i dan piksel fokus perhatian terdekat, dinormalisasi ke kisaran [0;1]. *Saliency piksel* didefinisikan ulang sebagai [11]–[13]:

$$\hat{S}_i = \bar{S}_i \left(1 - d_{foci}(i)\right). \quad (5)$$

Perhatikan, bahwa *Saliency* dari daerah yang tidak menarik, seperti daerah yang kabur atau homogen, tetap rendah, karena (4) akan mendominasi. Namun, *Saliency* dari latar belakang yang menarik di sekitar objek yang menonjol akan ditingkatkan dengan (5) [12].

Hasil dari ekstraksi menggunakan CAS dapat dilihat pada gambar 4, pada gambar 4 ekstraksi menggunakan software python versi 3.9.13.

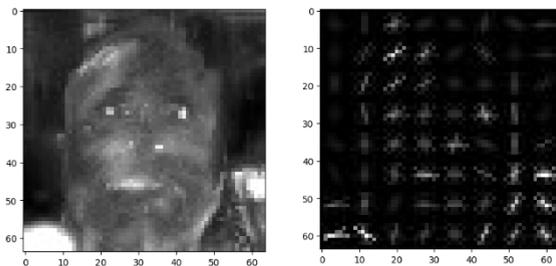


Gambar 4. Membandingkan hasil Saliency pada gambar satu objek dengan latar belakang yang tidak menarik

B.2. HISTOGRAM ORIENTED GRADIENT (HOG)

Sebagai deskriptor bentuk lokal berbasis tepi yang sukses, HOG pada awalnya digunakan untuk memecahkan masalah deteksi objek dengan mendeskripsikan informasi tepi lokal objek. Ide yang sama dari HOG dimanfaatkan di sini untuk menemukan arah gradien yang lebih dominan untuk setiap blok di citra [14]. Nilai besaran gradien untuk setiap piksel diakumulasikan tergantung pada sudut yang sesuai. Kemudian, untuk menemukan sudut dominan θ_d untuk

setiap blok pada gambar, kemudian memilih sudut gradien yang sesuai dengan nilai magnitudo gradien akumulasi maksimum G_d [15]. Berdasarkan kriteria tersebut, langkah-langkah algoritma yang digunakan untuk menemukan arah tepi yang dominan dirangkum dalam Algoritma 1. Contoh ilustrasi gambar grayscale yang berisi arah tepi yang berbeda untuk setiap BOI (*Block Of Interest*) [14]. Terdapat Tiga arah tepi, horizontal, vertikal dan diagonal, hampir ada dan tersebar sepanjang semua piksel gambar [14]–[17].



Gambar 5. contoh BOI untuk gambar *Micro-Expression Fear* yang berisi arah tepi yang berbeda

Dari gambar 5 merupakan hasil dari ekstraksi HOG dengan input gambar hasil dari ekstraksi CAS yang dilakukan menggunakan platform google colab. Proses ini memerlukan waktu sekitar 45-60 menit untuk semua gambar.

C. TAHAPAN K-NN

KNN adalah suatu algoritma sederhana yang menyimpan semua kondisi dan mengklasifikasikan ke dalam kondisi baru berdasarkan kemiripan nilai. Algoritma KNN telah digunakan sejak 1970 pada banyak aplikasi estimasi statistik dan pengenalan pola [18]. Metode ini dipilih dikarenakan dari penelitian sebelumnya mendapatkan hasil akurasi rata-rata maksimum 97% dalam jurnal “*Facial Expression Classification using KNN and Decision Tree Classifiers*” [19] dan 81% pada jurnal “*Face Identification Based on K-Nearest Neighbor*” [10]. Pada metode ini terdapat (6) adalah perhitungan jarak euclidian dimana K adalah jumlah fitur data, x dan y adalah data uji dan data latih. Algoritma K-NN pada dasarnya dilakukan dengan langkah berikut. Dimisalkan klasifikasi label ada dua yaitu label *Anger* dan label *Happy*.

Jika nilai $K=3$ dan data testing lebih dekat ke label *Anger* dengan dua tetangga dan satunya lebih dekat ke label *Happy* maka hasilnya akan diputuskan menjadi label *Anger*. Penentuan label diambil dari mayoritas terhadap K [10], [18]–[25].

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (6)$$

IV. HASIL DAN PEMBAHASAN

A. IMPLEMENTASI PADA PSEUDOCODE

Proses implementasi Script program pada python memiliki beberapa urutan yaitu proses ekstraksi fitur

menggunakan CAS. Hasil gambar dari CAS harus disimpan terlebih dahulu ke dalam folder dikarenakan proses ini memakan waktu 15 menit tiap gambar. Proses CAS selesai dalam 7 hari menggunakan 20 Notebook. Setelah itu hasil gambar yang sudah di ekstraksi menggunakan CAS dilakukan ekstraksi lagi menggunakan HOG. Hasil ekstraksi disimpan dan dijadikan .zip, kemudian dilakukan upload ke google drive. Setelah itu proses klasifikasi menggunakan K-NN, proses ini memisahkan 70% data training dan 30% data testing secara random. Berikut pseudocode Algoritma K-NN [22]:

Input:

X: data training,

Y: label kelas dari X,

K: jumlah tetangga terdekat.

Output:

Kelas sampel uji x.

Start

Tentukan Nilai K

Klasifikasikan (X, Y, x)

1. for setiap sampel x do

 Hitung jarak: $d(x, X) = \sqrt{\sum_{i=1}^n (x_i - X_i)^2}$

end for

2. Temukan tetangga terdekat

3. Tentukan kelas mayoritas

4. Klasifikasikan titik data baru x pada kelas mayoritas:

$C(x) = \text{argmax}_j \sum_{x_j \in \text{KNN}} C(X_j, Y_k)$

End.

Berikut langkah-langkah utama cara kerja algoritma KNN:

- Langkah pertama adalah memilih nilai K yang menentukan jumlah tetangga yang akan diperiksa untuk menentukan klasifikasi suatu bagian data. Menunjukkan Misalnya, jika K sama dengan 3, algoritma akan mempertimbangkan tiga tetangga terdekat.
- Hitung jarak Selanjutnya algoritma menghitung jarak antara titik data baru yang akan diklasifikasikan dengan seluruh titik data yang ada di set pelatihan. Untuk menghitung jarak ini menggunakan *Euclidean distance*.
- Mencari tetangga terdekat Setelah menghitung jarak, KNN akan menentukan K tetangga terdekat yang memiliki jarak terpendek ke titik data baru.
- Identifikasi kelas mayoritas Dari tetangga yang teridentifikasi, KNN menghitung jumlah titik data pada setiap kelas atau kategori. Kelas yang memiliki jumlah tetangga terbanyak akan dianggap sebagai klasifikasi untuk titik data baru tersebut.
- Klasifikasikan titik data baru Akhirnya, KNN akan menetapkan titik data baru ke dalam kelas yang telah ditentukan sebagai mayoritas oleh tetangga-tetangga terdekatnya. Dengan demikian, titik data baru akan mendapatkan label atau kategori yang sesuai.

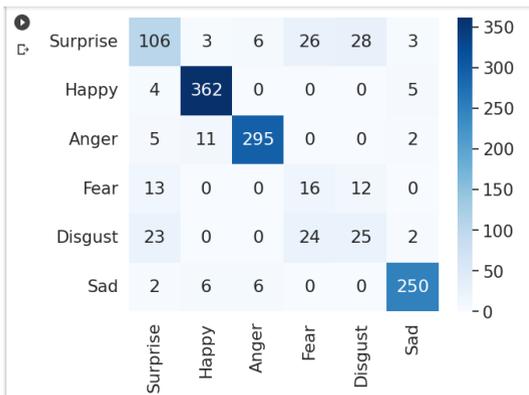
B. EVALUASI HASIL

Hasil dari evaluasi menggunakan 4 indikator utama yaitu akurasi, *precision*, *recall*, dan *f1-Score*. Hasil dari ketiga

perbandingan K=1, K=3 dan K=5 disajikan pada Tabel 1, 2 dan 3. Pada Gambar 6, Gambar 8 dan Gambar 10 merupakan *Confusion Matrix* dari masing-masing pengujian K-NN dan Ekstraksi RGB. Hasil *scatter plot* menunjukkan bahwa perubahan nilai K terhadap nilai *precision*, *recall* dan *precision* semakin meningkat. Adapun observasi yang dihasilkan adalah percobaan dua nilainya selalu lebih tinggi dibandingkan percobaan pertama.

TABEL I
NILAI EVALUASI DENGAN K=1
(PERSENTASE DARI GAMBAR 7)

	Precision	Recall	F1-Score
Surprise	69%	62%	65%
Happy	95%	98%	96%
Anger	96%	94%	95%
Fear	24%	39%	30%
Disgust	38%	34%	36%
Sad	95%	95%	95%
Akurasi Rata-rata	85%		



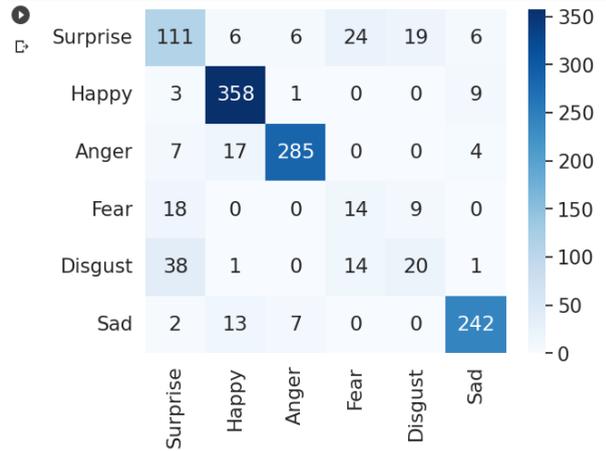
Gambar 6. *Confusion Matrix* K=1 dari hasil pengujian menggunakan google colab

	precision	recall	f1-score	support
0	0.69	0.62	0.65	172
1	0.95	0.98	0.96	371
2	0.96	0.94	0.95	313
3	0.24	0.39	0.30	41
4	0.38	0.34	0.36	74
5	0.95	0.95	0.95	264
accuracy			0.85	1235
macro avg	0.70	0.70	0.70	1235
weighted avg	0.86	0.85	0.86	1235

Gambar 7. Implementasi K-NN dengan K=1 dari hasil pengujian menggunakan google colab

TABEL II
NILAI EVALUASI DENGAN K=3
(PERSENTASE DARI GAMBAR 9)

	Precision	Recall	F1-Score
Surprise	62%	65%	63%
Happy	91%	96%	93%
Anger	95%	91%	93%
Fear	27%	34%	30%
Disgust	42%	27%	33%
Sad	92%	92%	92%
Akurasi Rata-rata	83%		



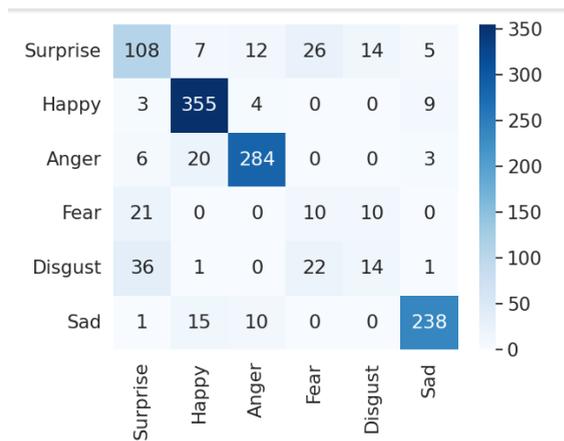
Gambar 8. *Confusion Matrix* K=3 dari hasil pengujian menggunakan google colab

	precision	recall	f1-score	support
0	0.62	0.65	0.63	172
1	0.91	0.96	0.93	371
2	0.95	0.91	0.93	313
3	0.27	0.34	0.30	41
4	0.42	0.27	0.33	74
5	0.92	0.92	0.92	264
accuracy			0.83	1235
macro avg	0.68	0.67	0.67	1235
weighted avg	0.83	0.83	0.83	1235

Gambar 9. Implementasi K-NN dengan K=3 dari hasil pengujian menggunakan google colab

TABEL III
NILAI EVALUASI DENGAN K=5
(PERSENTASE DARI GAMBAR 11)

	Precision	Recall	F1-Score
Surprise	62%	63%	62%
Happy	89%	96%	92%
Anger	92%	91%	91%
Fear	17%	24%	20%
Disgust	37%	19%	25%
Sad	93%	90%	92%
Akurasi Rata-rata	82%		



Gambar 10. *Confusion Matrix* K=5 dari hasil pengujian menggunakan google colab

	precision	recall	f1-score	support
0	0.62	0.63	0.62	172
1	0.89	0.96	0.92	371
2	0.92	0.91	0.91	313
3	0.17	0.24	0.20	41
4	0.37	0.19	0.25	74
5	0.93	0.90	0.92	264
accuracy			0.82	1235
macro avg	0.65	0.64	0.64	1235
weighted avg	0.81	0.82	0.81	1235

Gambar 11. Implementasi K-NN dengan K=5 dari hasil pengujian menggunakan google colab

Dari evaluasi hasil klasifikasi K-NN yang dilakukan dengan menggunakan ekstraksi CAS dan HOG terhadap 6 jenis *Micro-Expression* yaitu *surprise*, *happy*, *anger*, *fear*, *disgust*, dan *sad*. Akurasi yang didapatkan dibagi menjadi tiga yaitu dengan secara rata-rata nilai K=1 sebesar 85%, nilai K=3 sebesar 83% dan nilai K=5 sebesar 82%. Hasil Akurasi yang didapatkan menurun dengan penentuan K yang semakin tinggi sehingga ditetapkan K=1 untuk hasil yang maksimal dalam klasifikasi *micro-expression* enam label.

Adapun akurasi setiap jenis label pada K=1 adalah 69% untuk ekspresi *surprise*, 95% untuk ekspresi *happy*, 96% untuk ekspresi *anger*, 24% untuk ekspresi *fear*, 38% untuk ekspresi *disgust* dan 95% untuk ekspresi *sad*. Adapun hal yang paling mencolok adalah akurasi ekspresi *fear* dan *disgust* yaitu dibawah 40%. Ketika diobservasi dalam percobaan pertama, kedua dataset ini memiliki jumlah data yang paling sedikit dikarenakan siswa tidak dapat mengeluarkan ekspresi tersebut meskipun sudah dilakukan wawancara. Hal ini merupakan salah satu faktor kelemahan dari algoritma K-NN dengan ekstraksi fitur CAS dan HOG.

Pada observasi percobaan kedua dengan nilai K=3 terjadi penurunan akurasi pada 4 ekspresi tetapi pada ekspresi *disgust* terjadi peningkatan yang sebelumnya adalah 38% menjadi 42%. Dan pada observasi percobaan ke tiga dengan nilai K=5 terjadi penurunan akurasi pada 5 ekspresi dan yang paling mencolok adalah pada ekspresi *fear* yaitu dari 27% ke 17%. Hal ini disebabkan rata-rata ekspresi *fear* mengambil dataset dari affecnet sebanyak 180, dan dataset dari siswa tidak ada. Siswa sendiri sangat sulit memunculkan *micro-Expression* secara spontan pada ekspresi *fear* dan *disgust*. Hal ini menunjukkan bahwa banyaknya dataset mempengaruhi akurasi tiap ekspresi dari pengujian klasifikasi *micro-expression* menggunakan K-NN menggunakan fitur CAS dan HOG.

V. KESIMPULAN

Kesimpulan pada penelitian ini adalah ekstraksi fitur menggunakan CAS sangat tidak cocok untuk pengujian realtime dikarenakan untuk memproses 3 skala membutuhkan waktu sekitar 15-20 menit untuk 1 citra. CAS juga harus membutuhkan dataset yang RGB dikarenakan prinsip CAS yang pertama pertimbangan lokal tingkat rendah, termasuk faktor seperti kontras dan warna sangat mempengaruhi dari saliency dari citra tersebut. Dataset *fear* dan *disgust* sangat sulit didapat dikarenakan sangat sulit sekali para siswa menunjukkan kedua ekspresi tersebut

secara spontan. Hasil dari klasifikasi menggunakan K-NN didapatkan akurasi 85% dengan menggunakan K=1, sedangkan dengan K=3 mendapatkan akurasi 83%, sedangkan dengan K=5 mendapatkan akurasi 82%. Proses klasifikasi K-NN sangat cepat dan akurasinya juga sangat bagus untuk mengklasifikasikan *micro expression* dan sangat disarankan untuk dijadikan metode computer vision yang sangat mudah difahami.

PERAN PENULIS

Nikko Riestian Putra Wardoyo: Analisa formal, Pengumpulan dataset, Metodologi, Perangkat lunak, Visualisasi & Administrasi proyek.

Joan Santoso: Penulisan Review & Penyuntingan;

Esther Irawati Setiawan: Penulisan Review & Penyuntingan;

COPYRIGHT



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

DAFTAR PUSTAKA

- [1] X. Ben *et al.*, "Video-based facial micro-expression analysis: A survey of datasets, features and algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5826–5846, 2021.
- [2] L. Lei, T. Chen, S. Li, and J. Li, "Micro-expression recognition based on facial graph representation learning and facial action unit fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1571–1580.
- [3] Y. Liu, H. Du, L. Zheng, and T. Gedeon, "A neural micro-expression recognizer," in *2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019)*, 2019, pp. 1–4.
- [4] N. Van Quang, J. Chun, and T. Tokuyama, "CapsuleNet for micro-expression recognition," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, 2019, pp. 1–7.
- [5] N. Amynarto, Y. A. Sari, and R. C. Wihandika, "Pengenalan emosi berdasarkan ekspresi mikro menggunakan metode local binary pattern," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, pp. 3230–3238, 2018.
- [6] D. Goleman, R. Boyatzis, and A. McKee, "The emotional reality of teams," *J. Organ. Excell.*, vol. 21, no. 2, pp. 55–65, 2002.
- [7] P. Ekman, "What scientists who study emotion agree about," *Perspect. Psychol. Sci.*, vol. 11, no. 1, pp. 31–34, 2016.
- [8] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, and X. Fu, "CASME database: A dataset of spontaneous micro-expressions collected from neutralized faces," in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, 2013, pp. 1–7.
- [9] S.-T. Liong, Y. S. Gan, J. See, H.-Q. Khor, and Y.-C. Huang, "Shallow triple stream three-dimensional cnn (ststnet) for micro-expression recognition," in *2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019)*, 2019, pp. 1–5.
- [10] N. K. A. Wirdiani, P. Hridayami, N. P. A. Widiari, K. D. Rismawan, P. B. Candradinata, and I. P. D. Jayantha, "Face identification based on K-nearest neighbor," *Sci. J. Informatics*, vol. 6, no. 2, pp. 150–159, 2019.
- [11] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915–1926, 2011.

- [12] W. Kim and C. Kim, "A novel image importance model for content-aware image resizing," in *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 2469–2472.
- [13] M. Ahmadi, N. Karimi, and S. Samavi, "Context-aware saliency detection for image retargeting using convolutional neural networks," *Multimed. Tools Appl.*, vol. 80, pp. 11917–11941, 2021.
- [14] L. R. Cerna, G. Camara-Chavez, and D. Menotti, "Face detection: Histogram of oriented gradients and bag of feature method," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV)*, 2013, p. 1.
- [15] O. Déniz, G. Bueno, J. Salido, and F. la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognit. Lett.*, vol. 32, no. 12, pp. 1598–1603, 2011.
- [16] M. A. Hameed, M. Hassaballah, S. Aly, and A. I. Awad, "An adaptive image steganography method based on histogram of oriented gradient and PVD-LSB techniques," *IEEE Access*, vol. 7, pp. 185189–185204, 2019.
- [17] R. G. Guendel, F. Fioranelli, and A. Yarovoy, "Phase-based classification for arm gesture and gross-motor activities using histogram of oriented gradients," *IEEE Sens. J.*, vol. 21, no. 6, pp. 7918–7927, 2020.
- [18] M. M. Fouad, H. M. Zawbaa, T. Gaber, V. Snasel, and A. E. Hassanien, "A fish detection approach based on BAT algorithm," in *The 1st International Conference on Advanced Intelligent System and Informatics (AISII2015), November 28-30, 2015, Beni Suef, Egypt*, 2016, pp. 273–283.
- [19] M. Murugappan *et al.*, "Facial expression classification using KNN and decision tree classifiers," in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2020, pp. 1–6.
- [20] R. Mohamed, M. M. Yusof, N. Wahid, N. Murli, and M. Othman, "Bat algorithm and k-means techniques for classification performance improvement," *Indones J Electr Eng Comput Sci*, vol. 15, no. 3, pp. 1411–1418, 2019.
- [21] L. Farokhah, "Implementasi K-Nearest Neighbor untuk Klasifikasi Bunga Dengan Ekstraksi Fitur Warna RGB," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, pp. 1129–1135, 2020.
- [22] K. A. Sugiarta, I. Cholissodin, and E. Santoso, "Optimasi K-Nearest Neighbor Menggunakan Bat Algorithm Untuk Klasifikasi Penyakit Ginjal Kronis," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 10, pp. 10301–10308, 2019.
- [23] K. W. Mahardika, Y. A. Sari, and A. Arwan, "Optimasi K-Nearest Neighbour Menggunakan Particle Swarm Optimization pada Sistem Pakar untuk Monitoring Pengendalian Hama pada Tanaman Jeruk," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 9, pp. 3333–3344, 2018.
- [24] W. Nugroho, "Optimasi Metode K-Nearest Neighbours dengan Backward Elimination Menggunakan Dataset Software Effort Estimation," *Bianglala Inform.*, vol. 8, no. 2, pp. 129–133, 2020.
- [25] F. Fandiansyah, J. Y. Sari, and I. P. Ningrum, "Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor," *Ultim. J. Tek. Inform.*, vol. 9, no. 1, pp. 1–9, 2017.