

Perbandingan Identifikasi Wajah Dengan Ekstraksi Fitur Haralick Dan CNN

Fani Nurona Cahya¹

¹STMIK Nusa Mandiri, Cipinang Melayu, Kec. Makasar, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13620, Indonesia

e-mail: 14002395@nusamandiri.ac.id¹

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi : 27 Januari 2021

Revisi Akhir : 07 Februari 2021

Diterbitkan Online : 12 Februari 2021

Kata Kunci:

Feature Extraction, Random Forest, CNN.

Korespondensi :

Telepon / Hp : +62 852-1414-5900

E-mail : 14002395@nusamandiri.ac.id

A B S T R A K

Kecanggihan teknologi telah berdampak pada setiap segi kehidupan. Salah satunya pengolahan citra yang menjadikan objek wajah sebagai karakter utama. Deteksi wajah digunakan untuk mengetahui ada atau tidaknya wajah pada suatu gambar sehingga keberadaannya sangat vital. Penelitian ini adalah untuk mendapatkan algoritma pengenalan yang kuat dengan akurasi tinggi. Struktur umum proses pengenalan wajah dalam penelitian ini mencoba dua eksperimen yakni dengan menggunakan Feature Extraction Haralick dengan Random Forest, dan klasifikasi dengan menggunakan CNN. Penelitian feature extraction dengan haralick ini terdiri dari tiga tahap. Ini dimulai dengan tahap pra-pemrosesan: konversi ruang warna dan perubahan ukuran gambar, dilanjutkan dengan ekstraksi fitur wajah, dan kemudian set fitur yang diekstraksi diklasifikasikan. Dalam sistem ini, Random Forest Classifier dan CNN akan menjadi acuan novelty untuk merealisasikan tahap terakhir berdasarkan fitur wajah. Hasil akurasi dari eksperimen ini masih kurang sehingga perlu mencoba eksperimen lain.

1. PENDAHULUAN

Kemajuan teknologi sangat berdampak pada segi kehidupan. Tentu saja hal ini dipengaruhi oleh kebutuhan yang sangat kompleks. Pendeteksi wajah digunakan untuk mengetahui ada atau tidaknya wajah pada suatu citra sehingga keberadaannya sangat vital.

Dengan seiring berkembangnya zaman, teknologi terus berkembang sangat pesat. Banyak teknologi baru yang ditemukan salah satunya yaitu pengolahan citra digital. Identifikasi pada sebuah citra sudah lama dikembangkan, salah satunya yaitu membedakan tekstur yang ada pada citra. Tekstur citra bisa dibedakan dengan kerapatan, keseragaman, kekasaran dan keteraturan[1]. Suatu komputer tidak bisa membedakan tekstur seperti halnya penglihatan manusia, maka dengan demikian digunakan analisis tekstur untuk mengetahui pola suatu citra digital. Analisis tekstur bisa menghasilkan nilai dari ciri atau karakteristik tekstur yang kemudian bisa diolah oleh sebuah komputer untuk proses klasifikasi [2]. Arti klasifikasi adalah proses pengelompokan, dimana artinya memisahkan benda/entitas yang berbeda. Klasifikasi menganalisis sifat numerik citra fitur dan mengatur data ke dalam berbagai kategori [3].

Deep Learning sudah menjadi suatu topik hangat dalam dunia Machine Learning karena kapabilitasnya yang signifikan dalam memodelkan bermacam-macam data kompleks seperti citra dan suara. Salah satu problem pada visi komputer yang sudah lama di cari solusinya adalah klasifikasi objek pada citra secara umum. Bagaimana menduplikasi kemampuan manusia dalam memahami informasi citra, agar komputer dapat mengenali objek pada citra selayaknya manusia [4].

Proses pengenalan wajah adalah proses untuk mengenali wajah orang yang relevan dengan sistem penglihatan. Ini sudah menjadi alat interaksi manusia komputer yang penting karena

penggunaannya pada sistem keamanan, kontrol akses, pengawasan video, area komersial dan bahkan digunakan di jejaring sosial seperti pada facebook juga.

Setelah perkembangan kecerdasan buatan menjadi pesat, pengenalan wajah sekali lagi menarik perhatian karena sifatnya yang tidak mengganggu dan karena ini adalah metode yang utama untuk identifikasi orang bagi manusia dibandingkan dengan jenis teknik biometrik lainnya[5].

Kontribusi utama dari penelitian ini adalah agar mendapatkan algoritma pengenalan yang sangat kuat dengan hasil akurasi yang tinggi. Struktur umum dalam proses pengenalan wajah ini yakni mencoba dua eksperimen dengan menggunakan *Feature Extraction haralick* dengan Metode Random Fores, dan klasifikasi dengan menggunakan CNN.

2. PENELITIAN TERKAIT

Terdapat beberapa literatur yang digunakan sebagai acuan untuk mengembangkan aplikasi pengenalan wajah yang dirancang, antara lain:

Pertama, penelitian yang dilakukan oleh Sigit Wasista, Bima Sena Bayu D Sandra Agustyan Putra, pada penelitian ini membahas mengenai penggunaan pengenalan wajah pada sistem absensi dan menggunakan sistem absensi lama berbasis *smart card*. Tujuannya adalah untuk menguji seberapa tingkat keberhasilan sistem absensi berbasis pengenalan wajah dalam mengenali wajah seseorang menggunakan jarak sebagai parameter untuk mengukur tingkat akurasi [6].

Kedua, Shanshan Guo, Shiyu Chen and Yanjie Lie (2016). Mereka mengatakan dalam penelitiannya mencoba menggabungkan CNN dan SVM untuk mengenali citra wajah. CNN digunakan sebagai ekstraktor fitur untuk memperoleh fitur luar biasa secara otomatis. Sementara SVM digunakan sebagai

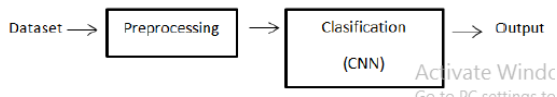
pengklasifikasian dari pada CNN untuk mengenali semua kelas [7].

Ketiga, Guofeng Zou, Guixia Fu, Mingliang Gao, Jinfeng Pan, Zhung Liu (2020). Pada penelitiannya mereka mengusulkan kerangka kerja fusi multi-fitur baru. Kemudian mengusulkan augmentasi data tingkat fitur berdasarkan transformasi Gabor, untuk bisa menangkap fitur wajah multi-level filter Gabor multi-skala Gabor Multi-skala dan multi-arah untuk bisa mewujudkan perluasan wajah di ruang fitur, Guofen Zou dkk juga mengusulkan skala kecil aditif dalam model CNN, yang cocok untuk kumpulan sample data kecil dan dapat mengekstrak secara efektif fitur dalam nonlinier dari berbagai pose wajah [8].

Keempat, Swarnima Singh, Durgesh Singh, Vikash Yadav (2020). Pada penelitiannya menjelaskan bahwa Algoritma Ekstraksi Fitur LBP dan HOG lebih disukai untuk aplikasi waktu nyata. Fitur HOG teknik ekstraksi telah digunakan dalam penelitiannya ini untuk mencoba mendeteksi menggunakan analisa citra. PCA penting metode fitur dalam metode wajah Eigen saat ini merupakan gelombang otak penting untuk hampir semua pengenalan wajah algoritma berkembang menjadi pemrograman baru dan metode lebih baik [9].

3. METODE PENELITIAN

Untuk melakukan penelitian kali ini yang pertama harus dilakukan adalah pengumpulan dataset sebagai data latih dan data uji. Kemudian Segmentasi dan ekstraksi ciri dilakukan pada masing-masing citra latih dan citra uji. Kemudian untuk pengenalan buah segar atau busuk berdasarkan informasi ciri tersebut menggunakan CNN.



Gambar 1. Alur Penelitian

3.1. Dataset

Dataset ini melewati beberapa seleksi di antara beberapa dataset yang tersedia di Kaggle.com. alasan memilih dataset ini karena penelitian ini baru beberapa kali di kembangkan sehingga peneliti ingin mencoba untuk mengembangkannya. Dataset ini berjumlah 440 dataset yang terbagi dari 40 kelas. Penelitian ini memaparkan klasifikasi beberapa sampel wajah.



Gambar 2. Contoh dataset

3.2. Pre-Processing

Pre-processing adalah salah satu tahapan yang sangat penting dalam data pada proses. Terkadang dalam data tersebut terdapat berbagai masalah yang bisa mengganggu hasil dari proses. Pre-processing merupakan salah satu tahapan menghilangkan permasalahan-permasalahan yang bisa mengganggu hasil dari pada proses data.

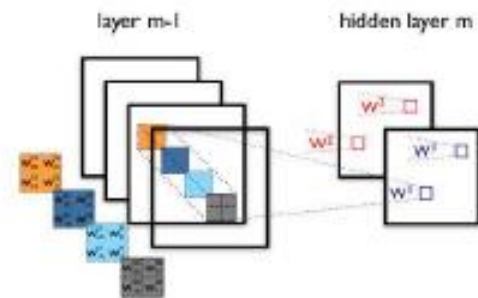
3.3. Metode CNN

A. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang dirancang untuk mengolah data dua dimensi. CNN termasuk dalam Deep Neural Network karena kedalaman jaringan yang tinggi dan banyak diaplikasikan di citra. CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. [10] Cara kerja CNN ini mempunyai kemiripan pada MLP, namun pada CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak sama dengan MLP yang setiap neuron berukuran satu dimensi

B. Konsep Convolutional Neural Network (CNN)

Cara kerja CNN mempunyai kemiripan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi. Pada CNN, data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti pada Gambar.



Gambar 3. Proses Konvolusi pada CNN

C. Arsitektur jaringan CNN

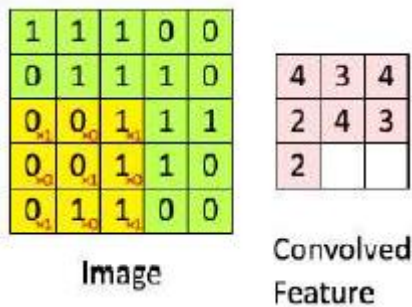
JST terdiri dari beberapa layer dan berbagai neuron pada masing-masing layer. Kedua hal tersebut tidak bisa ditentukan dengan aturan yang pasti dan berlaku berbeda-beda pada data yang berbeda.[11]

Sebuah CNN terdiri dari beberapa layer. Berdasarkan arsitektur LeNet5 [12], terdapat empat macam layer utama pada sebuah CNN namun yang diterapkan pada penelitian ini hanya tiga macam lapisan antara lain:

1) Convolution Layer

Convolution Layer menjalankan operasi konvolusi pada output dari layer sebelumnya. Konvolusi adalah suatu istilah matematis yang mengaplikasikan fungsi pada output fungsi lain secara berulang. Kernel bergerak dari sudut kiri atas ke kanan bawah.

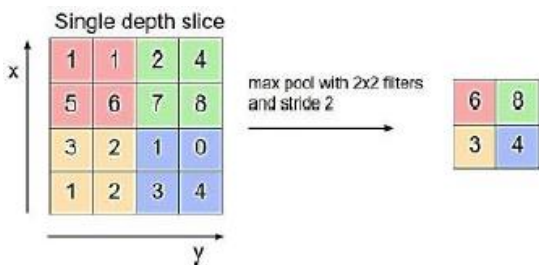
Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.



Gambar 4. Operasi Konvolusi

2) Subsampling Layer

Subsampling adalah proses mereduksi ukuran sebuah data citra. Dalam sebagian besar CNN, metode subsampling yang digunakan adalah max pooling. Grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya.



Gambar 5. Operasi Max Pooling

Menurut Springenberg dkk [13] penggunaan pooling layer pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah convolution layer dengan stride yang sama dengan pooling layer yang bersangkutan.

3) Fully Connected Layer

Layer ini adalah layer yang biasanya digunakan untuk penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, fully connected layer hanya dapat

diimplementasikan di akhir jaringan. Hal tersebut membuat penggunaan fully connected layer pada CNN sekarang tidak banyak dipakai.

D. Proses Training

Proses training merupakan tahapan dimana CNN digunakan untuk mendapatkan nilai akurasi yang tinggi dari klasifikasi yang dilakukan. Untuk memulai proses feedforward diperlukan jumlah dan ukuran layer yang akan dibentuk, ukuran subsampling, citra vektor yang diperoleh.

1) Proses Feedforward

Proses feed forward merupakan langkah pertama dalam proses training. Proses ini akan menghasilkan beberapa lapisan untuk mengklasifikasi data citra yang mana menggunakan bobot dan bias yang telah diperbarui dari proses backpropagation. Tahap ini juga akan digunakan kembali saat proses testing.

2) Proses Backpropagation

Proses backpropagation merupakan langkah kedua dari proses training. Pada tahap ini seperti yang telah dijelaskan pada sub bab 2.3.6 hasil proses dari feed forward di-trace kesalahannya dari lapisan output sampai lapisan pertama. Untuk memberikan ciri bahwa data tersebut telah di-trace diperoleh bobot dan bias yang baru.

3) Perhitungan Gradient

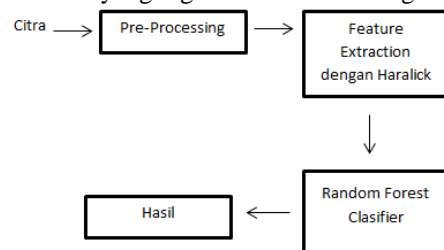
Pada proses gradient untuk jaringan konvolusi merupakan proses untuk memperoleh nilai bobot dan bias yang baru yang akan diperlukan saat training.

E. Proses Testing

Proses testing merupakan proses klasifikasi menggunakan bobot dan bias dari hasil proses training. Proses ini tidak jauh berbeda dengan proses training yang membedakannya tidak terdapat proses backpropagation setelah proses feedforward. Sehingga hasil akhir dari proses ini yakni menghasilkan akurasi dari klasifikasi yang dilakukan, data yang gagal diklasifikasi, nomor citra yang gagal diklasifikasi, dan bentuk network yang terbentuk dari proses feedforward. Dengan bobot dan bias yang baru proses feedforward diterapkan yang kemudian menghasilkan lapisan output. Lapisan output sudah fully connected dengan label yang disediakan. Hasil fully connected tersebut diperoleh data yang gagal dan berhasil diklasifikasi.

3.3. Metode Ekstraksi Feature Haralick dengan Random Forest

Alur penelitian yang digunakan adalah sebagai berikut :



Gambar 6. Model Haralick dengan random Forest

A. Pre-processing

Pre-processing Merupakan serangkaian proses untuk mempersiapkan citra sebelum proses segmentasi. Proses ekstraksi fitur dilakukan proses pengambilan ciri dari wajah yang menggambarkan karakteristik. Proses klasifikasi tingkat keparahan dilakukan menggunakan klasifikasi *Random Forest* [14].

B. Ekstraksi feature

Ekstraksi feature adalah salah satu tugas penting yang bisa meningkatkan efisiensi seluruh sistem. Fitur menjelaskan beberapa properti yang bisa dihitung dari gambar masukan yang diberikan di dalam proses. Ekstraksi fitur menggunakan data citra hasil tahap *segmentasi* citra. Tahap ekstraksi fitur ini menggunakan fitur warna, luas dan bentuk.

a) *Ekstraksi feature Haralick*

Fitur ini menceritakan korelasi intensitas antara keduanya piksel dalam gambar pada jarak tertentu yang di tetapkan arah. GLCM tidak hanya menawarkan hubungan antar-piksel tetapi juga ketergantungan dan priodisitas tingkat abu-abu spasial [15].

b) *Hu-Moment*

Untuk mengekstrak fitur *Hu-Moments* dari gambar, digunakan fungsi *cv2. Hu-Moments* yang disediakan oleh OpenCV. Itu berarti kita menghitung momen gambar dan mengubahnya menjadi *vektor* menggunakan *flatten*.

c) *Colour Histogram*

Pada proses ekstraksi ciri warna diawali dengan merubah aras warna RGB menjadi aras HSV. Nilai warna HSV dari masing-masing piksel yang menyusun gambar di kelompokkan menjadi 8 kelompok rentang nilai piksel warna . Tiap kelompok jumlah anggota kemudian dinormalisasi dengan cara di bagi dengan hasil perkalian panjang dan lebar gambar banyak piksel warna penyusun gambar.

C. Random Forest

Metode yang bisa digunakan dalam penelitian sangat banyak, namun pada penelitian kali ini penulis mencoba menggunakan metode *Random forest*. Karena Semakin banyak pohon, semakin akurat hasil klasifikasinya. Selain itu juga *RF* dapat menangani input variabel yang besar, menyeimbangkan error dalam *unbalanced* dataset. Teknik ini disebut *Balanced Random Forest*. Dalam algoritma *RF*, diperlukan algoritma untuk membangun *tree*. Salah satu *algoritma* yang dapat digunakan adalah *algoritma Classification and Regression Tree*. klasifikasi dengan membangun banyak pohon klasifikasi. [16].

Kelebihan menggunakan *random forest* adalah bisa mengklasifikasi data yang memiliki atribut yang tidak

kumplit, bisa dipakai untuk klasifikasi dan regresi, namun tidak terlalu bagus untuk regresi, lebih cocok untuk pengklasifikasian data serta dapat digunakan untuk menangani data sampel yang banyak.

Proses klasifikasi pada *random forest* berawal dari memecah data sampel yang ada kedalam *decision tree* secara acak. Setelah pohon terbentuk, maka akan dilakukan *voting* pada setiap kelas dari data sampel..

kemudian dilakukan prediksi untuk melihat bagaimana performa model *Random Forest* terhadap *testing data*. Untuk mempermudah dalam melihat performa model digunakan *confussion matrix*.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Metode CNN

a) Arsitektur CNN

Arsitektur CNN yang diusulkan di dalam penelitian ini terdiri dari 3 convolutional layer, 1 dense layer, dan 1 output layer dengan skema sebagaimana dapat dilihat pada gambar.7.

```

Model: "sequential"
-----
Layer (type)                Output Shape         Param #
-----
conv2d (Conv2D)              (None, 64, 64, 16)  80
max_pooling2d (MaxPooling2D) (None, 32, 32, 16)  0
conv2d_1 (Conv2D)            (None, 32, 32, 32)  2080
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)  0
conv2d_2 (Conv2D)            (None, 16, 16, 64)  8256
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64)    0
dropout (Dropout)           (None, 8, 8, 64)    0
flatten (Flatten)            (None, 4096)         0
dense (Dense)                (None, 500)          2048500
dropout_1 (Dropout)         (None, 500)          0
dense_1 (Dense)              (None, 40)           20040
-----
Total params: 2,078,956
Trainable params: 2,078,956
Non-trainable params: 0
    
```

Gambar 7. Arsitektur CNN

Selanjutnya model dikompilasi dengan parameter `loss='categorical_crossentropy'`, `optimizer='adam'`, dan `metrics=['accuracy']`.

b) Distribusi Data Train, Test, dan Valid

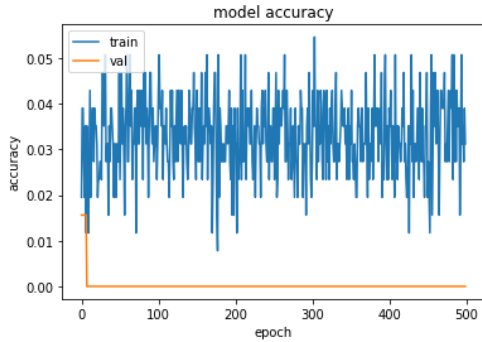
Dataset		
400		
Train		Test
256		64
Train	Valid	
256	80	

Table 1. Distribusi Data Train, Test, dan Valid

Selanjutnya data training dan validasi tersebut digunakan dalam proses training, tuning, dan evaluasi model CNN sedangkan data test digunakan untuk menguji performa dari model hasil training tersebut.

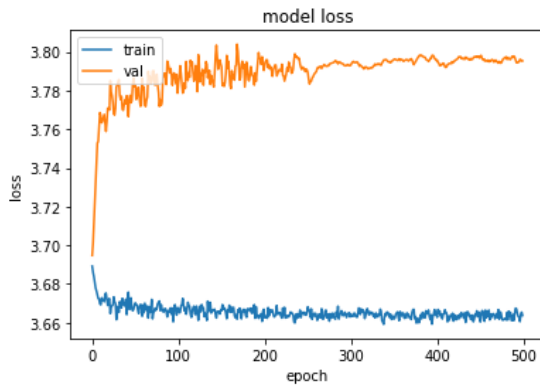
Selanjutnya dilakukan training dan evaluasi terhadap model tersebut dengan batch size = 32 dan jumlah epoch sebanyak 500 epochs.

c). Model Accuracy



Gambar 8. Model Accuracy

d). Model Loss



Gambar 9. Model Loss

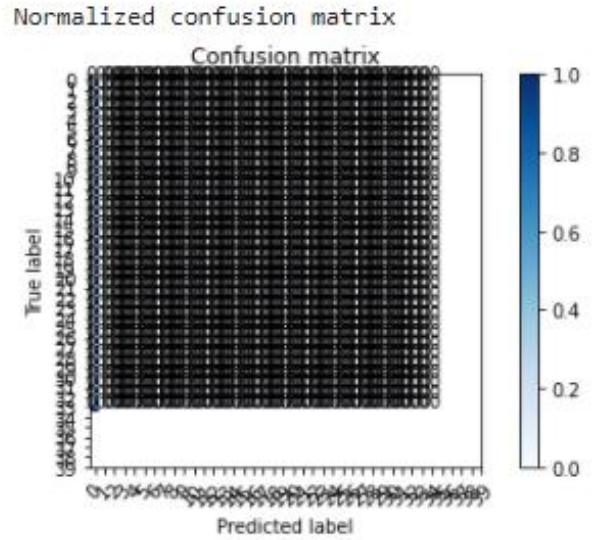
Untuk mengetahui performa klasifikasi maka dilakukan evaluasi terhadap metrik klasifikasi dan confusion matrix.

e) Classification Metrics

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2
1	0.00	0.00	0.00	4
2	0.00	0.00	0.00	2
3	0.00	0.00	0.00	4
4	0.00	0.00	0.00	2
5	0.00	0.00	0.00	2
6	0.00	0.00	0.00	1
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	3
9	0.00	0.00	0.00	5
10	0.00	0.00	0.00	2
11	0.00	0.00	0.00	0
12	0.00	0.00	0.00	1
13	0.00	0.00	0.00	3
14	0.00	0.00	0.00	0
15	0.00	0.00	0.00	1
16	0.00	0.00	0.00	4
17	0.00	0.00	0.00	1
18	0.00	0.00	0.00	3
19	0.00	0.00	0.00	0
20	0.00	0.00	0.00	2
21	0.00	0.00	0.00	2
22	0.00	0.00	0.00	1
23	0.00	0.00	0.00	2
24	0.00	0.00	0.00	0
25	0.00	0.00	0.00	2
26	0.00	0.00	0.00	2
27	0.00	0.00	0.00	1
28	0.00	0.00	0.00	1
29	0.00	0.00	0.00	4
30	0.00	0.00	0.00	1
31	0.00	0.00	0.00	4
32	0.00	0.00	0.00	2
33	0.00	0.00	0.00	0
34	0.00	0.00	0.00	3
35	0.00	0.00	0.00	3
36	0.00	0.00	0.00	2
37	0.00	0.00	0.00	3
38	0.00	0.00	0.00	0
39	0.00	0.00	0.00	1
micro avg	0.00	0.00	0.00	80
macro avg	0.00	0.00	0.00	80
weighted avg	0.00	0.00	0.00	80
samples avg	0.00	0.00	0.00	80

Gambar 10. Classification Metrics

f) Normalized Confusion Matrix



Gambar 11. Confusion Matrix

Berikut Contoh hasil klasifikasi pada data testing dapat dilihat pada Gambar.12.



Gambar 12. Contoh Hasil Klasifikasi Data Test

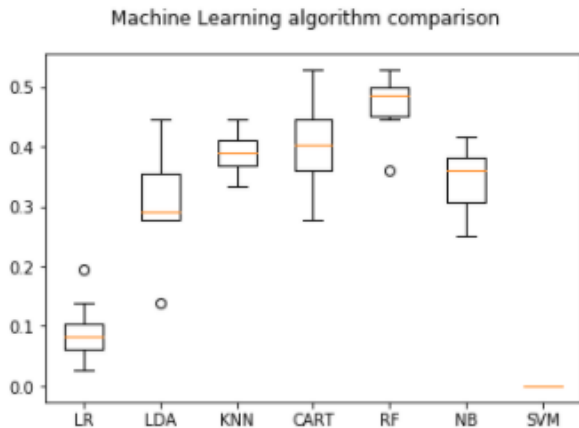
4.2. Hasil Metode Haralick dengan Random Forest

Hasil eksperimen dengan menggunakan tiga feature extraction Hu-Moment, Haralic dan Histogram dengan Algoritma Random Forest mendapatkan nilai akurasi minim sekali seperti pada data di bawah.

LR	0.088889
LDA	0.305556
KNN	0.388889
CART	0.405556
RF	0.475000
NB	0.341667
SVM	0.000000

Table 2. Hasil Akurasi

Berikut ini boxplot untuk perbandingan ke 7 Metodenya:



Gambar 13. Boxplot

Dari hasil komparasi bahwa metode Logistic Regression merupakan metode yang lumayan baik di antara yang lain karena menghasilkan Akurasi paling tinggi. Untuk eksperimen selanjutnya dapat dilakukan dengan melakukan perbandingan antara penggunaan feature extraction dan algoritma lain untuk mencari nilai akurasi yang paing tinggi.

5. KESIMPULAN

Dalam penelitian kali ini yaitu Feature Extractin dengan Hu-Moment, Haralick, Histogram dengan Algoritma Random Forest dan metode klasifikasi dengan menggunakan Convolutional Neural Network kurang baik untuk menentukan kebenaran dari klasifikasi citra objek. Hal ini terbukti dengan hasil akurasi Feature Haralick untuk identifikasi wajah sebesar 47%. Dan dengan menggunakan metode menggunakan Convolutional Neural Network(CNN) hasil akurasinya sangat kecil juga yaitu 0.05%. Hal ini membuktikan bahwa klasifikasi menggunakan metode CNN relatif kurang handal terhadap perubahan parameter yang dilakukan. Sebaiknya coba dilakukan dengan menggunakan data *training* yang baik dan optimal, maka subset dari data *training* tersebut juga akan menghasilkan klasifikasi yang baik.

DAFTAR PUSTAKA

[1] dan R. I. Y. Garis K, I. Santoso, “Klasifikasi Citra dengan Matriks Ko-Okurensi Aras Keabuan (Gray Level Co-Occurrence Matrix-GLCM) pada Limakelas Biji-Bijian,” 2011.

[2] I. Permatasari dan T. Sutojo, “Pengenalan Ciri Garis Telapak Tangan Menggunakan Ekstraksi Fitur (GLCM) dan Metode (KNN),” 2016.

[3] D. Syahid, Jumadi, and D. Nursantika, “Sistem Klasifikasi Jenis Tanaman Hias Daun Philodendron Menggunakan Metode K-Nearest Neighbor (KNN) Berdasarkan Nilai Hue, Saturation, Value (HSV),” *JOIN*, vol. I, no. 1, pp. 20–23, 2016.

[4] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.

[5] X. He, Z. Li, X. Liu, and T. Zhang, “Log facies recognition based on convolutional neural network,” *Shiyou Diqu Wuli Kantan/Oil Geophys. Prospect.*, vol. 54, no. 5, pp. 1159–1165, 2019, doi: 10.13810/j.cnki.issn.1000-7210.2019.05.024.

[6] S. Wasista, B. S. B. D, and S. A. Putra, “Sistem Pengenalan Wajah Pada Mesin Absensi Mahasiswa Menggunakan Metode PCA Dan DTW,” *13th Ind. Electron. Semin. 2011 (IES 2011)*, vol. 2011, no. Ies, pp. 224–229, 2011.

[7] S. Guo, S. Chen, and Y. Li, “Face recognition based on convolutional neural network & support vector machine,” *2016 IEEE Int. Conf. Inf. Autom. IEEE ICIA 2016*, no. August, pp. 1787–1792, 2017, doi: 10.1109/ICInfA.2016.7832107.

[8] G. Zou, G. Fu, M. Gao, J. Pan, and Z. Liu, “A new approach for small sample face recognition with pose variation by fusing Gabor encoding features and deep features,” *Multimed. Tools Appl.*, vol. 79, no. 31–32, pp. 23571–23598, 2020, doi: 10.1007/s11042-020-09076-1.

[9] S. Singh, D. Singh, and V. Yadav, “Face recognition using hog feature extraction and svm classifier,” *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 9, pp. 6437–6440, 2020, doi: 10.30534/ijeter/2020/244892020.

[10] K. Fukushima, “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position,” *Biol. Cybern.*, 1980.

[11] D. Stathakis, “How Many Hidden Layers And Nodes,” *Int. J. Remote Sens.*, 2008.

[12] Stanford University, “An Introduction to Convolutional Neural Network,” *Vis. Imaging Sci. Technol. Lab, Stanford Univ. [Online]*.

[13] J. T. Springenberg, “A. Dosovitskiy, T. Brox and M. Riedmiller, “Striving For Simplicity: The All Convolutional Net,” *ICLR 2015*, 2015.

[14] L. Ratnawati and D. R. Sulistyaningrum, “Penerapan Random Forest untuk Mengukur Tingkat Keparahan Penyakit pada Daun Apel,”

vol. 8, no. 2, pp. A71–A77, 2019.

- [15] S. Gayathri, A. K. Krishna, V. P. Gopi, and P. Palanisamy, “Automated Binary and Multiclass Classification of Diabetic Retinopathy Using Haralick and Multiresolution Features,” *IEEE Access*, vol. 8, pp. 57497–57504, 2020, doi: 10.1109/ACCESS.2020.2979753.
- [16] D. Fransiska Amalia Kurniawan, “Analisis dan Implementasi Random Forest dan Regression Tree (CART) Untuk Klasifikasi pada Misuse Intrusion Detection System,” *Fak. Tek. Inform.*, no. Data Mining, pp. 1–7, 2011.