



Eksperimen *Layer Pooling* menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode CNN

Yovi Pratama¹, Errissya Rasywir^{1,*}, Fachruddin², Desi Kisbianty³, Beni Irawan²

¹ Computer Science Faculty, Informatics Engineering Study Program, Universitas Dinamika Bangsa, Jambi, Indonesia

² Computer Science Faculty, Information System Study Program, Universitas Dinamika Bangsa, Jambi, Indonesia

³ Computer Science Faculty, Computer System Study Program, Universitas Dinamika Bangsa, Jambi, Indonesia

Email: ¹yovi.pratama@gmail.com, ^{2,*}errissya.rasywir@gmail.com, ³fachruddin.stikom@gmail.com, ⁴desikisbianty@unama.ac.id
⁵beben_delpiero@yahoo.co.id

Email Penulis Koresponden: errissya.rasywir@gmail.com

Submitted: 11/06/2023; Accepted: 29/06/2023; Published: 29/06/2023

Abstrak—*Deep Learning* khususnya *Convolutional Neural Network* (CNN) terbukti handal dalam mengolah data dari berbagai platform bahasa pemrograman dengan memanfaatkan *deep learning*. Dalam penelitian ini, kami melakukan modifikasi dengan perhitungan variansi statistik. Modifikasi yang dilakukan adalah mengganti perhitungan pada *Pooling Layer* yang umumnya menggunakan dua rumus yaitu *max pooling* dan *average pooling*. Kami menggunakan standar deviasi untuk mengubah nilai intensitas gambar yang dikurangi. Dengan eksperimen penelitian yang dibangun diharapkan mampu melakukan pengenalan wajah sebagai indikator pengujian modifikasi. Eksperimen *Layer Pooling* menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode CNN ini antara lain jenis dataset yang digunakan adalah dataset Aberdeen https://pics.stir.ac.uk/2D_face_sets.htm. Dari hasil eksperimen yang dilakukan diperoleh bahwa nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer *Adagrad* senilai 77.844% untuk penggunaan *max pooling* dan 79.844% untuk *pooling* dengan standar deviasi. Untuk fungsi aktivasi *Selu* dan optimizer *RMSprop* senilai 77.986% untuk penggunaan *max pooling* dan 75.986% untuk *pooling* dengan standar deviasi. Nilai tertinggi dengan fungsi aktivasi *Softplus* dan optimizer *Sgd* senilai 77.844% untuk penggunaan *max pooling* dan 76.344% untuk *pooling* dengan standar deviasi. Untuk fungsi aktivasi *Tanh* dan optimizer *Adadelta* senilai 87.844% untuk penggunaan *max pooling* dan 85.844% untuk *pooling* dengan standar deviasi. Fungsi aktivasi *Elu* dan optimizer *Adam* senilai 87.853% untuk penggunaan *max pooling* dan 85.285% untuk *pooling* dengan standar deviasi. Dengan menggunakan fungsi aktivasi *Elu* dan optimizer *Adamax* senilai 87.842% untuk penggunaan *max pooling* dan 86.242% untuk *pooling* dengan standar deviasi. Nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer *Nadam* senilai 87.845% untuk penggunaan *max pooling* dan 86.345% untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel. Dari keseluruhan eksperimen disebutkan bahwa penggunaan *pooling* dengan teknik nilai tertinggi atau *max pooling* masih memberikan nilai yang lebih baik dibanding menggunakan perhitungan standar deviasi dengan hasil tuning terbaik adalah menggunakan fungsi aktivasi *Elu* dan Optimiser *Adam* yakni sebesar 87.853%.

Kata Kunci: CNN; Citra; Eksperimen; Standar Deviasi; *Deep Learning*

Abstract—*Deep Learning*, especially the *Convolutional Neural Network* (CNN) has proven to be reliable in processing data from various programming language platforms by utilizing *deep learning*. In this study, we modified it by calculating the statistical variance. The modifications made are replacing calculations on the *Pooling Layer* which generally use two formulas, namely *max pooling* and *average pooling*. We use the standard deviation to change the reduced image intensity value. With the research experiments built, it is expected to be able to perform facial recognition as an indicator for testing modifications. The *Layer Pooling* experiment uses the Standard Deviation for Classifying Face Image Datasets with the CNN Method, including the type of dataset used is the Aberdeen dataset https://pics.stir.ac.uk/2D_face_sets.htm. From the results of the experiments conducted, it was found that the highest value was using the *Elu* activation function and the *Adagrad* optimizer worth 77.844% for *max pooling* and 79.844% for *pooling* with a standard deviation. The *Cellu* activation function and the *RMSprop* optimizer are 77.986% for *max pooling* and 75.986% for *pooling* with a standard deviation. The highest score with the *Softplus* activation function and the *Sgd* optimizer is 77.844% for *max pooling* usage and 76.344% for *pooling* with standard deviation. The *Tanh* activation function and the *Adadelta* optimizer are 87.844% for *max pooling* and 85.844% for *pooling* with a standard deviation. The *Elu* activation function and the *Adam* optimizer are 87.853% for the use of *max pooling* and 85.285% for *pooling* with a standard deviation. By using the *Elu* activation function and the *Adamax* optimizer, the value is 87.842% for *max pooling* and 86.242% for *pooling* with a standard deviation. The highest score is using the *Elu* activation function and the *Nadam* optimizer with a value of 87.845% for *max pooling* usage and 86.345% for using standard deviation calculations as pixel *pooling*. From all experiments it was stated that the use of *pooling* with the highest value technique or *max pooling* still gave a better value than using the standard deviation calculation with the best tuning results using the *Elu* activation function and *Adam's Optimiser*, which was 87.853%.

Keywords: CNN; Image; Experiment; Standard Deviation; *Deep Learning*

1. PENDAHULUAN

Pengenalan dengan menggunakan data biometrik, seperti citra wajah, sidik jari, telapak tangan, pola iris mata dan lainnya sangat dapat diandalkan. Penggunaan *biometric* tersebut dapat digunakan untuk proses otentifikasi, identifikasi, klasifikasi, deteksi, segmentasi dan pekerjaan lainnya untuk *task* pengenalan pola dari data citra [1]–[3]. Pengenalan pola dengan karakteristik fisiologis (seperti sidik jari, pola iris, wajah, dan siluet tangan) merupakan pengenalan pola citra yang memiliki karakteristik fisik yang relatif stabil [4]–[6]. Metode *deep learning* yang dikenali sebagai suatu kemajuan yang sangat pesat dalam ilmu kecerdasan buatan. *Deep Learning* mampu meningkatkan akurasi terhadap berbagai tugas pada berbagai fokus dalam pengolahan citra [7]–[10]. Dalam penelitian ini, akan

digunakan salah satu metode yang paling handal untuk tugas pengolahan citra dari seluruh varian pembelajaran mendalam [8], [10].

Metode terhandal dalam proses mengolah citra baik citra *biometric* atau non-biometrik adalah algoritma Jaringan syaraf Konvolusional atau dalam istilahnya disebut *Convolutional Neural Network (CNN)* [3], [11]–[13]. *Convolutional Neural Network* merupakan sebuah metode yang pembelajaran mesin (*machine learning*) yang dibangun dari perbaikan terus-menerus pada bagian jaringannya yang berlapis-lapis atau disebut *Multi-Layer Perceptron (MLP)* [12], [14]–[16]. Kelebihan dari metode CNN dikenal sebagai metode khusus pengolahan citra yang secara otomatis mengekstraksi fitur utama pada sebuah gambar tanpa melalui ekstraksi fitur manual. Selain itu, metode CNN juga terbukti mampu lebih efisien dibandingkan metode neural network lainnya terutama untuk memori dan kompleksitas dalam mengolah gambar digital [14], [17].

CNN yang merupakan bagian dari *Deep Neural Network*, memiliki banyak tingkat jaringan didalamnya [18]–[21]. Seperti yang dikatakan terdapat banyaknya jenis lapisan dalam algoritma CNN [3], [16], [22], penelitian ini akan melakukan modifikasi perhitungan dari dalam jenis-jenis lapisannya [12], [23], [24]. Algoritma berbasis Jaringan Syaraf terinspirasi dari cara kerja bagian otak manusia yaitu Visual Cortex. Korteks visual adalah bagian dari otak manusia yang bertanggung jawab untuk memproses informasi visual dari dunia luar. Ini memiliki berbagai lapisan dan setiap lapisan memiliki fungsinya sendiri yaitu setiap lapisan mengekstrak beberapa informasi dari gambar atau visual apa pun dan akhirnya semua informasi yang diterima dari setiap lapisan digabungkan dan gambar/visual tersebut ditafsirkan atau diklasifikasikan [8], [25], [26].

Demikian pula, CNN memiliki berbagai filter, dan setiap filter mengekstrak beberapa informasi dari gambar seperti tepi, berbagai jenis bentuk (vertikal, horizontal, bulat), dan kemudian semua ini digabungkan untuk mengidentifikasi gambar. Kelemahan JST konvensional adalah terlalu banyak perhitungan untuk model JST untuk melatih gambar berukuran besar dan berbagai jenis saluran gambar [27]. Kerugian berikutnya adalah tidak dapat menangkap semua informasi dari sebuah gambar sedangkan model CNN dapat menangkap dependensi spasial dari gambar. Alasan lainnya adalah ANN sensitif terhadap letak objek pada citra yaitu jika lokasi atau tempat objek yang sama berubah maka tidak akan dapat diklasifikasikan dengan baik. Dengan komponen CNN yang mempunyai model yang bekerja dalam dua langkah: ekstraksi fitur dan Klasifikasi. Secara umum, CNN terdiri dari lapisan input, lapisan konvolusi, *fully-connected layer*, dan lapisan output. Pada bagian *pooling* perhitungan dilakukan secara konvensional antara lain dengan metode *max pooling* dan *average pooling*. Sampai saat ini perhitungan pada *layer pooling* perlu dilakukan dengan metode *pooling* yang lain.

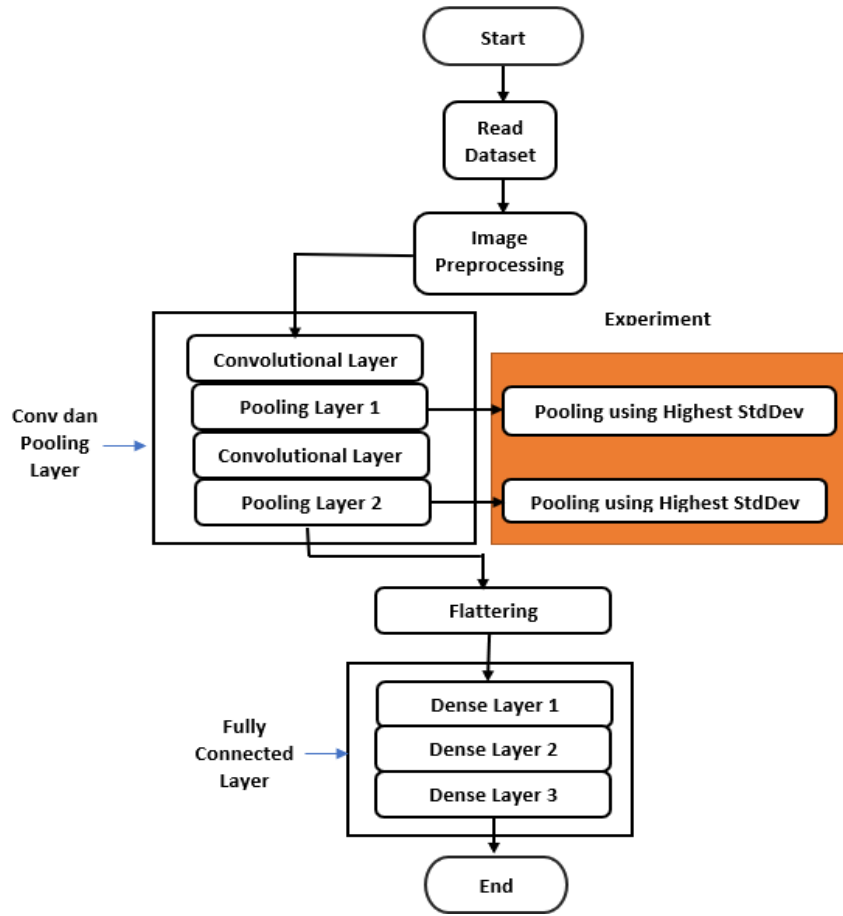
Beberapa Penelitian terdahulu yang melakukan penelitian mengenai modifikasi CNN pada *layer pooling* antara lain adalah "*Interpretation of intelligence in CNN-pooling processes: a methodological survey*" penelitian ini Artikel ini menyajikan pemilihan fitur global nilai, probabilitas, peringkat, dan domain yang diubah untuk ditangani di *layer pooling* [28]. Pada penelitian dengan judul "*Particular object retrieval with integral max-pooling of CNN activations*" menggunakan gambar integral untuk ditangani *max-pooling* pada aktivasi lapisan konvolusional, memungkinkan kita melokalkan objek yang cocok secara efisien [28]. Penelitian dengan judul "*Cross-convolutional-layer pooling for image recognition*" mengusulkan cara baru untuk mengekstraksi representasi gambar dari dua lapisan konvolusional [13]. Penelitian "*Mixed pooling for convolutional neural networks*" menggunakan *pooling* sebagai penggabungan operasi deterministik dengan prosedur stokastik dengan secara acak menggunakan metode pengumpulan maksimum dan pengumpulan rata-rata konvensional [12].

Dalam statistika, perhitungan dengan *max pooling* merupakan *pooling* piksel dari hasil pencarian nilai tertinggi dan *average pooling* sebagai nilai rerata. Pada penelitian ini, kami melakukan modifikasi yang dengan varian perhitungan statistik. Modifikasi yang dilakukan adalah mengganti perhitungan pada *Pooling Layer* yang pada umumnya menggunakan dua rumus yakni *max pooling* dan *average pooling*. Kami menggunakan standar deviasi untuk menggantikan nilai intensitas citra yang direduksi. Dengan eksperimen riset yang dibangun diharapkan dapat melakukan pengenalan wajah sebagai indikator pengujian modifikasi.

2. METODOLOGI PENELITIAN

2.2 Kerangka Kerja Penelitian

Batasan masalah Eksperimen Layer Pooling menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode CNN ini antara lain jenis dataset yang digunakan adalah dataset Aberdeen https://pics.stir.ac.uk/2D_face_sets.htm. Untuk Metode pengenalan citra yang digunakan adalah *Convolutional Neural Network (CNN)*. Evaluasi pengenalan yang digunakan adalah akurasi dan confusion matrix. Modifikasi yang dilakukan dengan mengubah rumus *max pooling* dan *average pooling* dengan rumus standar deviasi. Dengan tujuan antara lain adalah untuk melakukan Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia, serta Melakukan evaluasi dan analisis terhadap Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia. Kerangka kerja penelitian Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Klasifikasi Citra Wajah Manusia dapat dilihat pada Gambar 1 berikut, gambar ini merepresentasikan tahapan proses yang dilakukan dalam penelitian agar penelitian dapat berjalan dengan baik dan tujuan yang telah ditetapkan dapat tercapai.



Gambar 1. Bagan Alur Metodologi Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia

a. Penelitian Awal

Masalah yang dibahas adalah bagaimana Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia. Teori yang diambil tentang kode program, *deep learning*, *Convolutional neural network*, akan diujikan guna melakukan eksperimen yang dibahas dalam penelitian ini.

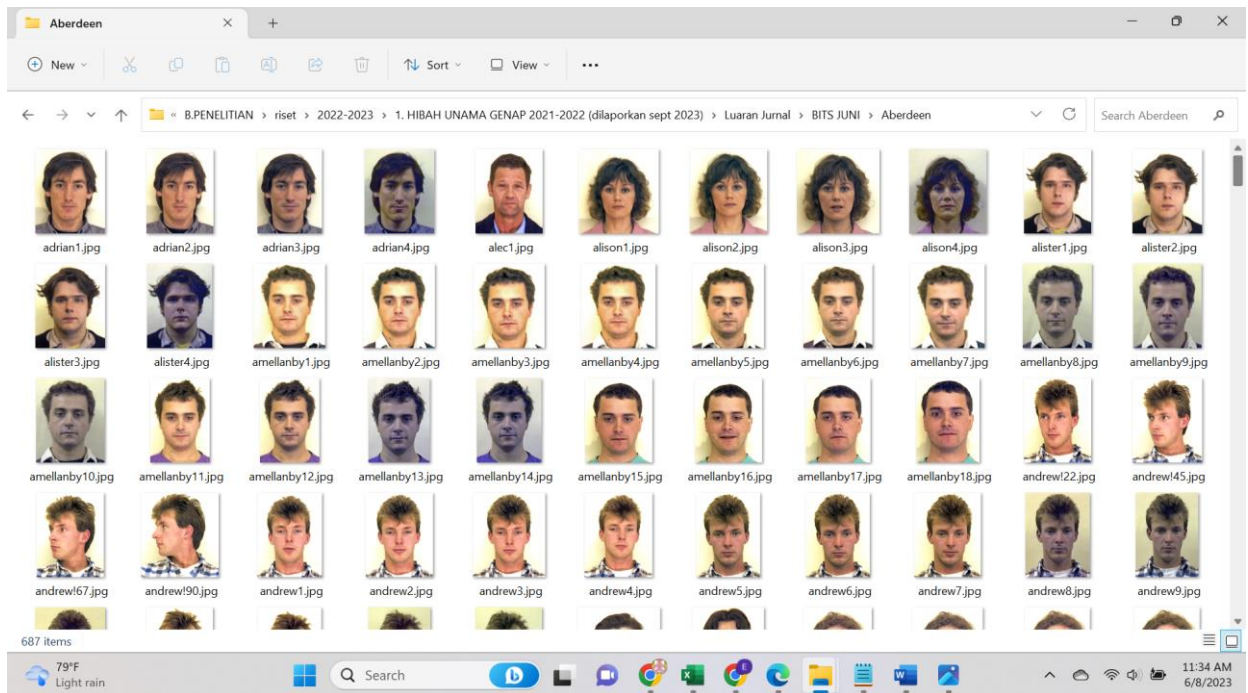
b. Pengumpulan Data Kode program untuk dipersiapkan dalam eksperimen.

Data diambil dari beberapa proyek perangkat lunak berupa kode program. Kode program ini akan ditransformasi menjadi dataset dengan *feature* adalah berupa term yang berasal dari sintak pemrograman. Selanjutnya data *testing* dan data *training* akan dibagi menjadi dua bagian yakni data *training* berasal dari kode program yang telah diberikan label sebagai kelas plagiat atau tidak.

```

1 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/818e-3d1-e0d4-4550-8ec1-a0e2a7628787_0-76_jpg.jpeg", "annotation": ""}
2 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/61c3-32be-805a-786d4a5c8c-b101c8130b65_8d6ee7884442939e454d88615", "annotation": ""}
3 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/aa5a997-b248-4724-9002-b29117725102_8rcMvSY7x5i1Uu61_jpg.jpeg", "annotation": ""}
4 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/c4efa88f-8218-4675-b5ce-30a03bf8b8f4_1.png", "annotation": ""}
5 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/dcc5b187-29fa-4734-83f4-3bc7a4117457_1f9131304d4d265684fae960106d1", "annotation": ""}
6 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/4ce4255a-831f-4330-8241-9988e8f9e27_2a63028312eb60800927f8352ba", "annotation": ""}
7 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/45ff-4dc-e24f-4239-b4cc-8d8e597f629a_2867c48f00008578-3199800-ima", "annotation": ""}
8 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/d5fe7eb-f17d-4bed-872b-17a4336eb17_2-hd9mj_jpg.jpeg", "annotation": ""}
9 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/5b6d8f2e-ae21-44ae-bc f1-19d8a8e98b65_3c4f3c819d8771e25a9b462ce2", "annotation": ""}
10 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/93927e0d-c77a-49b6-8c4d-a259d00a5868_3d---4K---2d-HD-DTS-ATMOS-48", "annotation": ""}
11 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/212a2218-8c0c-4530-8095-504ef81bc37_4_jpg.jpeg", "annotation": ""}
12 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/bfbdca44-4614-4cea-a0fa-e0a311b663d2_4ac27f35-8419-4c89-910a-a9e6", "annotation": ""}
13 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/49d8b065-f074-4f93-9f72-85c117cae877_05-families-fighting-for-cha", "annotation": ""}
14 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/b6596eb0-d5a8-432e-8a88-6f262af39484_0Shelpingfamiliesgetthead_0", "annotation": ""}
15 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/80c76616-ed44-4015-9935-9c6661c5c5d_5_job-Interview_0474_f_cemes", "annotation": ""}
16 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/a7417d6e-d342-469f-872c-8455addbbba_5adbc8a38f8d8940b64e-758", "annotation": ""}
17 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/c8cc5f4f-6684-443f-8f41-622e780fb59e_5b2d5c81a0000c504ce216_jpg", "annotation": ""}
18 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/4379b57c-a6e2-449a-8fb7-d58153da5b2b_5c8670b4917b67efef8c4f73842", "annotation": ""}
19 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/594e1406-fa78-4613-9e9d-4e79e20a2f44_6c9f5faff0f62545b0bf77d9355", "annotation": ""}
20 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/14d92e7b-1705-48ed-b755-777921f8fa9_074df77dce1c7344c8c39f593", "annotation": ""}
21 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/5dcb5230-c342-4f78-8c9a-37c35e852289_7c9e84ba6417cd45932c69f6128", "annotation": ""}
22 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/b4bf497b-424f-46ee-a039-34804a1992c1_7e09d2b4-1.jpg.jpeg", "annotation": ""}
23 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/98c726f1-44ef-42b7-99ef-ce14f4fb6fff_0811d1.jpg.jpeg", "annotation": ""}
24 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/0847266-e971-4077-b089-1936296bc3c_10-cover1d1-lana-27x358_j", "annotation": ""}
25 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/a7417d6e-d342-469f-872c-8455addbbba_12familyseparation-sub-arctic", "annotation": ""}
26 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/d7f85c01-117c-4e3d-a607-2690f8a1b397_0013.jpg.jpeg", "annotation": ""}
27 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/6ba70532-4198-4484-b262-4d83b79e960_13_Favorite_Memorable_Movie", "annotation": ""}
28 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/66fab77d-5988-4760-a346-ab29886c49c_014a8f16-c5ce-43fc-bf88-a895", "annotation": ""}
29 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/4cc4504c-d0d7-46d8-b619-f8e8003c910_14-scene5-From-mojito-and-y", "annotation": ""}
30 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/4d809740-8905-49a1-bd07-03d588b27f7f_16best-scenes14.jpg.jpeg", "annotation": ""}
31 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/130f1009-0a91-42cf-b4ee-6b8453e6534a_17de631fb19739677253ca30453", "annotation": ""}
32 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/e53eb4ea-9e38-4f94-a102-529a878b1fbd_18qdnvgv1c2.jpg.jpeg", "annotation": ""}
33 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/3301740-6008-436a-9d05-2e59772ca474_19c50632fa4d4f8b1580572964", "annotation": ""}
34 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/2c108780-c3aa-4659-a577-f4545cf999d_19-101-fax-scans_mocrop_w71", "annotation": ""}
35 {"content": "http://com.dataturks.a96-123.open.s3.amazonaws.com/2c9fa7b064277d86016431e33e4e0b3d/1ec3874f-ed0e-424b-8ec3-803913718979_22-reunification-migrant-far", "annotation": ""}
    
```

Gambar 2. Data Anotasi CitraDataset Aberdeen.



Gambar 3. Cuplikan Citra Dari Dataset Aberdeen

c. Merancang & Implementasi Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia.

Kegiatan ini berupa pengukuran perangkat lunak yang mana hasil pengukuran tersebut digunakan dalam melakukan Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Klasifikasi Citra Wajah Manusia. Kemudian hasil deteksi akan dievaluasi dengan menggunakan Akurasi.

$$s = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}} \quad (2)$$

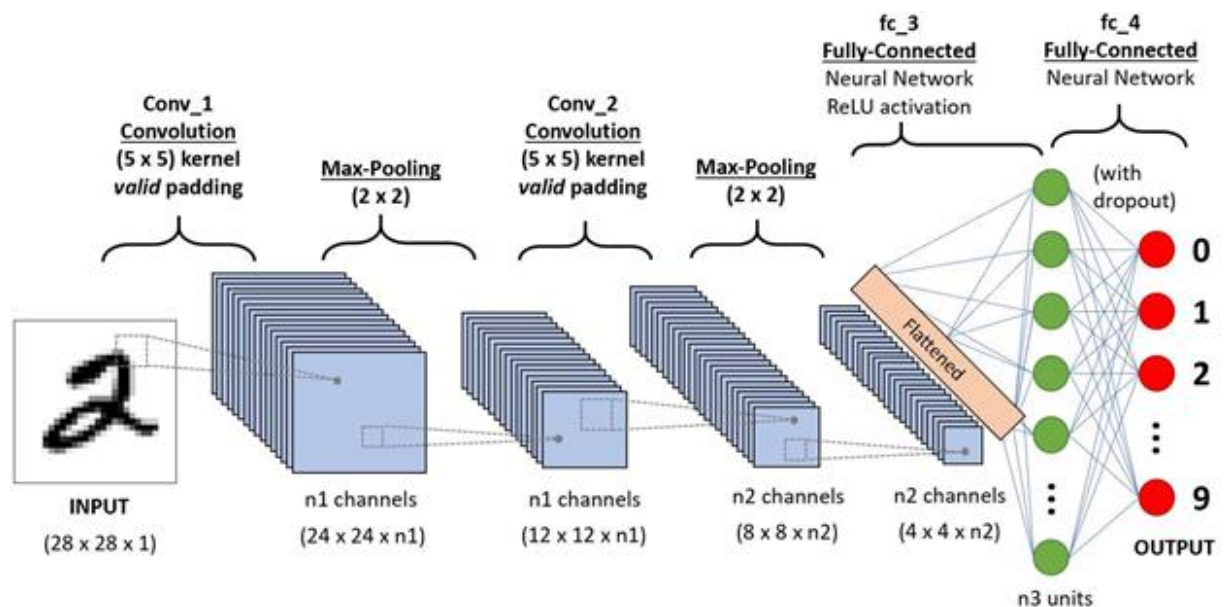
Keterangan :

S = Standar Deviasi Data Intensitas Citra yang di uji

X = Data Intensitas Citra yang di uji

n = Jumlah Semua Data Intensitas Citra yang di uji

Algoritma *Convolutional Neural Network* (CNN) ini memiliki jaringan terdiri dari lapisan input, 5 lapisan konvolusi, 3 *fully connected layer*, dan lapisan output dapat dilihat pada gambar berikut:



Gambar 4. Visualisasi Layer dalam Algoritma Convolutional Neural Network

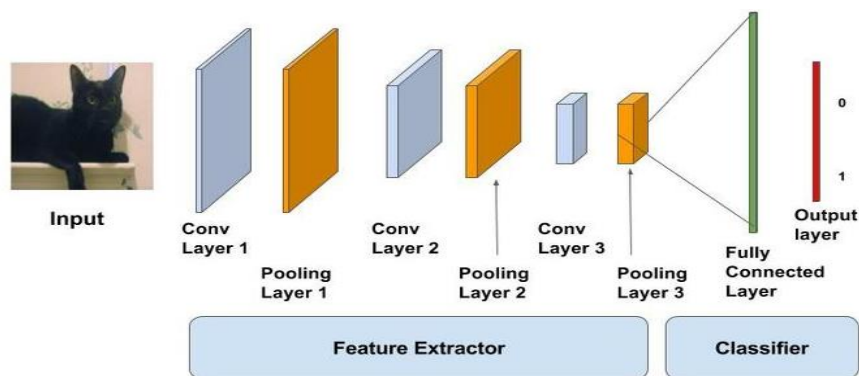
Pada gambar 7 di atas bagian yang layer bernama *Pooling Layer* pada umumnya dilakukan dengan menggunakan perhitungan *max layer* ataupun *average layer*. Namun modifikasi pada penelitian ini menggunakan perhitungan standar deviasi yang rumusnya telah dijelaskan pada langkah ke-3 sebelum poin ini.

d. Melakukan Eksperimen dengan Kode program

Kegiatan ini melakukan eksperimen dengan cara melakukan Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia. Pengenalan dilakukan dengan alur yang terdapat pada gambar 8. Setelah selesai di *training* selanjutnya proses *testing*. Proses testing ini terdiri dari model classifier dengan pengujian menggunakan data test untuk melakukan label prediction, selanjutnya dilakukan *detection* dan *metric evaluation* untuk mengevaluasi hasil eksperimen.

e. Hasil awal Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia.

Kegiatan ini pelaporan hasil eksperimen awal dari alur sistem pada gambar 8 dengan cara melakukan komparasi terhadap Kode program terhadap data Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia.



Gambar 5. Lapisan yang akan dikenakan Modifikasi (*feature Extraction*)

f. *Progress Report* eksperimen.

Kegiatan ini dilakukan untuk melaporkan langkah kegiatan apa saja yang telah dilakukan setelah eksperimen awal diperoleh dengan cara melakukan komparasi terhadap Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia.

g. Evaluasi Akhir Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia.

Pada kegiatan ini dilakukan evaluasi akhir terhadap Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Klasifikasi Citra Wajah Manusia.

3. HASIL DAN PEMBAHASAN

.Pada bagian ini, akan dijelaskan hasil dari eksperimen Layer Pooling Experiment with Standard Deviation for Facial Image Dataset Classification using CNN.

3.1 Analisis Hasil Implementasi *Modifikasi Layer Pooling*

Pada bagian ini dijelaskan hasil implementasi dari alur metodologi Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia dengan dataset Aberdeen. Eksperimen yang kami lakukan adalah di bagian *Pooling Layer* yakni pemilihan piksel untuk mereduksi citra terpilih sebagai fitur menggunakan menggunakan perhitungan standar deviasi. Pada umumnya, bagian yang layer bernama *Pooling Layer* dilakukan dengan menggunakan perhitungan *max layer* ataupun *average layer*. Namun modifikasi pada penelitian ini menggunakan perhitungan standar deviasi. Selain itu kami juga melakukan percobaannya pada bagian *fully Connected Layer* dengan menggunakan beberapa jenis fungsi aktivasi dan optimizer sebagai proses tuning untuk mendapatkan jaringan terbaik dalam melakukan klasifikasi dataset aberdeen.

```

for dirname, _, filenames in os.walk('/aberdeen/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]]
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80, facecolor = 'w', edgecolor = 'k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()

```

Gambar 6. Cuplikan Implementasi Dari Proses Inisialisasi Pembacaan Dataset Aberdeen

Gambar 6 di atas cuplikan implementasi dari proses inisialisasi pembacaan dataset aberdeen yang dilakukan dalam penelitian modifikasi *layer pooling* menggunakan standar deviasi pada cnn untuk pengenalan citra wajah manusia ini. proses ini adalah langkah awal program yang dibangun dengan bahasa pemrograman Python sebagai tools pengujian eksperimen. Dalam bahasa pemrograman ini sudah banyak terdapat library dan fungsi yang mendukung pekerjaan untuk proses inisialisasi pembacaan dataset Aberdeen pada penelitian ini.

```

from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

```

Gambar 7. Cuplikan Implementasi Dari Proses Inisialisasi Praproses Dataset Aberdeen

Gambar 7 di atas Cuplikan Implementasi Dari Proses Inisialisasi Praproses Dataset Aberdeen yang dilakukan dalam penelitian modifikasi *layer pooling* menggunakan standar deviasi pada CNN untuk pengenalan citra wajah manusia ini. proses ini adalah langkah awal program yang dibangun dengan bahasa pemrograman Python sebagai tools pengujian eksperimen. Dalam bahasa pemrograman ini sudah banyak terdapat library dan fungsi yang mendukung pekerjaan untuk proses Cuplikan Implementasi Dari Proses Inisialisasi Praproses Dataset Aberdeen pada penelitian ini.

```

[ ] model = models.Sequential()
    model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(432, 528, 3)))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))

```

Gambar 8. Cuplikan Implementasi Dari Feature Extraction dengan Convolutional Layer dan Pooling Layer serta Fungsi Aktivasi

Gambar 8 di atas Cuplikan Implementasi Dari Feature Extraction dengan Convolutional Layer dan Pooling Layer serta Fungsi Aktivasi yang dilakukan dalam penelitian modifikasi *layer pooling* menggunakan standar deviasi pada CNN untuk pengenalan citra wajah manusia ini. proses ini adalah langkah awal program yang dibangun dengan bahasa pemrograman Python sebagai tools pengujian eksperimen. Dalam bahasa pemrograman ini sudah banyak terdapat library dan fungsi yang mendukung pekerjaan untuk proses Cuplikan Implementasi Dari Feature Extraction dengan Convolutional Layer dan Pooling Layer serta Fungsi Aktivasi pada penelitian ini.

```
[ ] model.add(layers.Flatten())
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(10))
```

Gambar 9. Cuplikan Implementasi Proses *Flattering* Output dari *Feature Layer* .

Gambar 9 di atas Cuplikan Implementasi Proses *Flattering* Output dari *Feature Layer* yang dilakukan dalam penelitian modifikasi *layer pooling* menggunakan standar deviasi pada CNN untuk pengenalan citra wajah manusia ini. proses ini adalah langkah awal program yang dibangun dengan bahasa pemrograman Python sebagai tools pengujian eksperimen. Dalam bahasa pemrograman ini sudah banyak terdapat *library* dan fungsi yang mendukung pekerjaan untuk proses Cuplikan Implementasi Proses *Flattering* Output dari *Feature Layer* pada penelitian ini.

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

history = model.fit(train_images, train_labels, epochs=10,
                   validation_data=(test_images, test_labels))
```

Gambar 10. Cuplikan Implementasi *Tunning Optimizer* pada *Fully-Connected Layer*.

Gambar 10 di atas Cuplikan Implementasi *Tunning Optimizer* pada *Fully-Connected Layer* yang dilakukan dalam penelitian modifikasi *layer pooling* menggunakan standar deviasi pada CNN untuk pengenalan citra wajah manusia ini. proses ini adalah langkah awal program yang dibangun dengan bahasa pemrograman Python sebagai tools pengujian eksperimen. Dalam bahasa pemrograman ini sudah banyak terdapat *library* dan fungsi yang mendukung pekerjaan untuk proses Cuplikan Implementasi *Tunning Optimizer* pada *Fully-Connected Layer* pada penelitian ini.

3.2 Analisis Hasil Eksperimen pada *Modifikasi Layer Pooling dan Fully Connected Layer*

Pada bagian ini ditampilkan hasil eksperimen dari klasifikasi dataset Aberdeen dengan menggunakan algoritma *Convolutional Neural network*. Sebagaimana yang diketahui, algoritma *Convolutional Neural Network* adalah salah satu pembelajaran mesin yang lebih mendalam dibanding algoritma pada machine learning pada umumnya. Dimana Algoritma *Convolutional Neural network* memiliki dua lapisan utama yakni lapisan yang bekerja sebagai feature Extraction yang berisi *Convolution Layer* dan *Pooling Layer*, Lalu lapisan berikutnya yang bertugas melakukan pekerjaan klasifikasi untuk dataset pada penelitian ini. Lapisan ini disebut juga *Fully Connected Layer* yang berisi layer-layer pada umumnya sebuah jaringan syaraf tiruan. Pekerjaan *experiment* kami berada di kedua layer ini.

Pada layer *feature Extraction* kami mencoba melakukan perhitungan yang berbeda pada CNN biasa. Pada CNN biasa, *Pooling Layer* biasanya menggunakan Teknik *pooling* dengan mengambil nilai piksel terbesar dari sebuah blok citra keluaran *convolutional* yang berukuran 2 x 2, teknik ini disebut dengan *Max Pooling*. Ada juga teknik pooling dengan menggunakan *Average Pooling*, pada teknik ini dihitung nilai rata-rata dari 4 piksel yang berada dalam blok ukuran 2 x 2. Namun dalam penelitian ini, pada *layer pooling* dalam jaringan CNN kami, digunakan nilai standar deviasi yang di hitung dengan rumus pada nomor 1 untuk memperoleh nilai hasil *pooling*. Namun, dalam eksperimennya kami tetap melakukan pooling secara normal menggunakan nilai *max pooling*. Untuk nilai *average pooling* kami belum menggunakannya sebagai eksperimen. Untuk eksperimen pada bagian *fully connected Layer*, juga dilakukan, kami melakukan tuning pada bagian *Activation Function* dan juga pada bagian optimizer. Beberapa jenis *Activation Function* yang kami gunakan dalam tuning antara lain adalah *Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid*. Selanjutnya, untuk jenis optimizer yang digunakan adalah Adagrad, RMSprop, Sgd, Adadelata, Adam, Adamax dan Nadam.

Tabel 1. Hasil Eksperimen *Pooling Layer* dan *Activation Function* dan *Optimizer* dengan Adagrad.

No	Experiment in Fully Connected Layer		Percentage Accuration	
	Activation Function	Optimizer	Max Pooling	StdDev Pooling
1	Relu	Adagrad	76.742	78.742
2	Linear	Adagrad	71.132	73.132
3	Sigmoid	Adagrad	75.787	77.787
4	Softmax	Adagrad	76.742	78.742
5	Tanh	Adagrad	75.123	77.123
6	Elu	Adagrad	77.844	79.844
7	Selu	Adagrad	76.234	78.234
8	Softplus	Adagrad	77.845	79.845
9	Softsign	Adagrad	72.573	74.573
10	Hard_Sigmoid	Adagrad	75.123	77.123

Tabel 1 di atas adalah hasil eksperimen *Pooling Layer* dan *Activation Function* dan *optimizer* pada bagian *fully connected Layer*. Tabel diatas adalah hasil *tuning* jenis *Activation Function Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid* dengan jenis oprimizer Adagrad. Sebelumnya pada bagian *feature layer* dibagian *pooling* kami melakukan reduksi dimensi dari blok 2 x 2 piksel dengan menggunakan *max pooling* yang kami bandingkan dengan pooling dengan standar deviasi. Dari hasil percobaan tersebut dihasilkan nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer *Adagrad* senilai **77.844%** untuk penggunaan max pooling dan **79.844 %** untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel.

Tabel 2. Hasil Eksperimen *Pooling Layer* dan *Activation Function* dan Optimizer dengan Adagrad.

No	Experiment in Fully Connected Layer		Percentage Accuration	
	Activation Function	Optimizer	Max Pooling	StdDev Pooling
1	Relu	RMSprop	77.853	75.853
2	Linear	RMSprop	76.742	74.742
3	Sigmoid	RMSprop	77.856	75.856
4	Softmax	RMSprop	75.489	73.489
5	Tanh	RMSprop	76.896	74.896
6	Elu	RMSprop	75.268	73.268
7	Selu	RMSprop	77.986	75.986
8	Softplus	RMSprop	77.974	75.974
9	Softsign	RMSprop	72.976	70.976
10	Hard_Sigmoid	RMSprop	75.254	73.254

Tabel 2 di atas adalah hasil eksperimen *Pooling Layer* dan *Activation Function* dan *optimizer* pada bagian *fully connected Layer* dengan hasil *tuning* jenis *Activation Function Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid* menggunakan jenis oprimizer RMSprop. Sebelumnya pada bagian *feature layer* dibagian *pooling* kami melakukan reduksi dimensi dari blok 2 x 2 piksel dengan menggunakan *max pooling* yang kami bandingkan dengan pooling dengan standar deviasi. Dari hasil percobaan tersebut dihasilkan nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Selu* dan optimizer RMSprop senilai **77.986%** untuk penggunaan max pooling dan **75.986%** untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel.

Tabel 3. Hasil Eksperimen *Pooling Layer* dan *Activation Function* dan Optimizer dengan Adagrad.

No	Experiment in Fully Connected Layer		Percentage Accuration	
	Activation Function	Optimizer	Max Pooling	StdDev Pooling
1	Relu	Sgd	76.742	75.242
2	Linear	Sgd	71.132	69.632
3	Sigmoid	Sgd	75.787	74.287
4	Softmax	Sgd	76.422	74.922
5	Tanh	Sgd	75.787	74.287
6	Elu	Sgd	76.734	75.234
7	Selu	Sgd	75.123	73.623
8	Softplus	Sgd	77.844	76.344
9	Softsign	Sgd	72.573	71.073
10	Hard_Sigmoid	Sgd	75.123	73.623

Tabel 3 di atas adalah hasil eksperimen *Pooling Layer* dan *Activation Function* dan *optimizer* pada bagian *fully connected Layer* dengan hasil *tuning* jenis *Activation Function Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid* menggunakan jenis oprimizer Sgd. Sebelumnya pada bagian *feature layer* dibagian *pooling* kami melakukan reduksi dimensi dari blok 2 x 2 piksel dengan menggunakan *max pooling* yang kami bandingkan dengan pooling dengan standar deviasi. Dari hasil percobaan tersebut dihasilkan nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Softplus* dan optimizer Sgd senilai **77.844%** untuk penggunaan max pooling dan **76.344%** untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel.

Tabel 4. Hasil Eksperimen *Pooling Layer* dan *Activation Function* dan Optimizer dengan Adagrad.

No	Experiment in Fully Connected Layer		Percentage Accuration	
	Activation Function	Optimizer	Max Pooling	StdDev Pooling
1	Relu	Adadelta	86.456	84.456
2	Linear	Adadelta	85.787	83.787
3	Sigmoid	Adadelta	86.742	84.742
4	Softmax	Adadelta	85.734	83.734
5	Tanh	Adadelta	87.844	85.844
6	Elu	Adadelta	82.573	80.573

7	Selu	Adadelta	86.234	84.234
8	Softplus	Adadelta	87.845	85.845
9	Softsign	Adadelta	82.573	80.573
10	Hard_Sigmoid	Adadelta	85.462	83.462

Tabel 4 di atas adalah hasil eksperimen *Pooling Layer* dan *Activation Function* dan *optimizer* pada bagian *fully connected Layer* dengan hasil *tuning* jenis *Activation Function Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid* menggunakan jenis oprimizer Adadelta. Sebelumnya pada bagian *feature layer* dibagian *pooling* kami melakukan reduksi dimensi dari blok 2 x 2 piksel dengan menggunakan *max pooling* yang kami bandingkan dengan *pooling* dengan standar deviasi. Dari hasil percobaan tersebut dihasilkan nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Tanh* dan optimizer Adadelta senilai **87.844%** untuk penggunaan *max pooling* dan **85.844%** untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel.

Tabel 5. Hasil Eksperimen *Pooling Layer* dan *Activation Function* dan *Optimizer* dengan Adagrad.

No	Experiment in Fully Connected Layer		Percentage Accuration	
	Activation Function	Optimizer	Max Pooling	StdDev Pooling
1	Relu	Adam	85.324	85.252
2	Linear	Adam	86.752	84.357
3	Sigmoid	Adam	85.857	85.623
4	Softmax	Adam	87.123	83.623
5	Tanh	Adam	85.123	86.353
6	Elu	Adam	87.853	85.285
7	Selu	Adam	86.785	86.345
8	Softplus	Adam	87.845	80.734
9	Softsign	Adam	82.234	84.245
10	Hard_Sigmoid	Adam	85.745	84.245

Tabel 5 di atas adalah hasil eksperimen *Pooling Layer* dan *Activation Function* dan *optimizer* pada bagian *fully connected Layer* dengan hasil *tuning* jenis *Activation Function Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid* menggunakan jenis oprimizer Adam. Sebelumnya pada bagian *feature layer* dibagian *pooling* kami melakukan reduksi dimensi dari blok 2 x 2 piksel dengan menggunakan *max pooling* yang kami bandingkan dengan *pooling* dengan standar deviasi. Dari hasil percobaan tersebut dihasilkan nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer Adam senilai **87.853%** untuk penggunaan *max pooling* dan **85.285%** untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel.

Tabel 6. Hasil Eksperimen *Pooling Layer* dan *Activation Function* dan *Optimizer* dengan Adagrad.

No	Experiment in Fully Connected Layer		Percentage Accuration	
	Activation Function	Optimizer	Max Pooling	StdDev Pooling
1	Relu	Adamax	86.842	85.242
2	Linear	Adamax	81.845	80.245
3	Sigmoid	Adamax	85.956	84.356
4	Softmax	Adamax	86.231	84.631
5	Tanh	Adamax	85.754	84.154
6	Elu	Adamax	87.842	86.242
7	Selu	Adamax	86.842	85.242
8	Softplus	Adamax	87.732	86.132
9	Softsign	Adamax	82.473	80.873
10	Hard_Sigmoid	Adamax	85.753	84.153

Tabel 6 atas adalah hasil eksperimen *Pooling Layer* dan *Activation Function* dan *optimizer* pada bagian *fully connected Layer* dengan hasil *tuning* jenis *Activation Function Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid* menggunakan jenis oprimizer Adamax. Sebelumnya pada bagian *feature layer* dibagian *pooling* kami melakukan reduksi dimensi dari blok 2 x 2 piksel dengan menggunakan *max pooling* yang kami bandingkan dengan *pooling* dengan standar deviasi. Dari hasil percobaan tersebut dihasilkan nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer Adamax senilai **87.842%** untuk penggunaan *max pooling* dan **86.242%** untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel.

Tabel 7. Hasil Eksperimen *Pooling Layer* dan *Activation Function* dan *Optimizer* dengan Adagrad.

No	Experiment in Fully Connected Layer		Percentage Accuration	
	Activation Function	Optimizer	Max Pooling	StdDev Pooling
1	Relu	Nadam	82.523	81.023
2	Linear	Nadam	85.753	84.253

3	Sigmoid	Nadam	85.845	84.345
4	Softmax	Nadam	86.234	84.734
5	Tanh	Nadam	85.754	84.254
6	Elu	Nadam	87.845	86.345
7	Selu	Nadam	85.634	84.134
8	Softplus	Nadam	86.742	85.242
9	Softsign	Nadam	85.744	84.244
10	Hard_Sigmoid	Nadam	87.853	86.353

Tabel 7 atas adalah hasil eksperimen *Pooling Layer* dan *Activation Function* dan *optimizer* pada bagian *fully connected Layer* dengan hasil *tuning* jenis *Activation Function Relu, Linear, Sigmoid, Softmax, Tanh, Elu, Selu, Softplus, Softsign* dan *Hard_Sigmoid* menggunakan jenis oprimizer Nadam. Sebelumnya pada bagian *feature layer* dibagian *pooling* kami melakukan reduksi dimensi dari blok 2 x 2 piksel dengan menggunakan *max pooling* yang kami bandingkan dengan *pooling* dengan standar deviasi. Dari hasil percobaan tersebut dihasilkan nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer Nadam senilai **87.845%** untuk penggunaan *max pooling* dan **86.345%** untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel. Nilai ini menunjukkan bahwa hasil *tuning* terbaik pada jaringan CNN yang dibangun untuk dataset Aberdeen ini yakni menggunakan *Elu* dan Nadam. Dan Penggunaan nilai standar deviasi sebagai *pooling* pada piksel citra di *layer pooling* belum memberikan nilai yang lebih baik dibandingkan dengan penggunaan nilai *Max Pooling*.

4. KESIMPULAN

Eksperimen *Layer Pooling* menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode CNN ini antara lain jenis dataset yang digunakan adalah dataset Aberdeen https://pics.stir.ac.uk/2D_face_sets.htm. Untuk Metode pengenalan citra yang digunakan adalah *Convolutional Neural Network (CNN)*. Evaluasi pengenalan yang digunakan adalah akurasi. Modifikasi yang dilakukan dengan mengubah rumus *max pooling* dan *average pooling* dengan rumus standar deviasi. Dengan tujuan untuk evaluasi dan analisis terhadap Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia. Dari hasil eksperimen yang dilakukan diperoleh bahwa nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer *Adagrad* senilai 77.844% untuk penggunaan *max pooling* dan 79.844 % untuk *pooling* dengan standar deviasi. Untuk fungsi aktivasi *Selu* dan optimizer *RMSprop* senilai 77.986% untuk penggunaan *max pooling* dan 75.986% untuk *pooling* dengan standar deviasi. Nilai tertinggi dengan fungsi aktivasi *Softplus* dan optimizer *Sgd* senilai 77.844% untuk penggunaan *max pooling* dan 76.344% untuk *pooling* dengan standar deviasi. Untuk fungsi aktivasi *Tanh* dan optimizer *Adadelta* senilai 87.844% untuk penggunaan *max pooling* dan 85.844% untuk *pooling* dengan standar deviasi. Fungsi aktivasi *Elu* dan optimizer *Adam* senilai 87.853% untuk penggunaan *max pooling* dan 85.285% untuk *pooling* dengan standar deviasi. Dengan menggunakan fungsi aktivasi *Elu* dan optimizer *Adamax* senilai 87.842% untuk penggunaan *max pooling* dan 86.242% untuk *pooling* dengan standar deviasi. Nilai tertinggi adalah dengan menggunakan fungsi aktivasi *Elu* dan optimizer Nadam senilai 87.845% untuk penggunaan *max pooling* dan 86.345% untuk penggunaan perhitungan standar deviasi sebagai *pooling* piksel. Dari keseluruhan eksperimen disebutkan bahwa penggunaan *pooling* dengan teknik nilai tertinggi atau *max pooling* masih memberikan nilai yang lebih baik dibanding menggunakan perhitungan standar deviasi dengan hasil *tuning* terbaik adalah menggunakan fungsi aktivasi *Elu* dan Optimiser *Adam* yakni sebesar 87.853%.

ACKNOWLEDGMENT

Terima kasih kepada Sivitas Akademika UNAMA serta Yayasan Dinamika Bangsa atas dukungan moril dan materil sehingga penulisan artikel ilmiah ini dapat diselesaikan dengan baik. Artikel ini adalah luaran wajib dari penelitian Hibah UNAMA dengan judul “Modifikasi *Layer Pooling* menggunakan Standar Deviasi pada CNN untuk Pengenalan Citra Wajah Manusia”.

REFERENCES

- [1] Fachruddin, Saparudin, E. Rasywir, and Y. Pratama, “Network and layer experiment using convolutional neural network for content based image retrieval work,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 20, no. 1, pp. 118–128, 2022, doi: 10.12928/TELKOMNIKA.v20i1.19759.
- [2] Y. Pratama and E. Rasywir, “Eksperimen Penerapan Sistem Traffic Counting dengan Algoritma YOLO (You Only Look Once) V.4.,” *J. Media Inform. Budidarma*, vol. 5, no. 4, p. 1438, 2021, doi: 10.30865/mib.v5i4.3309.
- [3] Z. Wu et al., “MD-NDNet: a multi-dimensional convolutional neural network for false-positive reduction in pulmonary nodule detection,” *Phys. Med. Biol.*, vol. 65, no. 23, p. 235053, Dec. 2020, doi: 10.1088/1361-6560/aba87c.
- [4] T. M. Tamtelahitu, “Komparasi Algoritma Clustering dengan Dataset Penyebaran Covid-19 di Indonesia Periode Maret-Mei 2020,” *J. Teknol. Technoscientia*, vol. 13, no. 1, pp. 27–34, 2020.
- [5] Fachruddin, Y. Pratama, E. Rasywir, D. Kisbianty, Hendrawan, and M. R. Borroek, “Real Time Detection on Face Side Image with Ear Biometric Imaging Using Integral Image and Haar-Like Feature,” in *Proceedings of 2018 International*



- Conference on Electrical Engineering and Computer Science, ICECOS 2018, 2019, pp. 165–170, doi: 10.1109/ICECOS.2018.8605218.
- [6] G. Aprilian Anarki, K. Auliasari, and M. Orisa, “PENERAPAN METODE HAAR CASCADE PADA APLIKASI DETEKSI MASKER,” *JATI (Jurnal Mhs. Tek. Inform., vol. 5, no. 1, pp. 179–186, 2021, doi: 10.36040/jati.v5i1.3214.*
- [7] S. Kornblith, J. Shlens, and Q. V. Le, “Do better imagenet models transfer better?,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 2656–2666, 2019, doi: 10.1109/CVPR.2019.00277.*
- [8] M. Ju, H. Luo, Z. Wang, B. Hui, and Z. Chang, “The application of improved YOLO V3 in multi-scale target detection,” *Appl. Sci., vol. 9, no. 18, 2019, doi: 10.3390/app9183775.*
- [9] H. Bunyamin, Heriyanto, S. Novianti, and L. Sulistiani, “Topic clustering and classification on final project reports: A comparison of traditional and modern approaches,” *IAENG Int. J. Comput. Sci., vol. 46, no. 3, pp. 1–6, 2019.*
- [10] G. Liu, J. C. Nouaze, P. L. T. Mbouembe, and J. H. Kim, “YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3,” *Sensors (Switzerland), vol. 20, no. 7, pp. 1–20, 2020, doi: 10.3390/s20072145.*
- [11] S. K. Yoki Donzia and H. K. Kim, “Implementation of recurrent neural network with sequence to sequence model to translate language based on tensorflow,” *Lect. Notes Eng. Comput. Sci., vol. 2237, pp. 375–379, 2018.*
- [12] P. Wanda and H. J. Jie, “DeepSentiment: Finding malicious sentiment in online social network based on dynamic deep learning,” *IAENG Int. J. Comput. Sci., vol. 46, no. 4, pp. 1–12, 2019.*
- [13] Z. Wu et al., “MD-NDNet: a multi-dimensional convolutional neural network for false-positive reduction in pulmonary nodule detection,” *Phys. Med. Biol., vol. 65, no. 23, p. 235053, 2020, doi: 10.1088/1361-6560/aba87c.*
- [14] K. Celikmih, O. Inan, and H. Uguz, “Failure Prediction of Aircraft Equipment Using Machine Learning with a Hybrid Data Preparation Method,” *Sci. Program., vol. 2020, pp. 1–10, 2020, doi: 10.1155/2020/8616039.*
- [15] A. P. Rodrigues et al., “Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques,” *Comput. Intell. Neurosci., vol. 2022, 2022, doi: 10.1155/2022/5211949.*
- [16] X. Li, L. Bing, W. Lam, and B. Shi, “Transformation Networks for Target-Oriented Sentiment Classification,” *arxiv, 2018.*
- [17] D. Zhao, Y. Liu, H. Yin, and Z. Wang, “A novel multi-scale CNNs for false positive reduction in pulmonary nodule detection,” *Expert Syst. Appl., vol. 207, p. 117652, 2022, doi: https://doi.org/10.1016/j.eswa.2022.117652.*
- [18] D. Z. Abidin, S. Nurmaini, Erwin, E. Rasywir, and Y. Pratama, “Indoor Positioning System in Learning Approach Experiments,” *J. Electr. Comput. Eng., vol. 2021, 2021, doi: 10.1155/2021/6592562.*
- [19] S. Nurmaini, A. Darmawahyuni, A. N. S. Mukti, M. N. Rachmatullah, F. Firdaus, and B. Tutuko, “Deep learning-based stacked denoising and autoencoder for ECG heartbeat classification,” *Electron., vol. 9, no. 1, 2020, doi: 10.3390/electronics9010135.*
- [20] A. B. Adege, H. P. Lin, G. B. Tarekegn, Y. Y. Munaye, and L. Yen, “An indoor and outdoor positioning using a hybrid of support vector machine and deep neural network algorithms,” *J. Sensors, vol. 2018, 2018, doi: 10.1155/2018/1253752.*
- [21] Y. Wang, J. Gao, Z. Li, and L. Zhao, “Robust and accurate Wi-Fi fingerprint location recognition method based on deep neural network,” *Appl. Sci., vol. 10, no. 1, 2020, doi: 10.3390/app10010321.*
- [22] J. Yu, B. Yang, J. Wang, J. K. Leader, D. O. Wilson, and J. Pu, “2D CNN versus 3D CNN for false-positive reduction in lung cancer screening,” *J. Med. Imaging, vol. 7, no. 5, p. 51202, 2020, doi: 10.1117/1.JMI.7.5.051202.*
- [23] F. Ullah, M. R. Naeem, L. Mostarda, and S. A. Shah, “Clone detection in 5G-enabled social IoT system using graph semantics and deep learning model,” *Int. J. Mach. Learn. Cybern., 2021, doi: 10.1007/s13042-020-01246-9.*
- [24] Y. Wang et al., “EV-gait: Event-based robust gait recognition using dynamic vision sensors,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 6351–6360, 2019, doi: 10.1109/CVPR.2019.00652.*
- [25] D. Spatharakis et al., “A scalable Edge Computing architecture enabling smart offloading for Location Based Services,” *Pervasive Mob. Comput., vol. 67, p. 101217, 2020, doi: 10.1016/j.pmcj.2020.101217.*
- [26] Hendry and R. C. Chen, “Using deep learning to predict user rating on imbalance classification data,” *IAENG Int. J. Comput. Sci., vol. 46, no. 1, 2019.*
- [27] Q. Zheng, X. Tian, M. Yang, and H. Wang, “Differential learning: A powerful tool for interactive content-based image retrieval,” *Eng. Lett., vol. 27, no. 1, pp. 202–215, 2019.*
- [28] N. Akhtar and U. Ragavendran, “Interpretation of intelligence in CNN-pooling processes: a methodological survey,” *Neural Comput. Appl., vol. 32, no. 3, pp. 879–898, 2020, doi: 10.1007/s00521-019-04296-5.*