

# Implementasi *Deep Learning* Menggunakan *Convolutional Neural Network* untuk Sistem Pengenalan Wajah

Noviana Dewi<sup>1</sup>, Fiqih Ismawan<sup>2</sup>

<sup>1</sup>Department of Informatic, Universitas Budi Luhur, Indonesia

<sup>2</sup>Department of Informatic, Universitas Indraprasta PGRI, Indonesia

---

## Article Info

### Article history:

Received: Feb 15, 2021

Revised: Feb 22, 2021

Accepted: Mach 12, 2021

---

### Keywords:

Sistem Pengenalan Wajah  
CNN

*Deep Learning*  
*Computer Vision*  
*OpenFace*

---

## ABSTRACT

Face recognition system is generally divided into two stages, face detection system, which is a pre-processing step followed by a facial recognition system. This step will quickly be done by humans but it takes a long time for the computer. This ability of humans is what researchers want to duplicate in the last few years as biometric technology in computer vision to create a model of face recognition in computer. Deep learning becomes a spotlight in developing machine learning, the reason because deep learning has reached an extraordinary result in computer vision. Based on that, the author came up with an idea to create a face recognition system by implementing deep learning using the CNN method and applying library openFace. The CNN methods are still superior and widely used because they have good accuracy. The initial process was taking a picture of the face to be used as a dataset. From this dataset, face preprocessing will be carried out, that is, to extract the facial vector features into 128-d and to classify the facial vector. The contribution of this research is the addition of features to improve the accuracy of the facial recognition system using the CNN method. The results of this research get a precision value of 98.4%, a recall of 98% and an accuracy of 99.84%.

Copyright © 2021 Universitas Indraprasta PGRI.  
All rights reserved.

---

## Corresponding Author:

Noviana Dewi,  
Department of Informatic,  
Universitas Budi Luhur,  
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan.  
Email: noviand313@gmail.com

---

## 1. PENDAHULUAN

Sistem pengenalan wajah menjadi topik yang sering dipelajari di bidang *computer vision* dalam beberapa dekade ini. Sistem ini telah diaplikasikan dalam beberapa bidang, misalnya, *smartphone* untuk *facelock*, imigrasi dan juga di media sosial untuk fitur *facial tagging*[1]. Sistem pengenalan wajah secara umum dibagi menjadi dua tahap, yaitu sistem deteksi wajah yang merupakan tahap awal (*pre-processing*) dan dilanjutkan dengan sistem pengenalan wajah (*face recognition*)[2]. Tahap tersebut begitu cepat dilakukan oleh manusia tetapi butuh waktu yang lama bagi komputer. Kemampuan manusia itulah yang ingin diduplikasi oleh para peneliti dalam beberapa tahun belakangan ini sebagai teknologi biometrik dalam bidang *computer vision*. Tujuan penduplikasian itu adalah membentuk suatu model pengenalan citra wajah pada komputer.

Belakangan ini *Deep Learning* menjadi sorotan dalam pengembangan *Machine Learning*. Alasannya karena *Deep Learning* telah mencapai hasil yang luar biasa dalam visi komputer. *Deep Learning* merupakan suatu metode pembelajaran yang dilakukan mesin untuk dapat memahami dan mengklasifikasi suatu objek, yakni utamanya dalam aplikasi yang dibangun ini adalah wajah yang ditangkap dalam bentuk citra[3]. *Convolutional Neural Network* (CNN) merupakan salah satu metode dalam *Deep Learning* yang dibuat untuk menutupi kelemahan dari metode sebelumnya. Terdapat beberapa kelemahan dalam metode sebelumnya, tetapi dengan model ini sejumlah parameter bebas dapat dikurangi dan deformasi gambar input seperti translasi, rotasi dan skala dapat ditangani.

Berbagai penelitian sudah mulai dilakukan untuk menguji beberapa metode *face recognition* untuk menemukan sebuah metode yang baik dengan akurasi pengenalan yang tinggi. Salah satunya yang dilakukan oleh Arsal pada tahun 2020, dalam penelitian ini hasil percobaan menunjukkan tingkat akurasi sebesar 95% menggunakan *deep learning* dengan metode *Convolutional Neural Network* [4]. Penelitian yang dilakukan oleh Achmad pada tahun 2019, dalam penelitian ini mengusulkan klasifikasi emosi berdasarkan ciri wajah untuk membantu meningkatkan kualitas pengujian konsumen. Metode yang digunakan pada penelitian ini adalah *Convolutional Neural Network* (CNN). Hasil pengujian menunjukkan nilai *learning rate* tertentu dapat melatih arsitektur lebih baik dibandingkan dengan nilai *learning rate* lain. Nilai akurasi terbaik pada penelitian ini sebesar 86,4% dan rata-rata akurasi sebesar 80,7%[5]. Penelitian yang dilakukan oleh Meenakshi pada tahun 2019, Dalam penelitian ini mengusulkan aplikasi sistem pengenalan wajah menggunakan metode *Convolutional Neural Network* (CNN). Sistem pengenalan wajah yang diusulkan diimplementasikan dan dianalisis dalam platform MATLAB 2018. Hasil akurasi dari penelitian ini yaitu 98,75% [6].

Seiring dengan banyaknya pengembangan dan riset tentang *Deep Learning*, banyak *library* yang bermunculan dengan fokus mempelajari tentang jaringan syaraf tiruan . salah satu contohnya yaitu OpenFace. OpenFace merupakan *library* yang digunakan untuk keperluan *face recognition* dan didasarkan pada desain *neural network* dari DeepFace dan GoogleNet dengan beberapa modifikasi.

Dari penelitian sebelumnya, maka penulis merancang sistem pengenalan wajah dengan menggunakan metode *Convolutional Neural Network* dengan mengaplikasikan *library* OpenFace untuk memperdalam pengetahuan tentang *deep learning* dan mengetahui tingkat akurasi yang dihasilkan.

### 1.1. Pengenalan Wajah

Pengenalan wajah adalah salah satu teknologi *biometric* yang telah banyak diaplikasikan dalam sistem keamanan selain pengenalan retina mata, pengenalan sidik jari dan iris mata. Dalam aplikasinya sendiri pengenalan wajah menggunakan sebuah kamera untuk menangkap wajah seseorang kemudian dibandingkan dengan wajah yang sebelumnya telah disimpan didalam *database* tertentu[7].

Pengenalan wajah melibatkan banyak variabel, misalnya citra sumber, cira hasil pengolahan citra, citra hasil ekstraksi dan data profil seseorang. Dibutuhkan juga alat pengindra berupa sensor kamera dan metode untuk menentukan apakah citra yang ditangkap oleh *webcam* tergolong wajah manusia atau bukan, sekaligus untuk menentukan informasi profil yang sesuai dengan citra wajah yang dimaksud.

### 1.2. Convolutional Neural Network

*Convolutional Neural Network* (CNN) adalah salah satu metode klasifikasi yang termasuk ke dalam kelompok *deep learning* yang menggunakan *layer* konvolusi untuk mengonvolusi suatu input dengan *filter*. CNN terdiri dari dua tahapan utama yaitu *feature learning* dan *classification*. Pada tahapan *feature learning* terdiri dari *convolution layer*, ReLU (fungsi aktivasi) dan *pooling layer* sedangkan pada tahap *classification* terdiri dari *flatten*, *fully-connected layer*, dan prediksi. Pada setiap bagian CNN terdapat dua proses utama, yaitu *feed-forward* dan *backpropagation*[5].

### 1.3. Caffe Deep Learning

CAFFE (*Convolutional Architecture for Fast Feature Embedding*) merupakan salah satu dari sekian banyak *deep learning framework* yang dapat digunakan untuk merancang dan menjalankan klasifier *neural network*.

Caffe menggunakan format prototxt Google untuk spesifikasi jaringan *neural*. Setiap lapisan didefinisikan secara terpisah dan *input* dan *output* adalah array multidimensi yang disebut blob. Lapisan ditumpuk secara vertikal dengan masukan di bagian bawah dan keluaran di atas. Setiap lapisan mengharapkan sejumlah bagian bawah dan atas gumpalan dengan dimensi tertentu. Merangkai blob dan lapisan ini menghasilkan jaringan neural yang dalam [8].

### 1.4. OpenFace

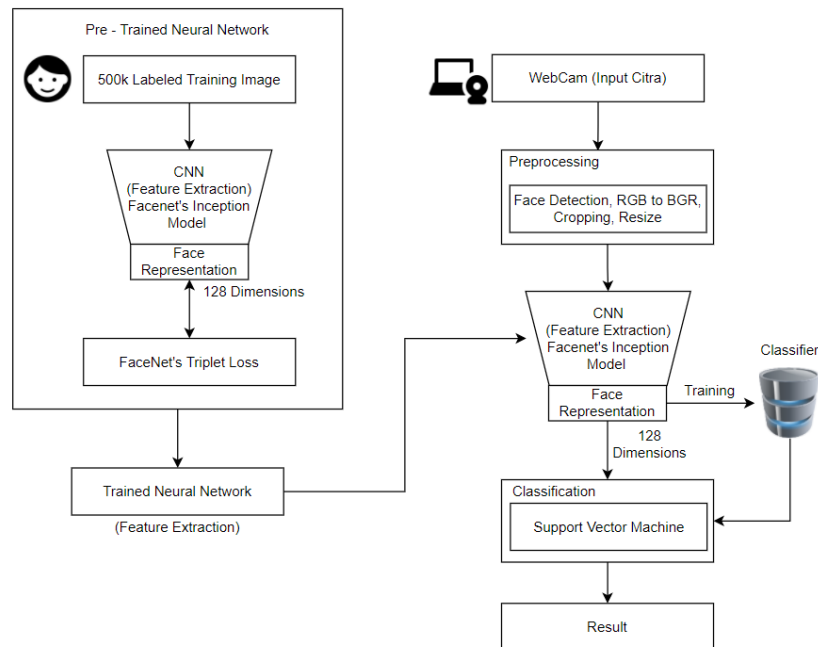
OpenFace merupakan *library* yang digunakan untuk keperluan *face recognition* dan didasarkan pada desain *neural network* dari DeepFace dan GoogleNet dengan beberapa modifikasi. OpenFace menggunakan dlib 22 untuk mendeteksi wilayah wajah dalam gambar dan menghasilkan kotak yang mengelilingi setiap wajah yang dapat berada dibawah pose yang berbeda. OpenFace menggunakan transformasi affine 2D sebagai metode *preprocessingnya* yaitu mengatur sudut hidung dan mata relatif dekat ke lokasi rata-rata dengan mengubah ukuran dan memotong gambar ke tepi bangunan terkenal yang dihasilkan oleh detektor wajah dlib. Hasil transformasi ini adalah citra yang dinormalisasi berukuran 96 x 96 piksel.

## 2. METODE

Pada penelitian ini, metode CNN yang digunakan untuk pengenalan wajah terdiri dari beberapa tahapan. Tahapan tersebut dimulai dengan pengumpulan data pelatihan yang digunakan untuk pembelajaran *neural network*. Metode *Convolutional Neural Network* (CNN) yang digunakan dalam penelitian ini memiliki *pre-trained neural network* sebagai metode ekstraksi fitur. Data citra wajah pelatihan digunakan untuk melatih *neural network*. Pelatihan

terdiri dari beberapa proses. Proses pertama adalah pendeteksian wajah yang terdapat dalam citra gambar. Ketika terdapat citra wajah maka dilakukan proses ekstraksi ciri yang akan menghasilkan nilai-nilai ciri setiap wajah. Nilai ciri ini disimpan dan digunakan sebagai pengenalan (*classifier*). Ketika dilakukan pengujian, maka *classifier* akan digunakan untuk klasifikasi wajah. Setiap proses yang dilakukan akan dibahas lebih rinci pada bagian selanjutnya.

Design metode penelitian ditunjukkan pada gambar 1, dimana alur kerja dari sistem ini terdiri dari beberapa tahapan.



Gambar 1. Proses sistem pengenalan wajah

### 2.1. Pre – Trained Neural Network

Dalam penelitian ini, digunakan sebuah model *pre – trained neural network* yaitu sebuah *neural network* yang telah dilatih menggunakan banyak *dataset* wajah. *Pre-trained neural network* yang digunakan menggunakan model *openface nn4.small.2* yang telah melakukan pelatihan terhadap 500 ribu *dataset* citra pelatihan. Proses *pre-trained neural network* bekerja dengan beberapa data pelatihan, yaitu citra wajah pelatihan orang yang dikenal, citra wajah pelatihan dengan orang yang sama dan citra wajah orang yang berbeda. Ketiga data citra tersebut kemudian diproses untuk menghasilkan nilai ciri untuk setiap citra menggunakan algoritma *Convolutional Neural Network (CNN)*. *Neural Network* ini kemudian dilatih kembali untuk memastikan bahwa 2 citra dengan orang yang sama memiliki nilai ciri yang identik sedangkan memiliki nilai yang berbeda untuk orang yang berbeda. Proses ini diulang untuk data pelatihan lainnya sampai didapatkan kondisi *neural network* yang ideal.

Hasil pelatihan *Neural Network* tersebut yang digunakan sebagai *Pre-Trained Neural Network* untuk proses pengenalan wajah selanjutnya.

### 2.2. Image Acquisition / Input Citra

Proses yang akan dilakukan pertama adalah tahap akuisisi citra berfungsi untuk pengambilan citra wajah menggunakan webcam.

### 2.3. PreProcessing

Berikut ini merupakan langkah – langkah dari tahapan *preprocessing*.



Gambar 2. Langkah – langkah *PreProcessing*

Pada gambar 2, dijelaskan bahwa Setelah pengambilan citra wajah (*capture image*), data wajah akan dilakukan beberapa proses yaitu proses konversi citra RGB - BGR menggunakan *function blobFromImage* dari *OpenCV*. Selanjutnya dilanjutkan dengan proses *cropping* yaitu memotong gambar yang tidak perlu (yang

diambil hanya lah citra wajah saja). Kemudian dilanjutkan oleh proses *resize* yaitu untuk mengatur ukuran citra wajah sesuai dengan resolusi yang telah ditetapkan yaitu sebesar 299 x 299 pixel.

Setelah dilakukan proses *resize*, kemudian dilakukan tahap deteksi wajah menggunakan model Caffe Deep Learning. Berikut ini merupakan langkah – langkah *Pre-trained Caffe Deep Learning* :

a. *Data Preparation*

pada langkah ini dilakukan *preprocessing* kemudian menyimpannya dalam format yang dapat dibaca oleh Caffe. Penulis menggunakan *script* python untuk menangani *pre – processing* dan penyimpanan gambar.

b. *Model Definition*

Pada langkah ini, penulis memilih arsitektur CNN dan penulis mendefinisikan parameternya dalam *file* konfigurasi dengan ekstensi *.prototxt*.

c. *Model Training*

Kami melatih model dengan menjalankan satu perintah Caffe dari terminal. Setelah melatih model, kita akan mendapatkan model yang dilatih dalam *file* dengan ekstensi *.caffemodel*.

## 2.4. Feature Extraction

Ekstraksi fitur merupakan proses yang berfungsi untuk mendapatkan informasi yang efektif dan berguna untuk membedakan wajah – wajah orang yang berbeda dari citra wajah yang telah diselaraskan [9].

Setelah dilakukan *preprocessing* wajah dengan melakukan deteksi, pemotongan (*cropping*) dan mengubah ukuran (*resize*) wajah. Metode untuk ekstraksi ciri menggunakan *Convolutional Neural Network* (CNN). Metode CNN yang telah menggunakan *Pre-Trained Neural Network* untuk merepresentasikan wajah manusia dengan menghasilkan 128 nilai ciri wajah [10].

Proses ekstraksi ciri wajah dengan metode *Convolutional Neural Network* memiliki beberapa tahapan *layer deep neural network*. Tahapan tersebut dimulai dari model *input* citra yang telah dinormalisasi 96x96x3. *Layer* yang digunakan dalam ekstraksi ciri hanya sampai pada *layer inception* dengan *pooling* akhir yang menghasilkan 128 nilai ciri. Arsitektur *Convolutional neural network* selengkapnya dapat dilihat pada tabel 1.

Tabel 1. Convolutional Neural Network nn4.small2 Openface

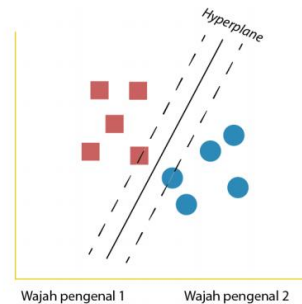
Type	Output Size	#1x1	#3x3 reduce	#3x3	#5x5 reduce	#5x5	Pool proj
conv1 (7 x 7 x 3,2)	48 x 48 x 64						
max pool + norm	24 x 24 x 64						m 3 x 3,2
inception (2)	24 x 24 x 192		64	64			
norm + max pool	12 x 12 x 192						m 3 x 3,2
inception (3a)	12 x 12 x 256	64	96	96	16	32	m, 32p
inception (3b)	12 x 12 x 320	64	96	96	32	64	ℓ <sub>2</sub> , 64p
inception (3c)	6 x 6 x 640		128	128	32	64,2	m 3 x 3,2
inception (4a)	6 x 6 x 640	256	96	96	32	64	ℓ <sub>2</sub> , 128p
inception (4e)	3 x 3 x 1024		160	160	64	128,2	m 3 x 3,2
inception (5a)	3 x 3 x 736	256	96	96			ℓ <sub>2</sub> , 96p
inception (5b)	3 x 3 x 736	256	96	96			m, 96p
avg pool	736						
Linear	128						
ℓ <sub>2</sub> normalization	128						

Berdasarkan tabel 1 digambarkan sebuah arsitektur ekstraksi citra menggunakan model nn4.small2 Openface *Convolutional Neural Network*. Setiap baris pada tabel tersebut menunjukkan *layer Convolutional Neural Network* dan setiap kolom menunjukkan parameter *filtering pooling* dan *inception layer*. Pada proses *convolution* conv1, *input* citra diproses menghasilkan *output* dengan ukuran 48x48x64. *Output* dari conv1 selanjutnya dilakukan *pooling* dan normalisasi menggunakan *kernel pool* proj m 3 x 3,2. Proses *pooling* menghasilkan *output* 24x24x192. Hasil *output* ini digunakan dalam proses *inception* dengan menggunakan 64 *kernel* #3x3 *reduce* dan 64 *kernel* #3x3. *Output* dari proses *inception* dinormalisasi kembali dengan *max-pool* proj m 3x3,2. *Output* ini digunakan dalam beberapa *layer inception* dengan beberapa *filter* sehingga menghasilkan *output* akhir 736 nilai dan kembali dilakukan *avg pooling* dan dinormalisasi menghasilkan 128 nilai ekstraksi. *Convolutional Neural Network* dengan 128 nilai pengukuran dihasilkan untuk setiap wajah dan akan disimpan dalam *database*. Data nilai untuk sebuah pengenalan dilakukan proses *training* dengan model *Support Vector Machine*.

## 2.5. Classification

Klasifikasi citra wajah dilakukan untuk mengetahui wajah siapa yang terdapat dalam citra. Untuk membandingkan nilai-nilai wajah digunakan algoritma klasifikasi *Support Vector Machine* (SVM). *Support Vector Machine* (SVM) merupakan metode klasifikasi yang diperkenalkan pertama kali oleh Vapnik pada tahun 1998[11].

Proses klasifikasi dilakukan dengan membandingkan nilai citra yang ada dalam *database(classifier)* dengan data citra inputan yang diuji. Nilai ciri yang mirip akan diklasifikasikan ke dalam sebuah kelas yang sama. Citra yang telah dilatih memiliki pembatas yang disebut *hyperplane* untuk setiap pengenalnya. *Hyperplane* ini yang memisahkan nilai ciri wajah satu dengan lainnya. *Hyperplane* diperoleh dengan menempatkan sebuah pembatas dari jarak nilai terjauh dari 2 kelas berbeda. Gambar 4 menunjukkan *hyperplane* dari 2 kelas wajah yang berbeda.



Gambar 4. Hyperplane SVM

Metode SVM yang digunakan adalah *Linear kernel SVM*. *Linear kernel* merupakan fungsi kernel yang paling sederhana. *Linear kernel* digunakan ketika data yang dianalisis sudah terpisah secara linear. *Linear kernel* cocok ketika terdapat banyak fitur dikarenakan pemetaan ke ruang dimensi yang lebih tinggi tidak benar – benar meningkatkan kinerja seperti pada klasifikasi teks. Fungsi ini baik digunakan ketika data sudah terpisah secara *linear*, kernel ini dilakukan dengan persamaan 1 [12] dimana  $K(x_i, x_j)$  merupakan fungsi kernel sedangkan  $x_i$  dan  $x_j$  adalah pasangan dua data *training*.

$$K(x_i, x_j) = x_i \cdot x_j \quad (1)$$

Dalam melakukan analisis dengan fungsi *linear kernel SVM* ini, untuk mendapatkan akurasi yang optimal maka akan mencari nilai parameter  $C$  (*Cost*). Parameter  $C$  yang digunakan untuk pertama adalah dengan nilai  $C = 0.1$ , kemudian nilai parameter ini diperbesar untuk mencari akurasi yang paling optimal pada *Linear kernel SVM*.

## 2.6. Recognition Result

Setelah melalui proses *pre – processing*, ekstraksi ciri kemudian dilakukan proses identifikasi wajah, ketika suatu citra wajah pengguna cocok dengan data citra dalam *database* maka akan menampilkan identitas dari pengguna yang memiliki citra wajah tersebut.

## 3. HASIL DAN PEMBAHASAN

### 3.1. Pengujian Sistem

Pada bagian ini dijelaskan mengenai pengujian sistem menggunakan *Convolutional Neural Network (CNN)* menggunakan dataset uji. Pada tahap pertama, citra data pelatihan harus melalui tahap pendeteksian wajah, konversi citra, dan *cropping*. Hasil pendeteksian wajah dan *cropping* ditunjukkan pada gambar 5.



Gambar 5. Hasil Cropping Image

Setelah dilakukan *preprocessing* kemudian dilakukan ekstraksi ciri. Pada penelitian ini terdapat 4 ekstraksi fitur yang digunakan yaitu :

#### a. ROI

Proses ROI dalam penelitian ini digunakan untuk membatasi atau memperkecil area pemrosesan. Pembatasan area pemrosesan dilakukan dengan cara menentukan area wajah. Pembatasan area wajah dilakukan agar object – object yang berada diluar area tersebut tidak menjadi penambahan noise pada citra yang di proses. Tujuan daripada pengimplementasian ROI dalam penelitian ini adalah untuk meningkatkan waktu pemrosesan.

#### b. Bounding Box

*Bounding box* merupakan kotak imajiner yang mengelilingi objek yang teridentifikasi. *Bounding box* sendiri berbentuk kotak yang dimana besarnya sama seperti besar objek yang teridentifikasi tersebut. Pada proses *bounding box* mendapat data input dari hasil proses deteksi wajah, kemudian input tersebut diproses untuk mencari nilai X,Y,W,H. Kemudian ketika nilai X, Y, W, dan H telah ditemukan maka langkah selanjutnya membuat *bounding rectangle* dengan nilai-nilai tersebut.

Ciri tersebut digunakan untuk mengurangi tingginya waktu pemrosesan. Karena mampu menandai area tertentu sehingga dapat digunakan untuk mengoptimalkan kinerja sistem untuk mendeteksi, menghitung dan mengklasifikasi wajah secara *realtime*. Jika tidak, pemrosesan dilakukan pada seluruh piksel citra tanpa terkecuali. Berikut ini merupakan potongan program proses ekstraksi, yang kemudian dilanjutkan oleh proses penyematanan.

---

#### Potongan Program

```
# compute the (x, y)-coordinates of the bounding box for the face
box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
(startX, startY, endX, endY) = box.astype("int")

# extract the face ROI
face = image[startY:endY, startX:endX]
(fH, fW) = face.shape[:2]
```

---

Setelah penambahan ekstraksi fitur kemudian dilanjutkan dengan mengekstraksi nilai ciri setiap wajah dengan metode Convolutional Neural Network (CNN). Proses ini dilakukan untuk mengubah *dataset image* menjadi *vector* yang mendeskripsikan wajah yang disimpan dalam *file* dengan format *.pickle* (*embedding.pickle*). Proses mengubah *dataset image* menjadi *vector* 128 d dilakukan menggunakan *source code* seperti berikut.

---

#### Potongan Program

```
# Image to Vector 128 d
faceBlob = cv2.dnn.blobFromImage(face, 1.0 / 255, (96, 96), (0, 0, 0),
    swapRB=True, crop=False)
embedder.setInput(faceBlob)
vec = embedder.forward()
# Save The Facial Embeddings + Names To Disk
data = {"embeddings": knownEmbeddings, "names": knownNames}
f = open(args["embeddings"], "wb")
f.write(pickle.dumps(data))
```

---

Contoh ekstraksi 128 nilai ciri yang akan disimpan dalam *database* dapat dilihat pada gambar 4.2.



```
{'embeddings': [array([ 5.3776016e-02,  9.68738273e-03, -1.24139011e-01,  6.78037666e-03,
 5.58484837e-05,  1.66598258e-01,  5.95912184e-03,  3.39928893e-02,
-1.05585650e-01,  2.90589817e-02,  9.30784643e-02,  5.03563061e-02,
9.40679466e-02, -1.02654786e-01, -6.98419899e-02, -1.33277848e-02,
-1.11302838e-01,  1.55454483e-02,  1.69515330e-02,  1.25268161e-01,
1.57141283e-01, -2.08868368e-01,  1.30298896e-01,  6.74129725e-02,
4.38601747e-02,  3.04707131e-03,  1.18336594e-02, -2.35443041e-01,
2.11467054e-02,  4.30857038e-02, -7.72573042e-03, -6.08798033e-02,
-7.71721476e-04,  3.51456851e-02, -4.09566634e-02, -3.15224798e-03,
-1.19136289e-01,  1.17634185e-01, -1.97729878e-02, -8.45508054e-02,
-2.51323227e-02, -1.44393086e-01, -7.25957900e-02,  1.36477128e-01,
-1.68024033e-01, -8.44414011e-02,  1.35842666e-01, -2.44025402e-02,
-1.50551908e-01,  6.04188079e-02,  6.58188899e-03,  8.48342106e-03,
-4.16952372e-02,  2.12679692e-02,  7.29367808e-02,  2.41809171e-02,
4.15381055e-02,  1.10842356e-01, -1.39529221e-02, -1.31671026e-01,
6.52964925e-02,  5.76764420e-02,  2.64947742e-01, -9.17603374e-02,
1.02765240e-01, -2.52558840e-02, -6.67395592e-02,  1.67420954e-02,
-5.21468930e-02,  8.31360817e-02, -5.24013825e-02,  9.14673507e-02,
-1.25410244e-01, -3.04320119e-02,  4.16282527e-02, -3.55668999e-02,
-4.19986062e-03,  4.80670519e-02,  2.53615361e-02,  1.55926824e-01,
-1.84784219e-01, -1.12222277e-01,  2.31985636e-02, -3.27436067e-02,
-1.12596706e-01,  3.33692841e-02,  6.23553217e-02,  3.50700393e-02,
9.91241783e-02, -4.42125201e-02,  5.03931455e-02, -1.08140312e-01,
-5.3175947e-02, -7.05000199e-03, -4.34254371e-02, -1.40682325e-01,
-1.11367013e-02,  1.46667823e-01,  3.40531905e-02, -6.56842962e-02,
3.34408581e-02,  3.83625031e-02, -1.67287196e-02,  3.93162332e-02,
-7.00245611e-03,  8.94885957e-02, -1.66184008e-02,  1.41697541e-01,
4.65022996e-02, -1.46559745e-01, -4.54768840e-02,  1.19351014e-01,
-2.03560210e-02, -5.98199409e-02,  1.31956110e-01,  9.45748322e-02,
-2.67028473e-02, -5.89134432e-02,  1.10854061e-01,  2.02899562e-01,
-1.32537410e-02, -1.02225699e-01, -2.66834162e-02,  2.38756258e-02,
-3.45199890e-02,  1.10697187e-01,  1.04795858e-01,  1.03170067e-01],
dtypes=float32)}, 'names': ['Andri Wahyu Saputra']}]
```

Gambar 4.2 : vector 128-d

Proses selanjutnya yaitu proses klasifikasi. Pada proses klasifikasi pengujian menggunakan hasil proses *embedding* yang berupa nilai *vector* dari sebuah citra. Kemudian *vector* tersebut diteruskan melalui model pengenalan SVM yang hasilnya adalah prediksi dari citra tersebut. Selanjutnya yaitu mengambil nilai probabilitas tertinggi dari hasil prediksi yang diperoleh. Kemudian melanjutkan ke proses *encoder label* untuk mendapatkan nama dari citra tersebut.

---

#### Potongan Program

```
vec = embedder.forward()
# perform classification to recognize the face
```

---

```

preds = recognizer.predict_proba(vec)[0]
j = np.argmax(preds)
proba = preds[j]
name = le.classes_[j]
# draw the bounding box of the face along with the associated probability
text = "{}: {:.2f}%".format(name, proba * 100)

```

### 3.2. Pengujian Deteksi Wajah

Pengujian ini dilakukan untuk melihat seberapa besar keakuratan pendeteksian citra wajah dengan menggunakan algoritma *Caffe Deep Learning* yang akan digunakan untuk proses pengenalan didalam melakukan perekaman wajah pengguna aplikasi. Tabel 2 menunjukkan hasil pengujian dengan beberapa kriteria.

Tabel 2. Pengujian Deteksi Wajah

Citra Uji	Jenis Pengujian	Hasil	Keterangan
	Citra wajah tampak depan	Terdeteksi	Ketika wajah dengan pose menghadap depan, sistem masih mendeteksi keberadaan wajah.
	Citra wajah dengan posisi kepala menunduk	Terdeteksi	Ketika wajah dengan pose menghadap depan dan kepala menunduk, sistem masih mendeteksi keberadaan wajah.
	Pencahayaan Normal	Terdeteksi	Ketika pencahayaan dari citra tersebut normal, sistem dapat mendeteksi keberadaan wajah.
	Pencahayaan Redup	Terdeteksi	Ketika pencahayaan dari citra tersebut redup, sistem masih dapat mendeteksi keberadaan wajah.
	Pencahayaan Gelap	Terdeteksi	Ketika pencahayaan dari citra tersebut gelap, sistem masih dapat mendeteksi keberadaan wajah.

### 3.3. Pengujian Pengenalan Wajah

Pada penelitian ini, terdapat 250 citra dengan komposisi 150 citra wajah sebagai data latih dan 100 citra sebagai data uji seperti yang tertera pada Tabel 3.

Tabel 3. Dataset

Jumlah Sample Citra	Citra Per Orang	Total	
25	10	250	
Citra Latih	Citra Uji	Total Citra Latih	Total Citra Uji
6	4	150	100

Pengujian ini untuk mengukur tingkat *precision*, *recall* dan *accuracy* menggunakan *Confusion Matrix*. Secara umum *precision*, *recall* dan *accuracy* dapat dirumuskan berdasarkan gambar berikut [13].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 6. Confussion Matrix

*Precision*, digunakan untuk menghitung ketepatan antara informasi yang diminta dengan jawaban yang diberikan oleh sistem. *Recall*, digunakan untuk mengukur tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *Accuracy*, digunakan untuk membandingkan tingkat kedekatan antara nilai prediksi dengan nilai *actual*. Berdasarkan gambar 6, *precision*, *recall* dan *accuracy* memiliki cara perhitungan masing – masing. Perhitungan nilai dapat dihitung dengan persamaan berikut :

$$\text{Precision} = \frac{TP}{(TP+FP)} \times 100\%$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \times 100\%$$

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\%$$

Seperti halnya pada proses pengujian deteksi wajah, proses pengujian identifikasi wajah pun dilakukan dengan berbagai kriteria seperti yang dicantumkan di Tabel 4.

Tabel 4. Pengujian Identifikasi Wajah

Citra Uji	Jenis Pengujian	Hasil	Keterangan
	Citra wajah tampak depan	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Citra wajah dengan posisi kepala menunduk	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Pencahayaan Normal	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Pencahayaan Redup	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.
	Pencahayaan Gelap	Dapat Dikenali	Ketika wajah dengan pose menghadap depan, sistem dapat mengidentifikasi wajah.

Dari hasil pengujian sistem pengenalan wajah menggunakan metode CNN yang telah dilakukan, didapatkan persentase akurasi sistem pengenalan wajah menggunakan data uji sebesar 98%. Dari hasil pengujian tersebut terdapat 2 citra yang tidak dikenali oleh sistem dikarenakan jarak antara *object* dengan kamera terlalu dekat. Persentase akurasi ini dihitung dengan menggunakan persamaan berikut :

$$\text{Akurasi} = \frac{\text{Jumlah data uji yang dikenali}}{\text{Jumlah data uji}} \times 100\%$$

$$\text{Akurasi} = \frac{98}{100} \times 100\% = 98\%$$

Kemudian hasil persentase sistem pengenalan wajah menggunakan pengujian *Confusion Matrix* menghasilkan nilai *precision* sebesar 98,4%, *recall* sebesar 98% dan *accuracy* sebesar 99,84%. Untuk melakukan prediksi nilai *precision*, *recall* dan *accuracy* dapat dihitung dari tabel 4. Dengan sample pengujian citra pertama.



Tabel 4. Sample Hasil Percobaan Menggunakan Confusion Matrix

N = 100		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	4	0
	FALSE	0	96
TOTAL		4	96

Dengan perhitungan sebagai berikut :

$$Precision = \frac{4}{4+0} \times 100\% = 100\%$$

$$Recall = \frac{4}{4+0} \times 100\% = 100\%$$

$$Accuracy = \frac{4+96}{4+96+0+0} \times 100\% = 100\%$$

### 3.4. Pembahasan

Algoritma CNN merupakan salah satu algoritma *deep learning* yang fokus untuk mempelajari ciri dari suatu *object* secara lebih mendalam. Salah satu yang menjadi bagian terpenting dari berhasilnya algoritma ini untuk melakukan klasifikasi terhadap suatu *object* adalah banyaknya jumlah data. Semakin banyak jumlah data yang dilatih pada proses *training*, maka hasil akurasi yang didapat akan lebih tinggi. Begitupun sebaliknya, algoritma CNN ini tidak optimal ketika data yang dilatih sedikit. Oleh karena itu, perlu adanya operasi ekstraksi tambahan untuk meningkatkan hasilnya.

Pada penelitian ini terdapat penambahan 2 proses ekstraksi fitur, yaitu ROI dan *Bounding Box*. Ciri tersebut digunakan untuk mengurangi tingginya waktu pemrosesan. Dalam melakukan proses pengenalan wajah, algoritma CNN berhasil mendapatkan tingkat akurasi 98%. Dalam sistem pengenalan wajah ini hasil yang didapatkan seharusnya masih bisa ditingkatkan lagi karena jumlah data yang digunakan dalam penelitian ini terlalu sedikit.

## 4. PENUTUP

### 4.1. Simpulan

Berdasarkan hasil penelitian yang telah melakukan perancangan dan perencanaan pembuatan program *Face Recognition* serta melalui pengujian yang telah dilakukan maka dapat diambil kesimpulan antara lain, teknologi *Face Recognition* dapat dilakukan menggunakan *Convolutional Neural Network* (CNN). Untuk meningkatkan kinerja CNN, penulis menggunakan beberapa teknik pengoptimalan untuk melatih CNN. *Pre Trained CNN* dengan beberapa data tambahan untuk meningkatkan kemampuan generalisasi jaringan, yang membutuhkan lebih sedikit waktu untuk mengekstrak fitur wajah dari kumpulan data target. Proses pembuatan aplikasi ini dengan tahapan pembuatan *Face Recognition* yaitu akuisisi gambar, *preprocessing*, ekstraksi, klasifikasi, dan identifikasi data gambar. Tahapan tersebut dibuat dengan bahasa pemrograman python. Penelitian ini berhasil menggunakan *Face Recognition* oleh 25 orang *dataset* wajah yang terdiri dari 10 data wajah pada masing-masing orang. *Dataset* tersebut dipisahkan menjadi 2 tahapan data yaitu data latih dan data uji. Hasil dengan mengimplementasi *deep learning* menggunakan *framework Caffe Deep Learning* untuk proses deteksi wajah menghasilkan persentase 100% dengan kriteria yang sudah ditentukan sebelumnya. Sedangkan metode *Convolutional Neural Network* yang digunakan untuk proses klasifikasi menghasilkan persentase 98%, *precision* 98,4%, *recall* 98% dan *accuracy* 99,84%.

### 4.2. Saran

Berikut ini merupakan saran yang dapat digunakan sebagai bahan pertimbangan penelitian lebih lanjut yaitu untuk memperoleh hasil yang lebih baik dapat dilakukan dengan penambahan ciri atau metode yang berbeda. Dengan berkembangnya teknologi berbasis *mobile* maka perlu kiranya dicoba optimasi proses pengenalan wajah ini pada perangkat *handphone*. Sedangkan dalam aspek *hardware* dapat dilakukan dengan peningkatan kualitas

kamera yang digunakan untuk mendapatkan citra wajah dan peningkatan kualitas komputer yang digunakan dalam proses pengambilan citra wajah.

#### DAFTAR PUSTAKA

- [1] A. Santoso and G. Ariyanto, "Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah," *Emit. J. Tek. Elektro*, vol. 18, no. 01, pp. 15–21, 2018, doi: 10.23917/emit.v18i01.6235.
- [2] D. L. Z. Astuti and Samsuryadi, "Kajian Pengenalan Ekspresi Wajah menggunakan Metode PCA dan CNN," *Annu. Res. Semin. Fak. Ilmu Komput.*, vol. 4, no. 1, pp. 293–297, 2018.
- [3] A. P. Putera and P. N. Primandari, "Rancang Bangun Aplikasi Absensi Online Berbasis Android Menggunakan Metode Deep Learning Pada PT . Pelabuhan Indonesia III ( Persero )," 2020.
- [4] M. Arsal, B. Agus Wardijono, and D. Anggraini, "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN," *J. Nas. Teknol. dan Sist. Inf.*, vol. 6, no. 1, pp. 55–63, 2020, doi: 10.25077/teknosi.v6i1.2020.55-63.
- [5] Y. Achmad, R. C. Wihandika, and C. Dewi, "Klasifikasi emosi berdasarkan ciri wajah wenggunakan convolutional neural network," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 11, pp. 10595–10604, 2019.
- [6] S. Meenakshi, M. Siva Jothi, and D. Murugan, "Face recognition using deep neural network across variations in pose and illumination," *Int. J. Recent Technol. Eng.*, vol. 8, no. 1 Special Issue4, pp. 289–292, 2019.
- [7] M. R. Muliawan, B. Irawan, and Y. Brianorman, "Implementasi Pengenalan Wajah Dengan Metode Eigenface Pada Sistem Absensi," *J. Coding, Sist. Komput. Untan*, vol. 03, no. 1, pp. 41–50, 2015.
- [8] A. Kishore, "Designing Deep Learning Neural Networks using Caffe," no. January 2015, pp. 1–17, 2015, doi: 10.6084/M9.FIGSHARE.1546481.
- [9] J. Ali, "Rancang Bangun Sistem Pengenalan Wajah Dengan Metode Principal Component Analysis," vol. 1, no. 2, pp. 48–60, 2016.
- [10] K. Santoso and G. P. Kusuma, "Face Recognition Using Modified OpenFace," *Procedia Comput. Sci.*, vol. 135, pp. 510–517, 2018, doi: 10.1016/j.procs.2018.08.203.
- [11] M. Athoillah, "Pengenalan Wajah Menggunakan SVM Multi Kernel dengan Pembelajaran yang Bertambah," vol. 2, no. 2, pp. 84–91, 2017, doi: 10.15575/join.v2i2.109.
- [12] K. Mujib, A. Hidayatno, and T. Prakoso, "Pengenalan Wajah Menggunakan Local Binary Pattern (Lbp) Dan Support Vector Machine (Svm)," *Transient*, vol. 7, no. 1, p. 123, 2018, doi: 10.14710/transient.7.1.123-130.
- [13] M. F. Rahman, D. Alamsah, M. I. Darmawidjadja, and I. Nurma, "Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN)," *J. Inform.*, vol. 11, no. 1, p. 36, 2017, doi: 10.26555/jifo.v11i1.a5452.