



Artikel

# IMPLEMENTASI *DEEP LEARNING* MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK (CNN)* PADA EKSPRESI MANUSIA

Pulung Adi Nugroho<sup>1</sup>, Indah Fenriana<sup>2</sup>, Rudy Arijanto, M. Kom<sup>3</sup>

<sup>1,2,3</sup> Universitas Buddhi Dharma, Teknik Informatika, Banten, Indonesia

## SUBMISSION TRACK

Received: Agustus 17, 2020  
Final Revision: Agustus 21, 2020  
Available Online: September 1, 2020

## KEYWORD

*Deep Learning, Convolutional Neural Network, Klasifikasi Ekspresi Wajah, web flask*

## KORESPONDENSI

E-mail: [Pulungadi1497@gmail.com](mailto:Pulungadi1497@gmail.com)

## ABSTRAK

Dengan perkembangan teknologi yang semakin pesat, komputer dapat belajar sendiri seperti manusia, contohnya mengenali ekspresi wajah pada manusia, dengan membuat program yang biasa kita sebut *Mechine Learning*. *Mechine learning* dapat dibuat dengan banyak algoritma. Penulis membuat *Mechine Learning* dengan Metode *Deep Learning*, yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network (CNN)*. *CNN* dirancang khusus untuk pengenalan dan klasifikasi gambar. *CNN* memiliki beberapa lapisan (*layer*) yang mengekstrak informasi dari gambar dan menentukan klasifikasi dari gambar berupa skor klasifikasi. Aplikasi ini menggunakan Bahasa pemrograman *python*, web berbasis *flask*, *tensorflow*, dan *opencv*. Tahapan keseluruhan metode yang digunakan adalah tahap *pre-processing*, dan tahap klasifikasi. Proses training dilakukan dengan menggunakan *batch size*, dan *epoch* yang berbeda-beda untuk mendapatkan model terbaik. Dataset terdiri dari ekspresi senang, sedih, takut, jijik, netral, marah, dan kaget. Jumlah dataset tidak sepenuhnya mempengaruhi nilai akurasi, tetapi kedetailan citra untuk digunakan dataset sangat mempengaruhi hasil akurasi. Dengan *Epoch 100*, *Batch size 128* didapatkan hasil akurasi training sebesar 90% dan *validation* sebesar 65%. Hasil percobaan dari total 35 ekspresi, 28 ekspresi berhasil ditebak dengan benar dengan mendapatkan akurasi sebesar 80%.

## PENDAHULUAN

Ekspresi wajah adalah salah satu bentuk komunikasi *nonverbal* yang merupakan hasil dari satu atau lebih gerakan posisi otot pada wajah serta dapat menyampaikan keadaan emosi seseorang kepada orang yang mengamatinya.

Manusia dapat mengenali ekspresi wajah dengan baik, namun manusia tidak bisa mengklasifikasikannya, seiring

perkembangan teknologi informasi yang ada saat ini, kita bisa mengklasifikasikan ekspresi wajah dengan mesin, contohnya yaitu *Machine Learning*.

Klasifikasi dapat dilakukan dengan banyak metode, salah satunya yaitu menggunakan *Deep Learning*, *Deep Learning* merupakan bagian dari *Machine Learning* yang dapat mempelajari metode komputasinya sendiri.

Metode *Deep Learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network (CNN)*. Hal tersebut dikarenakan *CNN* berusaha meniru sistem pengenalan citra pada *visual cortex* manusia sehingga memiliki kemampuan mengolah informasi citra. Namun, *CNN* seperti metode *Deep Learning* lainnya, memiliki kelemahan yaitu proses pelatihan model yang lama. Dengan perkembangan perangkat keras, hal tersebut dapat diatasi menggunakan teknologi *General Purpose Graphical Processing Unit (GPGPU)*. *CNN* dirancang khusus untuk pengenalan dan klasifikasi gambar. *CNN* memiliki beberapa lapisan (*layer*) yang mengekstrak informasi dari gambar dan menentukan klasifikasi dari gambar berupa skor klasifikasi.

Oleh karena itu penulis ingin melakukan pengenalan ekspresi wajah dengan metode *Deep Learning* menggunakan *CNN* dengan dataset yang berbeda, dan menggunakan *dataset* yang jumlahnya lebih banyak, *dataset* diperoleh dari <https://www.kaggle.com/jonathanoheix/face-expression-recognition-dataset/version/1> *dataset* terdiri dari data *validation* dan data *train*, data *validation* berjumlah 28.821 ekspresi sedangkan data *train* berjumlah 7066 ekspresi, dan semua data terdiri dari gambar *grayscale* 48x48 piksel wajah.

Melihat dari latar belakang diatas penulis ingin membuat judul “Implementasi *Deep Learning* Menggunakan *CNN* Pada Ekspresi Manusia”.

## I. METODE

### Ekspresi

Ekspresi adalah salah satu bentuk komunikasi *nonverbal* yang merupakan hasil dari satu atau lebih gerakan posisi otot pada wajah serta dapat menyampaikan keadaan emosi seseorang kepada orang yang mengamatinya. Ekspresi wajah menyumbang 55% dalam penyampaian pesan, sedangkan bahasa dan suara masing-masing menyumbang 7% dan 38% . Manusia dapat mengetahui keadaan

emosi seseorang hanya dengan melihat ekspresi wajahnya, sedangkan komputer tidak mempunyai kemampuan ini. Pengolahan citra digital memungkinkan mesin atau komputer mengetahui keadaan emosi seseorang dengan pengenalan ekspresi wajah [5].

### Deep Learning

*Deep Learning* merupakan metode *learning* yang memanfaatkan *artificial neural network* yang berlapis-lapis (*multi layer*). *Artificial Neural Network* ini dibuat mirip otak manusia, dimana neuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit.

*Deep Learning* atau *deep structured learning* atau *hierarchical learning* atau *deep neural* merupakan metode *learning* yang memanfaatkan *multiple non-linear transformation*, *deep learning* dapat dipandang sebagai gabungan *machine learning* dengan *AI (artificial neural network)* [7].

Beberapa *algoritma* termasuk dalam kategori *Deep Learning* antara lain:

- a. *Convolutional Networks*
- b. *Restricted Boltzmann Machine (RBM)*
- c. *Deep Belief Networks (DBN)*
- d. *Stacked Autocoders*

### Convolutional Neural Network(CNN)

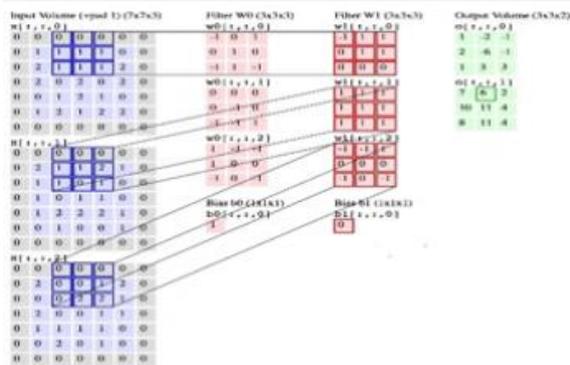
*Convolutional Neural Network (CNN)* adalah pengembangan dari *Multilayer Perceptron (MLP)* yang termasuk dalam *neural network* bertipe *feed forward* (bukan berulang). *Convolutional Neural Network* adalah *neural network* yang didesain untuk mengolah data dua dimensi. *CNN* termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [9]. *CNN* digunakan untuk menganalisis gambar visual, mendeteksi dan mengenali objek pada *image*, yang merupakan vektor berdimensi tinggi yang akan melibatkan banyak parameter untuk mencirikan jaringan. Secara garis besar, *CNN* tidak terlalu jauh berbeda dengan *neural network* biasanya. *CNN* terdiri dari *neuron* yang memiliki *weight*, *bias* dan *activation function*.



**Sumber: (Santoso and Ariyanto 2018)**  
**Gambar 1 CNN**

### Convolutional Layer

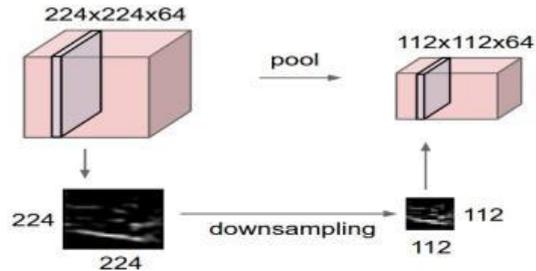
*Convolutional Layer* Seluruh data yang menyentuh lapisan *konvolusional* akan mengalami proses *konvolusi*. lapisan akan mengkonversi setiap filter ke seluruh bagian data masukan dan menghasilkan sebuah activation map atau *feature map 2D*. Filter yang terdapat pada *Convolutional Layer* memiliki panjang, tinggi dan tebal sesuai dengan channel data masukan. Setiap filter akan mengalami pergeseran dan operasi “dot” antara data masukan dan nilai dari filter. Lapisan konvolusional secara signifikan mengalami kompleksitas model melalui optimalisasi outputnya. Hal ini dioptimalkan melalui tiga parameter, *depth*, *stride* dan pengaturan *zero padding*.



Sumber: (Karpathy 2018)

Gambar 1 Proses Konvolusi dengan 2 filter untuk menghasilkan Feature Map/Activation map

*Pooling Layer* merupakan tahap setelah *Convolutional Layer*. *Pooling Layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu. Setiap pergeseran akan ditentukan oleh jumlah *stride* yang akan digeser pada seluruh area *feature map* atau *activation map*. Dalam penerapannya, pooling Layer yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Sebagai contoh, apabila kita menggunakan *Max Pooling 2x2* dengan *Stride 2*, maka pada setiap pergeseran filter, nilai yang diambil adalah nilai yang terbesar pada area 2x2 tersebut, Sedangkan *Average Pooling* akan mengambil nilai rata-rata.

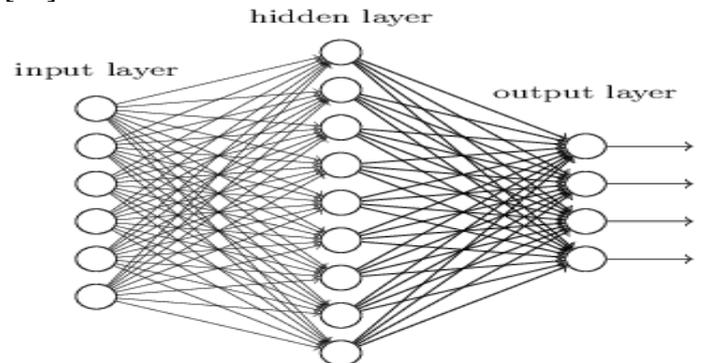


Sumber: (Santos, L. A. dos. 2018)

Gambar 2 Contoh Diagram MAX Pooling Layer

### Fully Connected Layer

*Feature map* yang dihasilkan oleh tahap sebelumnya berbentuk multidimensional array. Sehingga, Sebelum masuk pada tahap *Fully Connected Layer*, *Feature Map* tersebut akan melalui proses “flatten” atau *reshape*. Proses *flatten* menghasilkan sebuah vektor yang akan digunakan sebagai input dari *Fully Connected Layer*. *Fully Connected Layer* memiliki beberapa *Hidden Layer*, *Action Function*, *Output Layer* dan *Loss Function* [10].

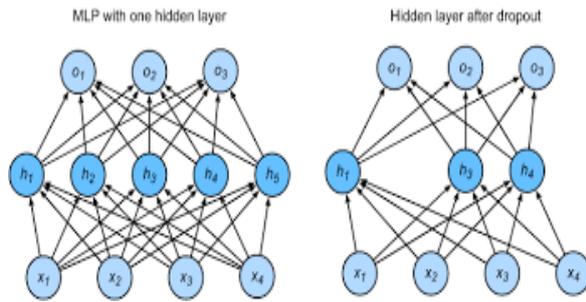


Sumber: (Murray 2015)

Gambar 3 Contoh Fully Connected Layer

### Dropout

*Dropout* merupakan salah satu usaha untuk mencegah terjadinya *overfitting* dan juga mempercepat proses learning. *Overfitting* adalah kondisi dimana hampir semua data yang telah melalui proses *training* mencapai persentase yang baik, tetapi terjadi ketidaksesuaian pada proses prediksi. Dalam sistem kerjanya, *Dropout* menghilangkan sementara suatu *neuron* yang berupa *Hidden Layer* maupun *Visible Layer* yang berada didalam jaringan [8].



Sumber: (Santoso and Ariyanto 2018)  
Gambar 4 dropout

### Flask

*Flask* adalah sebuah web framework yang ditulis dengan bahasa *Python* dan tergolong sebagai jenis *microframework*. *Flask* berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan *Flask* dan bahasa *Python*, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur *behaviour* suatu *web* dengan lebih mudah. *Flask* termasuk pada jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponennya seperti validasi form, *database*, dan sebagainya tidak terpasang secara *default* di *Flask*. Hal ini dikarenakan fungsi dan komponen-komponen tersebut sudah disediakan oleh pihak ketiga. *Flask* dapat menggunakan ekstensi yang membuat fitur dan komponen-komponen tersebut seakan diimplementasikan oleh *Flask* sendiri. Selain itu, meskipun *Flask* disebut sebagai *microframework* bukan berarti *Flask* mempunyai kekurangan dalam hal fungsionalitas. *Microframework* disini berarti bahwa *Flask* bermaksud untuk membuat core dari aplikasi ini sederhana mungkin tapi tetap dapat dengan mudah ditambahkan. Dengan begitu, fleksibilitas serta skalabilitas dari *Flask* dapat dikatakan cukup tinggi dibandingkan dengan *framework* lainnya.

### Python

*Python* merupakan bahasa pemrograman yang berorientasi obyek dinamis, dapat digunakan untuk bermacam-macam pengembangan perangkat lunak. *Python*

menyediakan dukungan yang kuat untuk integrasi dengan bahasa pemrograman lain dan alat-alat bantu lainnya. *Python* hadir dengan pustaka-pustaka standar yang dapat diperluas serta dapat dipelajari hanya dalam beberapa hari. Bahasa pemrograman yang interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Contoh kode program *python* dapat dilihat pada gambar dibawah ini:

```

1 print "Hello World"
2 print "Hello Again"
3 print "My Name Is 10-0"
4

```

Sumber: (Fitri, & dkk, 2017, p.23-24).  
Gambar 6 kode Python

### Open CV

*OpenCV (Open Source Computer Vision)* adalah library dari fungsi pemrograman untuk realtime visi komputer. *OpenCV* menggunakan lisensi BSD dan bersifat gratis baik untuk penggunaan akademis maupun komersial. *OpenCV* dapat digunakan dalam bahasa pemrograman *C*, *C++*, *Python*, *Java*, dan sebagainya. *OpenCV* dapat digunakan pada sistem operasi *Windows*, *Linux*, *Android*, *iOS* dan *Mac OS*. *OpenCV* memiliki lebih dari 2500 algoritma yang telah dioptimalkan [4].

**TensorFlow**

TensorFlow adalah open source library untuk machine learning yang di release oleh Google yang mendukung beberapa bahasa pemrograman [1]. Dalam proses Transfer Learning, Tensorflow berperan untuk memproses Inception-v3 Model untuk di training ulang menggunakan data yang baru dan kemudian menghasilkan classifier dengan komputasi yang cepat dan akurasi yang baik. Tensorflow dapat digunakan pada semua sistem operasi.

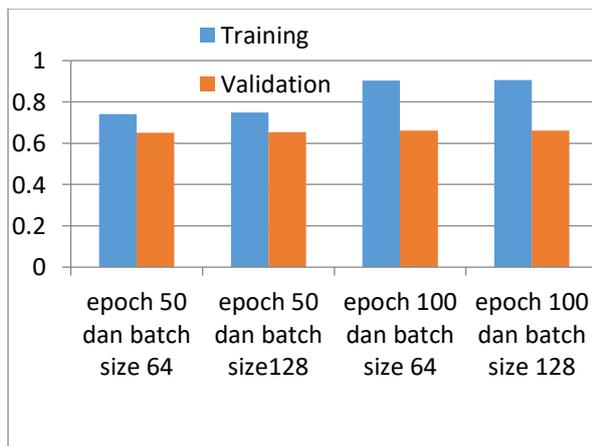
**II. HASIL**

Pada proses ini dilakukan pengujian terhadap data. Menggunakan data training pada 3993 citra ekspresi marah, 7164 citra ekspresi senang, 4938 citra ekspresi sedih, dan 3205 citra ekspresi kaget, 4982 citra ekspresi netral, 436 ekspresi citra jijik , dan 4103 ekspresi citra takut dengan menggunakan data latih 960 citra ekspresi marah, 1825 citra ekspresi senang, 1139 citra ekspresi sedih, 797 citra ekspresi kaget, 1216 ekspresi citra netral, 111 ekspresi citra jijik, dan 1018 ekspresi citra takut.

Pengujian dilakukan dengan epoch 50 dan 100, Learning Rate = 0.001, output layer = 7, mengganti nilai batch size. Menggunakan batch size 64 dan 128. Optimizer yang diuji adalah Adam. Hasil dari pengujian dapat dilihat table dibawah:

**Table**

**Table 1 Pengujian Data**



Berdasarkan hasil pengujian diatas, grafik yang menenunjukkan hasil akurasi learning paling bagus ditunjukkan pada proses epoch 100 dan batch size 128 yaitu dengan akurasi 0.9032 pada training dan 0.66 pada validation.

Hasil pengujian dengan model terbaik menghasilkan data citra yang terklarifikasi benar dapat dilihat pada table dibawah ini:

**Tabel 2. Tabel Confusion matrix**

	marah	jijik	takut	senang	netral	sedih	kaget
marah	0.53	0.01	0.08	0.05	0.16	0.14	0.03
jijik	0.15	0.62	0.05	0.01	0.05	0.08	0.03
takut	0.11	0	0.44	0.03	0.12	0.21	0.08
senang	0.02	0	0.02	0.82	0.07	0.05	0.02
netral	0.05	0	0.03	0.06	0.68	0.16	0.02
sedih	0.08	0	0.07	0.04	0.18	0.62	0.02
kaget	0.01	0	0.07	0.04	0.04	0.03	0.8

**Tabel 3. Tabel TP, FP, FN**

	True positive(TP)	False Positive(FP)	False Negative(FN)
marah	0.53	0.42	0.47
jijik	0.62	0.01	0.37
takut	0.44	0.32	0.55
senang	0.82	0.23	0.18
netral	0.68	0.62	0.32
sedih	0.62	0.67	0.39
kaget	0.8	0.2	0.19
Total	4.51	2.47	2.47

**Tabel 4. Tabel Recall dan Precision**

Precision dan Recall secara keseluruhan	
Recall	Precision
$\text{Recall} = \frac{TP}{(TP + FN)} \times 100\%$ $= \frac{4.51}{(4.51 + 2.47)} \times 100\%$ $= \frac{4.51}{6.98} \times 100\%$ $= 65\%$	$\text{Precision} = \frac{TP}{(TP + FP)} \times 100\%$ $= \frac{4.51}{(4.51 + 2.47)} \times 100\%$ $= \frac{4.51}{6.98} \times 100\%$ $= 65\%$

Dari data diatas juga didapatkan akurasi sebesar:

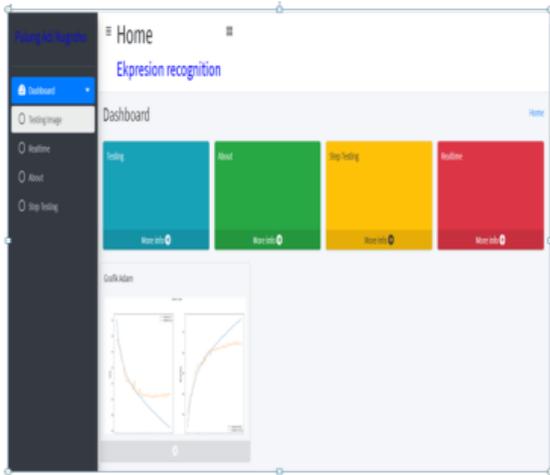
$$\text{Akurasi} = \text{TP} / \text{Total Dataset}$$

$$\text{Akurasi} = 4.51/6.98= 0.65$$

### III. PEMBAHASAN

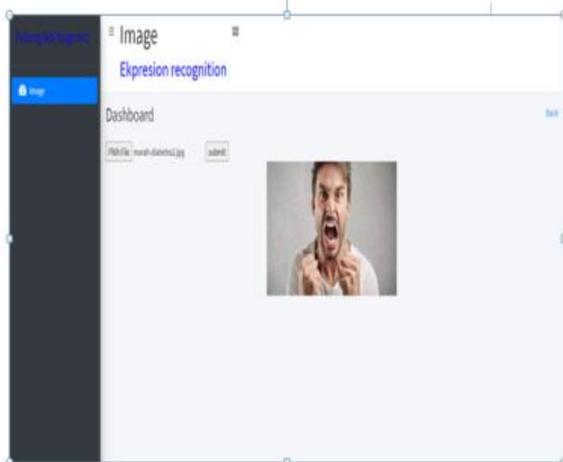
#### 3.1 Tampilan program

##### a. Tampilan awal



Gambar III.1 Tampilan Awal

##### b. Tampilan testing image



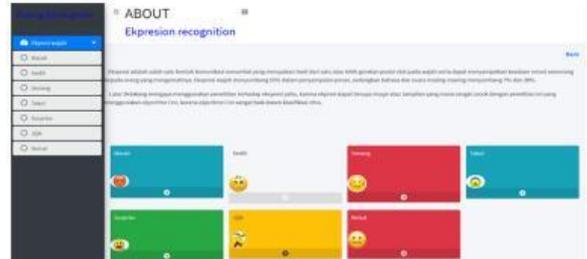
Gambar III.2 Tampilan Testing Image

##### c. Tampilan menu realtime



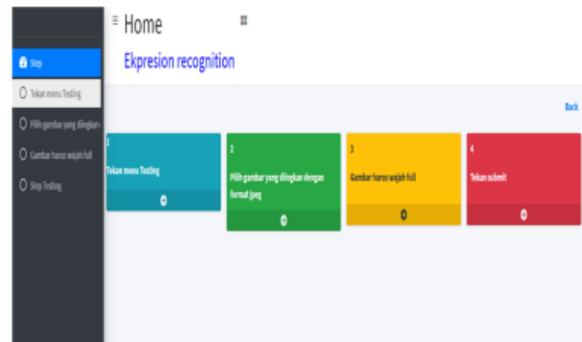
Gambar III.3 Tampilan menu realtime

##### d. Tampilan menu about



Gambar III.4 Tampilan Menu About

##### e. Tampilan menu step testing



Gambar III.5 Tampilan Menu step testing

### IV. SIMPULAN

Berdasarkan penelitian dan hasil penerapan metode *Convolutional Neural Network (CNN)* dalam mengklasifikasikan ekspresi wajah, dapat diambil kesimpulan sebagai berikut :

1. Jumlah dataset tidak sepenuhnya mempengaruhi nilai akurasi, tetapi kedetailan citra untuk digunakan dataset sangat mempengaruhi hasil akurasi.
2. Dari percobaan yang telah dilakukan didapatkan hasil *precision*, *recall*, dan akurasi sebesar 65%.
3. Metode *Convolutional Neural Network (CNN)* sangat cocok digunakan untuk menguji sebuah citra, karena prosesnya yang berlapis-lapis, terbukti dengan 35 citra, 28 citra bisa ditebak dengan benar walaupun ekspresi hanya berbeda tipis.

4. Penentuan *Batch size*, *Epoch*, *Optimizer*, dan jumlah *convolutional layer* sangat mempengaruhi akurasi yang akan didapatkan.
5. Diperlukan spec computer yang tinggi untuk melakukan proses learning.
6. Dengan *Epoch 100*, *Batch size 128* didapatkan hasil akurasi training sebesar 90% dan *validation* sebesar 65%.

### DAFTAR PUSTAKA

- [1]. Devikar, P. 2016. Transfer Learning for Image Classification of Various Dog Breeds. *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, Vol.5: 2707-2715.
- [2]. Fitri, Kiki Reski R, Ady Rahmansyah, dan Wahyuni . *Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D*, 2017.
- [3]. Karpathy, A. “CS231n Convolutional Neural Network for Visual.” *Stanford University, [Online].Available:*, 2018.
- [4]. Lazaro, Alvin , Joko Lianto Buliali, dan Bilqis Amaliah. “Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV.” *JURNAL TEKNIK ITS Vol. 6, No. 2*, 2017.
- [5]. Mustakim, Arif , Imam Santoso, dan Ajub Ajulian Zahra . “Pengenalan Ekspresi Wajah Manusia Menggunakan Tapis Gabor 2-D dan Support Vector Machine (SVM).” 2017. 2
- [6]. Murray, D. . “TensorFlow: Large-Scale Machine Learning on Heterogeneous.” 2015.
- [7]. Primartha, Rifkie. *Belajar Machine Learning Teori Dan Praktik*. Palembang, 2018. 1
- [8]. Santoso, Aditya, dan Gunawan Ariyanto . “Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah.” 2018. 5
- [9]. Suartika E. P, Arya Yudhi Wijaya Wijaya, dan Rully Soelaiman . “Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101.” 2016.
- [10]. Wahyono, Teguh. *Fundamental Of Python For Mechibe Learning*. Yogyakarta: Gava Media, 2018

## **BIOGRAFI**

**Pulung Adi Nugroho** lahir di KAB. Semarang, pada 14 Juli 1997. Dia adalah seorang mahasiswa Universitas Buddhi Dharma. Lulus di Program Studi Teknologi Informasi (S.Kom)..

**Indah Fenriana** menerima gelar Sarjana Komputer (S.Kom) dari Universitas Buddhi Dharma, Indonesia dan gelar Master dalam bidang Ilmu Komputer (M.Kom) dalam bidang Rekayasa Perangkat Lunak dari Universitas Eresha, Indonesia. Dia adalah dosen di Departemen Teknologi Informasi, Fakultas Sains dan Teknologi, Buddhi Dharma, Indonesia.

**Rudy Arijanto**, Saat ini bekerja sebagai dosen Tetap pada Program Studi Teknik Informatika di Universitas Buddhi Dharma.