

## IMPLEMENTASI DEEP LEARNING DALAM PENGKLASIFIKASIAN WAJAH MENGGUNAKAN LIBRARY TENSORFLOW PADA ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)

Rahmat Setiawan Usman<sup>1</sup>, Dewi Rahmawaty Isa<sup>2</sup>, Isran K. Hasan<sup>3</sup>  
<sup>1,2,3</sup> Program Studi Statistika, Fakultas MIPA, Universitas Negeri Gorontalo

e-mail: [rahmatusan573@gmail.com](mailto:rahmatusan573@gmail.com)

### Abstrak

*Convolutional neural network* merupakan salah satu metode *deep learning* yang berfungsi untuk mengenali dan mengklasifikasikan objek pada suatu citra. Contoh penerapannya adalah sistem pengenalan wajah yang terdiri dari proses deteksi dan klasifikasi. Pengenalan wajah oleh komputer dapat dipengaruhi oleh banyak hal seperti pencahayaan, ekspresi dan jumlah dataset yang diberikan. Penelitian ini bertujuan untuk mengetahui bagaimana cara pengimplementasian CNN untuk mengidentifikasi wajah menggunakan *Tensorflow* dengan bahasa pemrograman *python*. Jumlah dataset yang digunakan adalah sebanyak 120 data dan 10 responden secara total dengan kondisi pencahayaan dan sudut pengambilan gambar yang berbeda-beda. Selain dari dataset, pada penelitian ini juga menggunakan beberapa skenario yang berbeda dalam proses pelatihannya, yaitu perbedaan jumlah *epoch* dan perbedaan jumlah *learning rate*. Berdasarkan hasil dari pembahasan, didapatkan dua model. Pada model pertama didapatkan hasil akurasi sebesar 100% pada proses training dan 65% pada proses *testing*. Pada model kedua didapatkan hasil akurasi 100% pada proses *training* dan 75% pada proses *testing*. sehingga kinerja model yang dibuat pada penelitian ini dapat dikatakan optimal dalam mengenali objek pada beberapa kondisi pencahayaan dan sudut gambar.

**Kata Kunci:** *Deep learning, Convolutional neural network (CNN), Tensorflow, Image classification*

### Abstract

*The convolutional neural network is a deep learning method that functions to recognize and classify objects in an image. An example of its application is a facial recognition system which consists of a detection and classification process. Facial recognition by computers can be influenced by many things such as lighting, expressions, and the amount of dataset provided. This study aims to find out how to implement CNN to identify faces using Tensorflow with the Python programming language. The number of datasets used is 120 data and 10 respondents in total with different lighting conditions and shooting angles. Apart from the dataset, this study also uses several different scenarios in the training process, namely the difference in the number of epochs and the difference in the number of learning rates. Based on the results of the discussion, two models were obtained. In the first model, the results obtained an accuracy of 100% in the training process and 65% in the testing process. In the second model, the results obtained are 100% accuracy in the training process and 75% in the testing process. performance of the model made in this study can be said to be optimal in recognizing objects in several lighting conditions and image angles.*

**Keywords :** *Deep learning, Convolutional neural network (CNN), Tensorflow, Image classification*

## 1. PENDAHULUAN

*Deep learning* merupakan implementasi dari machine learning yang menerapkan algoritma jaringan syaraf tiruan tetapi dengan *hidden layer* yang lebih banyak. Banyaknya lapisan ini digunakan di antara input dan output (Goodfellow, Bengio and Courville, 2016). Convolutional neural network merupakan salah satu metode deep learning yang berfungsi untuk mengenali dan mendeteksi objek pada *digital vision*. Convolutional neural network juga dapat diterapkan pada wajah dikarenakan wajah manusia merupakan salah satu biometrik dan memiliki keunikan untuk tiap-tiap individu. Pengenalan biometrik atau identifikasi biometrik merupakan 2 pengenalan secara otomatis didasarkan pada karakteristik yang unik atau berbeda di tiap personal. Setiap personal memiliki karakteristik yang berbeda sehingga kita dapat mengenal satu sama lain. Wajah merupakan salah satu biometrik yang dapat digunakan untuk mengidentifikasi seseorang. Pengidentifikasian wajah sering digunakan karena terdapat perbedaan yang spesifik pada tiap-tiap manusia (Tumuli, Najoran and Sambul, 2017).

Pengenalan wajah memiliki banyak peranan, salah satunya dalam mendorong produktivitas dan menjaga keselamatan publik. Proses pengidentifikasian wajah dapat dikelompokkan menjadi 2 bagian umum, yang pertama adalah pendeteksian citra wajah, dimana komputer dapat menentukan letak wajah pada suatu citra dan setelahnya, akan dilakukan proses pengidentifikasian citra wajah yang prosesnya antara lain adalah deteksi fitur, penghapusan latar, dan identifikasi citra wajah. Pada penerapannya, pengenalan wajah lebih gampang dibuat ketika seseorang dalam kondisi prima atau tidak ada gangguan lain dalam proses pengenalannya. Namun, pada nyatanya kita tidak dapat mengatur seseorang agar selalu dalam kondisi wajah prima maka dari itu penelitian ini dilakukan agar pengenalan wajah seseorang dapat lebih maksimal walaupun wajah tidak di kondisi prima atau pengenalan wajah di segala kondisi (Nurfita and Ariyanto, 2018).

Adapun beberapa tantangan dalam pengidentifikasian wajah diantaranya adalah :

1. Ekspresi yang ganjil. Ekspresi yang ganjil atau tidak biasa merupakan tantangan dalam pengenalan wajah
2. Wajah terhalang oleh suatu objek berupa kacamata, rambut, syal, topi ataupun barang lainnya
3. Perbedaan intensitas dalam gambar. Beberapa bagian gambar mungkin saja memiliki intensitas cahaya yang lebih tinggi atau lebih rendah dari bagian lainnya.
4. Adanya banyak objek selain wajah sampel yang muncul dalam gambar.
5. Rendahnya resolusi gambar.
6. Perbedaan warna kulit.
7. Jauhnya jarak antara kamera dan wajah.

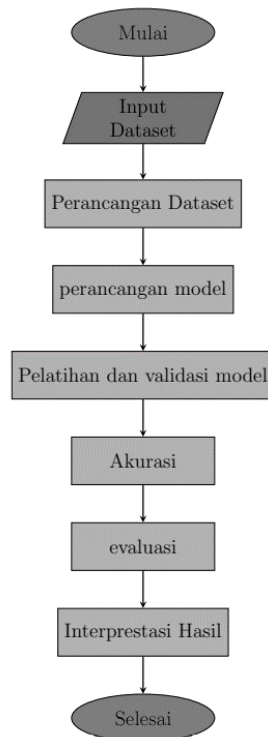
(Kumar, Kaur and Kumar, 2019)

Ada beberapa penelitian sebelumnya terkait topik ini seperti penelitian mengenai "*Convolutional neural network (CNN)* untuk pengenalan wajah secara *real-time*" oleh Muhammad Zufar dimana penelitian ini menggunakan *OpenCV python* dengan data citra wajah berukuran 48 x 48 pixel *grayscale* dan akurasi sebesar 87% (Zufar, 2016). Adapun penelitian lain yaitu "*CNN Modelling* untuk deteksi wajah berbasis gender menggunakan *Python*" oleh Warnia Nengsih, yang mana penelitian ini menggunakan salah satu *library* yaitu *keras* dan model yang digunakan menggunakan *pre-trained transfer learning* dari *Inception V3* dan menghasilkan akurasi sebesar 92,6% (Nengsih, 2020). Dari penelitian-penelitian sebelumnya maka penulis menerapkan beberapa hal yang berbeda diantaranya pada perancangan model menggunakan model yang belum di *training* dan juga pada pengambilan sampel menggunakan gambar RGB dan mengambil sampel yang lebih besar, oleh karena itu, penulis ingin melakukan penelitian pengidentifikasian wajah menggunakan

*tensorflow*. Penelitian ini bertujuan untuk mengetahui cara pengimplementasian *convolutional neural network* pada pengidentifikasian wajah, dan juga mengetahui hasil akurasinya pada berbagai nilai *epoch* dan *learning rate*.

## 2. METODE PENELITIAN

Penelitian menggunakan citra wajah dari 10 responden sebagai variabel penelitian dan diambil dari berbagai sisi. Penarikan sampel menggunakan *purposive sampling* dengan tujuan mendapatkan wajah yang berbeda secara spesifik. Perancangan dan pengolahan menggunakan *Microsoft VSCode* dan bahasa pemrograman *python*. Tahapan penelitian dapat dilihat dari diagram alir berikut :



**Gambar 1.** diagram alir penelitian

a. Perancangan dataset

Kumpulan data yang telah diambil kemudian dibagi dengan perbandingan 80:20 dan dilakukan pengubahan ukuran gambar agar ukuran dari gambar yang berbeda-beda menjadi setara dengan ukuran 150 x 150 piksel. Setelahnya, dilakukan pelabelan gambar yang berfungsi untuk memberikan dan menyimpan informasi dari tiap-tiap piksel gambar untuk dikenali.

b. Perancangan model CNN

Pada proses ini digunakan *library Tensorflow* yang ditulis dengan bahasa pemrograman *python*. Penggunaan *library* ini dikarenakan *tensorflow* memiliki mode *eager execution* yang memungkinkan untuk mengevaluasi setiap operasi secara terpisah (Ketkar, 2017).

c. Pelatihan dan validasi model

Pada tahap ini, dilakukan pelatihan dan validasi dari model yang telah dirancang. Proses ini dilakukan secara berulang dengan jumlah *epoch* dan *learning rate* yang berbeda untuk mengetahui model mana yang memiliki nilai akurasi paling tinggi dan nilai *loss* yang rendah.

d. Akurasi model

Model yang telah dibuat dan disimpan kemudian akan diuji tingkat keakuratannya. Nilai keakuratan yang dilihat hanya berdasarkan proses pelatihan dan validasi yang mana hal ini bergantung pada jumlah nilai *epoch* dan *learning rate* selama proses pelatihan.

e. Evaluasi model

Evaluasi model dilakukan untuk melihat bagaimana keakuratan model yang telah dibuat menggunakan data yang belum diberi label atau belum dilatih. Evaluasi model dilakukan dengan menggunakan *confusion matrix* dan juga menguji model secara langsung dengan dataset baru.

f. Output

Output pada penelitian ini adalah identifikasi wajah dari responden yang terklasifikasi dan juga model CNN yang telah dibuat.

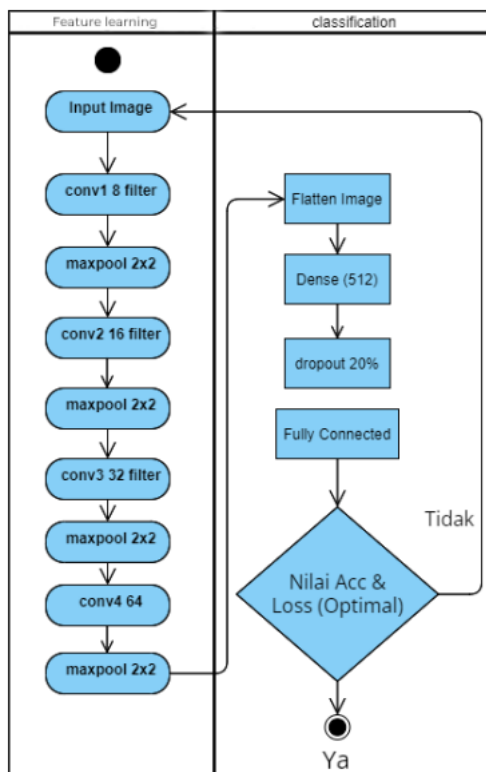
### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pengumpulan dataset

Data dikumpulkan menggunakan *Google form* dengan jumlah responden sebanyak 10 orang dan total data sebanyak 120 sampel. Data dari formulir tersebut lalu disimpan dalam dua folder terpisah yaitu *training* dan *testing* dengan perbandingan 80:20.

#### 3.2. Perancangan *Convolutional Neural Network*

Terdapat dua tahapan besar pada proses CNN, yaitu *feature learning* dan *classification*. Citra wajah yang dimasukkan ke dalam model menggunakan citra dengan besaran 150 x 150 x 3 piksel. Pada *feature learning* terdapat proses konvolusi dan *pooling* yang dilakukan sebanyak 4 kali secara berulang. Tiap-tiap konvolusi memiliki jumlah filter yang berbeda dan meningkat 2 kali lipat pada setiap proses konvolusi. Selanjutnya pada proses *pooling* menggunakan metode *max pooling* dengan matriks 2 x 2 yang berarti setiap matriks diambil nilai tertinggi untuk dimasukkan ke dalam model (Anggraini, 2020). Hasil dari proses tersebut menghasilkan *feature map* yang berbentuk matriks. Kemudian dilakukan proses *flatten* pada *feature map* yang dihasilkan yang berfungsi untuk mengubah matriks menjadi sebuah vector (O'Shea and Nash, 2015).



**Gambar 2.** Activity diagram CNN

Pada proses konvolusi pertama digunakan 8 filter, 3 x 3 kernel, dan 1 *strides*. Kemudian dilakukan proses *max pooling* menggunakan *pooling 2 x 2* dan *strides 2* yang berarti setiap pengambilan *pooling* dilakukan pergeseran sebanyak 2 langkah secara horizontal kemudian vertikal. Proses konvolusi dan *pooling* dilakukan sebanyak 4 kali dan tiap-tiap proses konvolusi diberikan *activation key* ReLU yang memiliki fungsi untuk mengubah nilai pada matriks citra selama proses dilakukan. Aktivasi tersebut mengubah nilai minus menjadi 0 pada matriks dan nilai lainnya tidak akan diubah (Nadira, 2019). Selanjutnya pada proses klasifikasi terdapat proses *flatten*, *densing*, dan *dropout* dimana pada *dense* menggunakan 512 parameter. Lalu kelas dari citra diklasifikasikan menjadi 10 kelas berdasarkan bobot nilai neuron pada *hidden layer* dengan menggunakan aktivasi *softmax classifier* dengan fungsi sebagai berikut :

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Dengan notasi  $f_j$  menunjukkan hasil perhitungan untuk setiap elemen ke- $j$  pada vektor output kelas. Argumen  $z$  adalah asumsi yang diberikan oleh model latih agar dapat diklasifikasikan oleh fungsi *softmax*. Hasil dari *softmax* lebih intuitif dan memiliki probabilitas yang lebih baik dan memungkinkan kita menghitung probabilitas pada semua label dan dapat diambil sebuah vektor dengan nilai nyata dan merubah nilai vektor dengan nilai dengan kisaran 0 sampai 1 (Qi, Wang and Liu, 2017) .

Setelah itu digunakan dilakukan penyesuaian *learning rate* dan *loss function* dimana pada penelitian ini digunakan *Adam Optimization* dan *cross-entropy loss function*. Proses tersebut menghasilkan model sebagai berikut :

**Tabel 1** Parameter CNN

Layer (type)	Output shape	Parameter
Konvolusi 1	(None, 150, 150,8)	224
Max pooling 1	(None, 75, 75, 8)	0
Konvolusi 2	(None, 75, 75, 16)	1.168
Max pooling 2	(None, 37, 37, 16)	0
Konvolusi 3	(None, 37, 37, 32)	4.640
Max pooling 3	(None, 18, 18, 32)	0
Konvolusi 4	(None, 18, 18, 64)	18.496
Max pooling 4	(None, 9, 9, 64)	0
Flatten	(None, 5184)	0
Dropout	(None, 5184)	0
Dense	(None, 512)	2.654.720
Dense	(None, 10)	5.130
Total		2.684.378

### 3.3. Pelatihan dan evaluasi model

Proses pelatihan ini bermaksud untuk melatih algoritma dalam mengenali data yang diberikan dan membentuk model berdasarkan proses pelatihannya. Pada proses ini digunakan 6 skenario *epoch* dan *learning rate* sedangkan pada proses evaluasi digunakan *confusion matrix* untuk mengetahui keakuratan dari model.

#### 1. Pelatihan dan validasi model

Proses pelatihan dilakukan sebanyak 6 kali dengan jumlah *learning rate* dan *epoch* yang berbeda. Hasil dari proses pelatihan adalah sebagai berikut :

- A. *Learning rate* 0,1 dan *epoch* 50 mendapatkan hasil akurasi pelatihan kurang dari 30% dengan grafik yang fluktuatif. Begitu pula pada nilai *loss* yang menunjukkan nilai yang sangat besar. Hal ini menunjukkan bahwa model mengalami *underfitting*.
- B. *Learning rate* 0,1 dan *epoch* 100 menunjukkan grafik fluktuatif dengan nilai akurasi sebesar 0% yang berarti semua nilai pada model ini hilang atau tidak dapat terdeteksi. Hal ini juga berarti model mengalami *underfitting*.
- C. *Learning rate* 0,01 dan *epoch* 50 menghasilkan akurasi yang cukup tinggi yaitu 60% tetapi memiliki nilai validasi yang sangat rendah. Nilai pada pelatihan dan validasi pada skenario ini berbanding terbalik. Hal ini menandakan bahwa model ini mengalami *overfitting*
- D. *Learning rate* 0,01 dan *epoch* 100 menghasilkan nilai akurasi pelatihan sebesar 10% yang menunjukkan bahwa model mengalami *underfitting*.
- E. *Learning rate* 0,001 dan *epoch* 50 mendapatkan nilai akurasi yang maksimal yaitu 100% dengan nilai *loss* dibawah 1. Maka dari itu, model ini akan dimasukkan ke tahap evaluasi.
- F. *Learning rate* 0,001 dan *epoch* 100 mendapatkan nilai akurasi yang sama dengan model *epoch* 50 hanya saja nilai *loss* yang diberikan menurun yaitu sebesar 0,2 dan model ini juga akan dilakukan proses evaluasi.

#### 2. Evaluasi model

Proses evaluasi menggunakan *confusion matrix* dengan cara memprediksikan dataset baru menggunakan model yang telah dibuat.

- A. Berikut adalah hasil pengujian model menggunakan *confusion matrix* pada model *learning rate* 0,001 dan *epoch* 50 :

**Tabel 2.** Hasil akurasi data uji model 1

confusion matrix	Predict class										
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	
actual class	C1	1	0	0	0	2	0	0	0	0	0
	C2	0	1	0	1	0	0	1	0	0	0
	C3	0	0	2	0	0	0	0	0	0	0
	C4	0	0	1	0	0	0	0	0	1	0
	C5	0	0	0	0	2	0	0	0	0	0
	C6	0	0	0	0	0	3	0	0	0	0
	C7	0	0	0	0	0	0	2	0	0	0
	C8	0	0	0	0	1	0	0	1	0	0
	C9	1	0	0	0	0	0	0	0	2	0
	C10	0	1	0	0	0	0	0	0	0	1

*Confusion matrix* menunjukkan bahwa dari 24 total data uji ada 8 data yang melenceng dari kelas yang sebenarnya. Dapat dilihat juga bahwa tidak ada dari kelas ke 4 yang dapat diidentifikasi sama sekali. Hal ini juga dapat dilihat pada nilai *precision*, *recall*, dan *accuracy*

**Tabel 3.** perbandingan nilai positif dan negatif model 1

Kelas	precision	Recall
C1	0,50	0,33
C2	0,50	0,33
C3	0,67	0,33
C4	0	0
C5	0,67	1
C6	0,60	1
C7	0,67	1
C8	1	0,50
C9	0,67	0,67
C10	1	0,5

Dengan nilai total akurasi yaitu

$$accuracy = \frac{true\ positive}{seluruh\ data\ testing} = \frac{15}{24} = 0,625$$

B. Hasil uji menggunakan *Confusion matrix* untuk model *learning rate* 0,001 dan *epoch* 100 adalah sebagai berikut :

**Tabel 4.** Hasil akurasi data uji model 2

Confusion Matrix	Predict class									
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	1	0	0	2	0	0	0	0	0	0
C2	0	3	0	0	0	0	0	0	0	0
C3	0	0	2	0	0	0	0	0	0	0
C4	0	0	0	2	0	0	0	0	0	0
actual class C5	0	0	0	0	2	0	0	0	0	0
C6	0	0	0	2	0	1	0	0	0	0
C7	1	0	0	0	0	0	1	0	0	0
C8	0	0	0	0	0	0	0	2	0	0
C9	0	1	0	0	0	0	0	0	2	0
C10	0	0	0	0	0	0	0	0	0	2

*Confusion matrix* menunjukkan bahwa setiap kelas memiliki nilai *true positive*. Dari ke-24 data hanya ada 6 yang tidak teridentifikasi dengan kelas yang sebenarnya dengan nilai *precision*, *recall* dan *accuracy* sebagai berikut :

**Tabel 5.** perbandingan nilai positif dan negatif model 2

Kelas	precision	Recall
C1	0,50	0,33
C2	0,75	1
C3	1	1
C4	0,33	1
C5	1	1
C6	1	0,33
C7	1	0,5
C8	1	1
C9	1	0,67
C10	1	1

Dengan nilai total akurasi sebesar 75% menjadikan model sebagai model dengan nilai akurasi tertinggi pada penelitian ini.

#### 4. KESIMPULAN

Pada implementasi CNN untuk pengidentifikasian wajah dilakukan dengan memasukkan citra wajah berbasis RGB 150 x 150 piksel dengan membagi data 20% data uji dan 80% data latih. Kernel yang digunakan berukuran 3 x 3, *strides* 1 dengan 4 lapisan filter dan metode *max pooling* digunakan di setiap satu kali filtrasi. Pada pelatihnnya, digunakan *epoch* 50 dan 100, skenario *learning rate* 0,1, 0,001, dan 0,0001 dengan total sebanyak 6 skenario pelatihan. Hasil dari tiap-tiap skenario adalah sebagai berikut :

- Learning rate* sebesar 0,1 dan *epoch* 50 menghasilkan akurasi sebesar 42%.
- Learning rate* sebesar 0,1 dan *epoch* 100 menghasilkan akurasi sebesar 12%.
- Learning rate* sebesar 0,01 dan *epoch* 50 menghasilkan akurasi sebesar 12%.
- Learning rate* sebesar 0,01 dan *epoch* 100 menghasilkan akurasi sebesar 8%.



- e. *Learning rate* sebesar 0,001 dan *epoch* 50 menghasilkan akurasi sebesar 62,5%.
- f. *Learning rate* sebesar 0,001 dan *epoch* 100 menghasilkan akurasi sebesar 75%.

## DAFTAR PUSTAKA

- Anggraini, W. (2020) *Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (CNN) Dengan Tensorflow*. UNIVERSITAS ISLAM NEGERI AR-RANIRY.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. USA: MIT PRESS.
- Ketkar, N. (2017) *Deep Learning with Python*. Berkeley, CA: Apress. doi: 10.1007/978-1-4842-2766-4.
- Kumar, A., Kaur, A. and Kumar, M. (2019) 'Face detection techniques: a review', *Artificial Intelligence Review*, 52(2), pp. 927–948. doi: 10.1007/s10462-018-9650-2.
- Nadira, M. (2019) *Implementasi Deep Learning dengan Metode Convolutional Neural Network untuk Identifikasi Citra Bahan Kulit Hewan*. UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAKARTA.
- Nengsih, W. (2020) 'CNN Modelling Untuk Deteksi Wajah Berbasis Gender Menggunakan Python', *Jurnal Komputer Terapan*, 6(2), pp. 190–199. doi: 10.35143/jkt.v6i2.3679.
- Nurfita, R. D. and Ariyanto, G. (2018) 'Implementasi Deep Learning berbasis Tensorflow untuk Pengenalan Sidik Jari', *Emitor: Jurnal Teknik Elektro*, 18(1).
- O'Shea, K. and Nash, R. (2015) 'An Introduction to Convolutional Neural Networks', *CoRR*, abs/1511.0.
- Qi, X., Wang, T. and Liu, J. (2017) 'Comparison of Support Vector Machine and Softmax Classifiers in Computer Vision', in *2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. IEEE, pp. 151–155. doi: 10.1109/ICMCCE.2017.49.
- Tumuli, A. D. L., Najoran, X. B. N. and Sambul, A. (2017) 'Implementasi Teknologi Biometrical Identification untuk Login Hotspot', *Jurnal Teknik Informatika*, 12(1). doi: 10.35793/jti.12.1.2017.17873.
- Zufar, M. (2016) *Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time*. Institut Teknologi Sepuluh Nopember.