

IMPLEMENTASI *DEEP LEARNING* BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI

Royani Darma Nurfiti, Gunawan Ariyanto

Program Studi Informatika Universitas Muhammadiyah Surakarta (UMS)
Surakarta, Indonesia

l200140038@student.ums.ac.id, ga154@ums.ac.id

Abstrak—Sistem pengenalan sidik jari banyak digunakan dalam bidang biometrik untuk berbagai keperluan pada beberapa tahun terakhir ini. Pengenalan sidik jari digunakan karena memiliki pola yang rumit yang dapat mengenali seseorang dan merupakan identitas setiap manusia. Sidik jari juga banyak digunakan sebagai verifikasi maupun identifikasi. Permasalahan yang dihadapi dalam penelitian ini adalah komputer sulit melakukan klasifikasi objek salah satunya pada sidik jari. Dalam penelitian ini penulis menggunakan *deep learning* yang menggunakan metode *Convolutional Neural Network* (CNN) untuk mengatasi masalah tersebut. CNN digunakan untuk melakukan proses pembelajaran mesin pada komputer. Tahapan pada CNN adalah input data, *preprocessing*, proses *training*. Implementasi CNN yang digunakan *library tensorflow* dengan menggunakan bahasa pemrograman python. Dataset yang digunakan bersumber dari sebuah website kompetisi verifikasi sidik jari pada tahun 2004 yang menggunakan sensor bertipe optical sensor “V300” by crossMatch dan didalamnya terdapat 80 gambar sidik jari. Proses pelatihan menggunakan data yang berukuran 24x24 pixel dan melakukan pengujian dengan membandingkan jumlah epoch dan *learning rate* sehingga diketahui bahwa jika semakin besar jumlah epoch dan semakin kecil *learning rate* maka semakin baik tingkat akurasi pelatihan yang didapatkan. Pada penelitian ini tingkat akurasi pelatihan yang dicapai sebesar 100%.

Kata Kunci : *Pengenalan sidik jari, Deep Learning, Convolutional Neural Network, Tensorflow, Python.*

I. PENDAHULUAN

Beberapa tahun terakhir sistem pengenalan sidik jari banyak digunakan dalam bidang teknologi biometrik untuk berbagai keperluan, misalnya pada smartphone, absensi, identifikasi maupun keamanan. Pengenalan sidik jari pada dasarnya adalah sistem pengenalan pola yang mengenali seseorang dengan menentukan keaslian sidik jarinya. Pengenalan sidik jari dapat digunakan sebagai sistem verifikasi yaitu mengotentikasi identitas seseorang dengan membandingkan citra sidik jari yang diambil dengan citra sidik jari yang telah disimpan di dalam sistem. Tahapan yang dilakukan untuk pengklasifikasian ada dua tahap. Pertama, citra sidik jari yang diinput akan melalui *preprocessing* yang rumit. Kedua, klasifikasi citra sidik jari[1]. Seperti yang dikemukakan oleh Maltoni et al (2009) bahwa

klasifikasi sidik jari adalah pengenalan pola yang telah mendapat perhatian besar untuk kesulitannya, karena variabilitas antar-kelas dan variabilitas intra-kelas yang besar[2].

Klasifikasi objek pada gambar secara umum menjadi salah satu masalah dalam visi komputer, bagaimana sebuah komputer dapat mencontoh kemampuan manusia dalam memahami informasi gambar, mengenali objek layaknya manusia, seperti mengenali kucing, anjing, pohon, mobil atau mengenali pola tertentu pada sebuah gambar. Bagi manusia hal ini menjadi pekerjaan yang sangatlah sederhana dan mudah, tetapi pada kenyataannya menjadi pekerjaan yang sukar bagi komputer, karena komputer hanya melihat nilai piksel dan data piksel sehingga sulit untuk diproses. Apalagi dengan berbagai variasi dari gambar sangat mempengaruhi pelatihan sehingga untuk mendapatkan hasil yang baik

menjadi lebih sulit dan mempengaruhi akurasi.

Pada Tugas Akhir ini akan mengimplementasikan *Deep Learning* dengan menggunakan metode *Convolutional Neural Network* dan mengaplikasikan *library Tensorflow* untuk mengetahui tingkat akurasi pelatihan yang dapat dicapai.

II. DASAR TEORI DAN TINJAUAN PUSTAKA

Deep Learning merupakan bagian dari *Machine Learning* yang terdiri dari banyak lapisan (*hidden layer*) dan membentuk tumpukan, lapisan tersebut adalah sebuah algoritma atau metode yang melakukan klasifikasi perintah yang diinput hingga menghasilkan output. Metode *Deep Learning* yang sedang berkembang salah satunya adalah *Convolutional Neural Network*. Jaringan ini menggunakan masukan berupa gambar, kemudian akan melalui lapisan konvolusi dan diolah berdasarkan filter yang ditentukan, setiap lapisan ini menghasilkan pola dari beberapa bagian citra yang memudahkan proses klasifikasi[3].

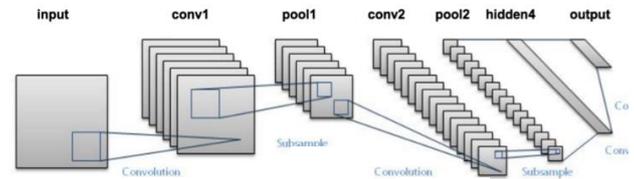
Perkembangan bidang *Deep Learning* saat ini telah dipermudah oleh banyaknya *library* dan *Application Program Interface (API)*. *Library* yang digunakan adalah *Tensorflow* yang merupakan antarmuka untuk mengekspresikan algoritma pembelajaran mesin dan untuk mengeksekusi perintah dengan menggunakan informasi yang dimiliki serta dapat membedakan objek satu dengan objek lainnya. *Tensorflow* memiliki fitur untuk menjalankan pelatihan model menggunakan *Central Processing Unit (CPU)* dan pelatihan model *Graphic Processing Unit (GPU)*. Namun dalam implementasi ini akan menjalankan pelatihan model dengan fitur *CPU*.

III. METODE

A. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu pengembangan dari jaringan syaraf tiruan yang terinspirasi dari

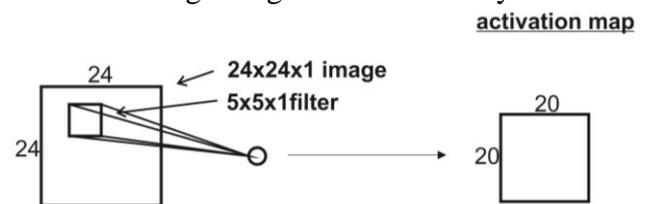
jaringan syaraf manusia dan biasa digunakan pada data gambar untuk mendeteksi dan mengenali suatu objek pada sebuah gambar. *CNN* terdiri dari neuron yang memiliki bobot, bias dan fungsi aktivasi. Alur proses *CNN* seperti Gambar 1.



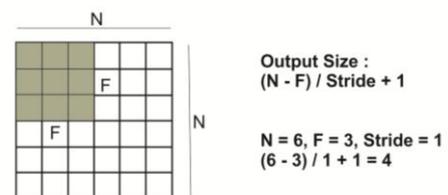
Gambar 1. Proses *Convolutional Neural Network* (Sumber: www.analyticsvidhya.com)

1. Convolutional Layer

Convolutional Layer bagian yang melakukan operasi konvolusi yaitu mengkombinasikan linier *filter* terhadap daerah lokal. *Layer* ini yang pertama kali menerima gambar yang diinputkan pada arsitektur. Bentuk *layer* ini adalah sebuah *filter* dengan panjang (pixel), lebar (pixel) dan tebal sesuai dengan *channel image* data yang diinputkan. Ketiga *filter* ini akan bergeser keseluruhan bagian gambar. Pergeseran tersebut akan melakukan operasi “dot” antara input dan nilai dari *filter* tersebut sehingga akan menghasilkan output yang disebut *activation map* atau *feature map*. Gambar 2 menampilkan proses konvolusi yang ada di dalam *convolution layer* dan Gambar 3 adalah cara menghitung nilai konvolusinya.



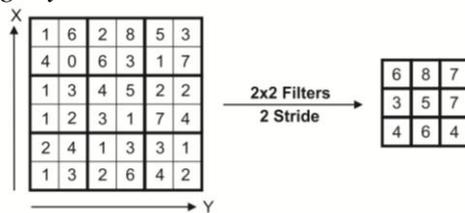
Gambar 2. Proses Konvolusi



Gambar 3. Rumus Menghitung Konvolusi

2. Pooling Layer

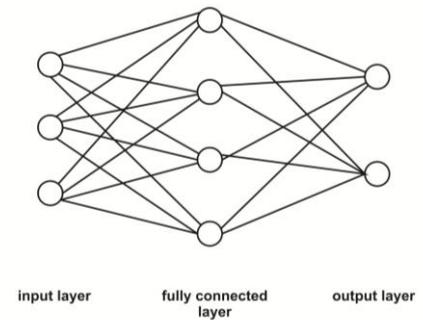
Pooling layer menerima output dari *convolution layer*, pada *layer* ini ukuran data citra akan direduksi. Prinsipnya *pooling layer* terdiri dari *filter* dengan ukuran tertentu dan *stride/langkah* kemudian bergeser keseluruh area *feature map*. Sebagian besar arsitektur CNN, metode *pooling* yang digunakan adalah *Max pooling*. *Max pooling* membagi output *convolution layer* menjadi beberapa grid kemudian setia pergeseran *filter* akan mengambil nilai terbesar dari setiap grid. Tergantung pada panjang langkahnya, gambar yang dihasilkan adalah sebagian kecil dari ukuran aslinya yang berguna untuk mengurangi dimensi data, sehingga mengurangi jumlah parameter pada langkah selanjutnya[4]. Gambar 4 menampilkan proses yang ada pada *pooling layer*.



Gambar 4. Proses *Pooling Layer*

3. Fully Connected Layer

Fully connected layer mengambil input dari hasil output *pooling layer* yang berupa *feature map*. *Feature map* tersebut masih berbentuk multidimensional array maka ini akan melakukan *reshape feature map* dan menghasilkan vektor sebanyak n-dimensi dimana n adalah jumlah kelas output yang harus dipilih program. Misalnya lapisan terdiri dari 500 neuron, maka akan diterapkan softmax yang mengembalikan daftar probabilitas terbesar untuk masing-masing 10 label kelas sebagai klasifikasi akhir dari jaringan[5]. Gambar 5 menampilkan proses yang ada dalam *fully connected layer*.



Gambar 5. Proses *Fully Connected Layer*

B. Perancangan Sistem

1. Menyiapkan Dataset

Dalam melaksanakan penelitian langkah pertama yang dilakukan adalah menyiapkan dataset yang berupa himpunan gambar sidik jari. Dataset digunakan untuk input yang akan diproses pada sistem. Pada penelitian ini akan menggunakan dataset dari <http://bias.csr.unibo.it/fvc2004/databases.asp> yang merupakan sebuah website kompetisi verifikasi sidik jari pada tahun 2004 yang menggunakan sensor bertipe optical sensor “V300” by CrossMatch berukuran 640x480 pixel resolusi 500dpi yang didalamnya ada 80 gambar sidik jari. Karena proses yang akan dijalani memerlukan gambar yang memiliki tinggi dan lebar yang sama maka setiap gambar akan dipotong menjadi 480x480 pixel. Gambar 6 menampilkan contoh gambar dari dataset yang digunakan.



Gambar 6. Contoh dataset sidik jari

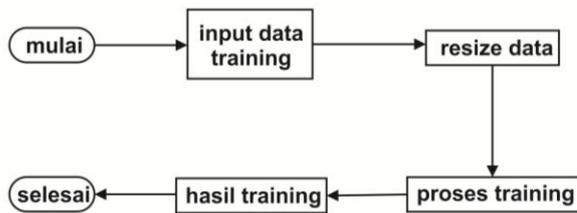
2. Data Preprocessing

Dataset yang digunakan adalah 480x480 pixel, ini menjadi ukuran yang sangat besar dan akan menyebabkan sistem terlalu berat untuk melakukan proses pelatihan, maka sebelum memasuki tahap pelatihan dilakukan

data *preprocessing* yaitu *resize* gambar menjadi 24x24 pixel.

3. Pelatihan Dataset

Pada tahap proses ini dataset yang telah siap akan dilatih dengan menggunakan metode *convolutional neural network*. Proses training ini merupakan tahapan dimana CNN dilatih untuk memperoleh akurasi yang tinggi dari klasifikasi yang dilakukan[6]. Alur prosesnya seperti pada Gambar 7.

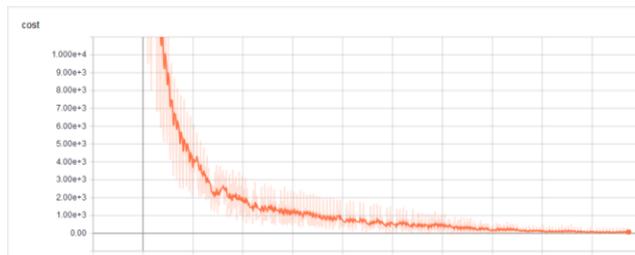


Gambar 7. Alur Proses Training

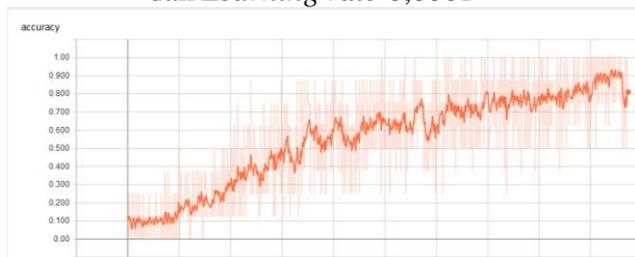
IV. HASIL DAN PEMBAHASAN

A. Pengujian Jumlah Epoch dan *Learning Rate*

Pengujian dilakukan untuk melihat pengaruh dari jumlah langkah pelatihan (epoch) dan jumlah *learning rate* yang digunakan terhadap performa sistem. Sebagai perbandingannya pengujian ini menggunakan jumlah epoch 150 dan epoch 250, sedangkan *learning rate* 0,0001, 0,001 dan 0,01. Hasil pengujian dapat dilihat pada Gambar 8.

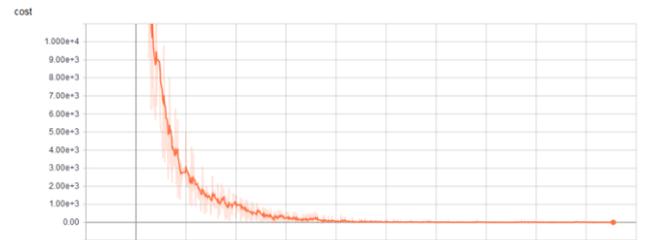


Gambar 8a. Training Loss sistem Epoch 150 dan *Learning rate* 0,0001

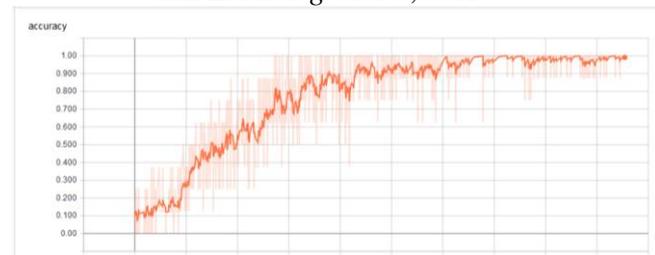


Gambar 8b. Training Accuracy sistem Epoch 150 dan *Learning rate* 0,0001

Berdasarkan Gambar 8a dan 8b, dapat diketahui bahwa dengan menggunakan langkah pelatihan 150 epoch dan *learning rate* 0,0001 menghasilkan tingkat akurasi 87,50%.



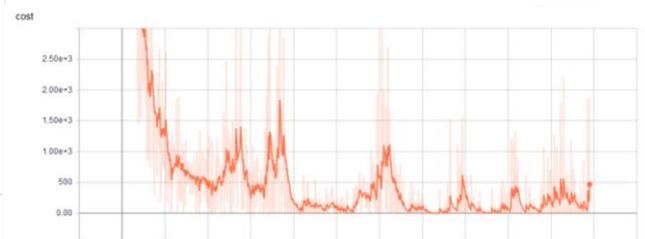
Gambar 9a. Training Loss sistem Epoch 250 dan *Learning rate* 0,0001



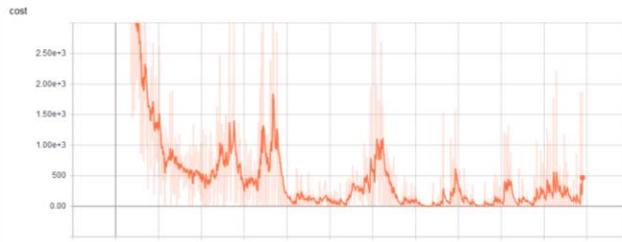
Gambar 9b. Training Accuracy sistem Epoch 250 dan *Learning rate* 0,0001

Berdasarkan Gambar 9a dan 9b, dapat diketahui bahwa dengan menggunakan langkah pelatihan 250 epoch dan *learning rate* 0,0001 mencapai tingkat akurasi 100%.

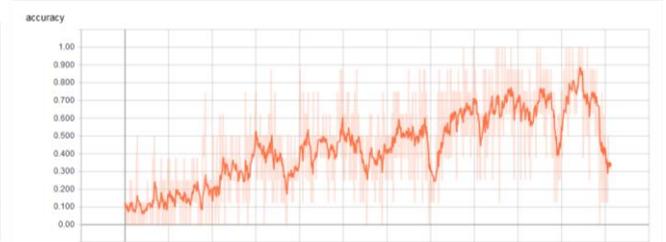
Dilihat dari gambar 8 dan 9, proses training yang dilakukan dapat dianalisis bahwa semakin banyak jumlah epoch maka semakin baik persentase akurasi data yang dihasilkan. Namun, semakin panjang jumlah epochnya maka semakin banyak juga waktu yang dibutuhkan untuk proses trainingnya.



Gambar 10a. Training Loss sistem Epoch 150 dan *Learning rate* 0,001



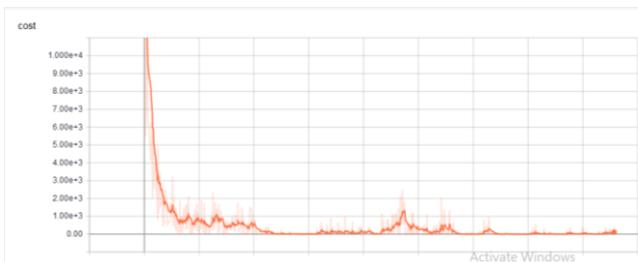
Gambar 10b. Training Accuracy sistem Epoch 150 dan *Learning rate* 0,001



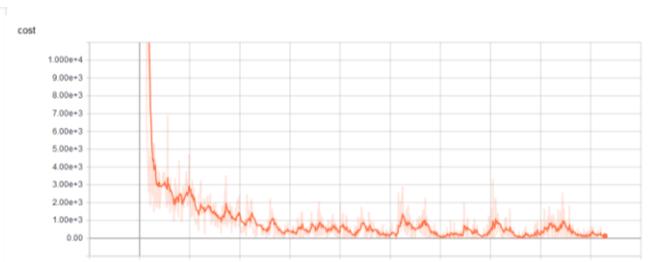
Gambar 12b. Training Accuracy sistem Epoch 150 dan *Learning rate* 0,01

Hasil training dengan epoch 150 dan *learning rate* 0,001 seperti yang ditampilkan pada Gambar 10a dan 10b mencapai akurasi 62,50%.

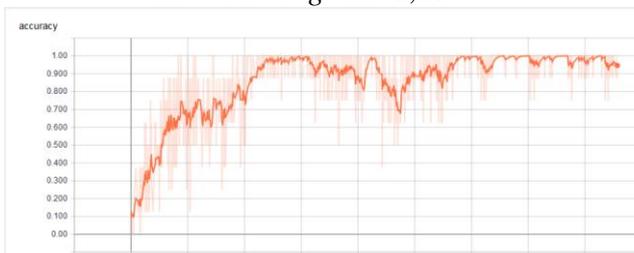
Berdasarkan Gambar 12a dan 12b, dapat dilihat bahwa dengan menggunakan langkah pelatihan 150 epoch dan *learning rate* 0,01 menghasilkan tingkat akurasi 37,50%.



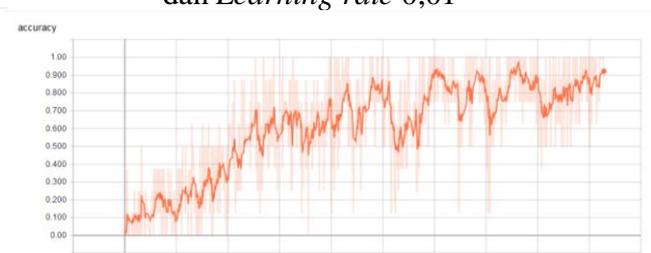
Gambar 11a. Training Loss sistem Epoch 250 dan *Learning rate* 0,001



Gambar 13a. Training Loss sistem Epoch 250 dan *Learning rate* 0,01



Gambar 11b. Training Accuracy sistem Epoch 250 dan *Learning rate* 0,001



Gambar 13b. Training Accuracy sistem Epoch 250 dan *Learning rate* 0,01

Pada Gambar 11a dan 11b, dengan menggunakan langkah pelatihan 250 epoch dan *learning rate* 0,001 mencapai akurasi 87,50%.

Berdasarkan Gambar 13a dan 13b, dapat dilihat bahwa dengan langkah pelatihan 250 epoch dan *learning rate* 0,01 menghasilkan tingkat akurasi 87,50%.

Dilihat dari hasil pengujian yang telah dilakukan dapat dilihat bahwa dengan menggunakan epoch yang semakin banyak hasilnya semakin baik dan dengan menggunakan *learning rate* yang semakin besar mengakibatkan tingkat akurasi yang buruk dan proses pelatihannya tidak stabil.



Gambar 12a. Training Loss sistem Epoch 150 dan *Learning rate* 0,01

B. Hasil Pengujian dan Analisis

Pengujian yang dilakukan menggunakan 150 epoch dan 250 epoch sedangkan *learning*

rate yang digunakan adalah 0,0001, 0,001, dan 0,01.

Tabel 1. Hasil dan Pengujian sistem

Jumlah Dataset	Image Size	Epoch	Learning rate	Accuracy (%)
80	24x24 px	150	0,0001	87,50
		150	0,001	62,50
		150	0,01	37,50
		250	0,0001	100
		250	0,001	87,50
		250	0,01	87,50

Berdasarkan tabel hasil pengujian sistem, dapat diketahui bahwa dataset berjumlah 80 dengan ukuran 24x24 px menggunakan epoch 150 pada saat proses training menggunakan *learning rate* 0,0001 menghasilkan tingkat akurasi 87,50%, menggunakan *learning rate* 0,001 menghasilkan akurasi 62,50% sedangkan dengan *learning rate* 0,01 menghasilkan tingkat akurasi 37,50%. Dan pengujian dengan epoch 250 menggunakan *learning rate* 0,0001 menghasilkan akurasi 100%, dengan *learning rate* 0,001 menghasilkan akurasi 87,50% dan dengan menggunakan *learning rate* 0,01 juga menghasilkan 87,50%. Dari perbandingan tersebut dapat diketahui bahwa jumlah epoch dan *learning rate* sangat berpengaruh pada tingkat akurasi. Semakin besar jumlah epoch maka semakin baik tingkat akurasinya, sedangkan jika semakin besar *learning rate* maka semakin buruk hasil akurasinya dan dalam proses pelatihannya tidak stabil.

V. KESIMPULAN

Penelitian ini berhasil mengimplementasikan *Deep Learning* dengan menggunakan metode *Convolutional Neural Network* dengan *library* tensorflow menghasilkan tingkat akurasi pelatihan yang baik dengan akurasi training sebesar 100%.

Proses training yang dilakukan menggunakan beberapa perbandingan antaranya jumlah epoch dan jumlah *learning rate*. Sehingga dapat disimpulkan bahwa jika semakin banyak epoch dan semakin kecil *learning rate* maka semakin baik hasil akurasi pelatihan yang didapatkan.

DAFTAR PUSTAKA

- [1] Jeon, W. S., & Rhee, S.Y. (2017). Fingerprint Pattern Classification Using Convolutional Neural Network. *International Journal of Fuzzy Logic and Intelligent System*. Vol.17, No.3, pp. 170-176. <http://dx.doi.org/10.5391/IJFIS.2017.17.3.170>
- [2] Mishelsanti, D., Guichi, Y., Ene, A-D., Stef, R., Nasrollahi, K., & Moeslund, T. B. (2017). Fast Fingerprint Classification with Deep Neural Network. In *VISAPP – International Conference on Computer Vision Theory and Applications*
- [3] Danukusumo, K.P., (2017). Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GUI. Skripsi. Universitas Atma Jaya Yogyakarta
- [4] Shukla, N., & Fricklas, K. (2018). Machine Learning with Tensorflow. Manning Publications
- [5] Dutt, Anuj, & Dutt, Aashi. (2017). Handwritten Digit Recognition Using Deep Learning. *International journal of Advanced Research in Computer Engineering & Technology (IJARCET)*. Volume 6, Issue 7
- [6] I Wayan, S. E. P., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS* Vol.5 No.1