IMPLEMENTASI MACHINE LEARNING DENGAN MENGGUNAKAN CNN (*CONVOLUTIONAL NEURAL NETWORK*) UNTUK KLASIFIKASI CITRA CANDI

Rikky Fajri[1], Linda Atika[2]
Fakultas Teknik Ilmu Komputer, Universitas Bina Darma
Email: rickyfajri1111@gmail.com[1], lindaatika@binadarma.ac.id[2]

ABSTRACT

One of the Machine Learning methods that is currently being developed is the Convolutional Neural Network method. This network is made with the assumption that the input used is in the form of images. CNN has been optimized to facilitate research in the field of Machine Learning, so that all basic techniques can be implemented faster and easier than if done using just plain Python code. The benefits of CNN for computer science education are the algorithms used are inspired by how the human brain works. Some people may be more familiar with artificial neural networks.

Keywords: *Machine Learning, CNN, Python*

ABSTRAK

Salah satu metode Machine Learning yang sedang berkembang saat ini adalah metode Convolutional Neural Network jaringan ini dibuat dengan asumsi bahwa masukan yang digunakan adalah berupa gambar. CNN telah dioptimasi untuk mempermudah penelitian pada bidang Machine Learning, sehingga semua teknik dasar dapat di implementasikan lebih cepat dan mudah dibanding jika dilakukan hanya dengan menggunakan sekedar kode Phyton biasa . Manfaat CNN bagi pendidikan ilmu komputer ialah algoritma yang dipakai terinspirasi dari bagaimana otak manusia bekerja. Beberapa orang mungkin lebih mengenalnya dengan jaringan syaraf tiruan.

Kata Kunci : Machine Learning, CNN, *Phyton*

1. PENDAHULUAN

Klasifikasi citra merupakan salah satu bagian permasalahan dalam *Computer Vision.* Tujuan dari klasifikasi citra adalah untuk mengklasifikasikan masukan citra kedalam beberapa kategori tertentu. Ide dari klasifikasi citra sendiri adalah dengan memberi komputer masukkan dari kumpulan angka yang diproses dari citra dan akan menghasilkan angka yang merepresentasikan kategori dari citra tersebut. Terlintas hal ini terlihat seperti pekerjaan yang sepele dan mudah bagi manusia, namun kenyataannya hal ini merupakan pekerjaan yang sangat sulit bagi komputer. Permasalahan ini dikarenakan komputer hanya melihat citra berupa deretan nilai piksel dan data piksel ini sangatlah sulit untuk diproses.

Variasi dari banyak citra dengan kasus seperti ini akan membuat fitur pelatihan yang sangat besar sehingga sulit untuk mendapatkan hasil yang baik. Tetapi kurangnya dari variasi pada citra yang dipelajari bukan berarti akan semakin mudah untuk mendapat hasil yang baik. Misal pada dataset candi yang digunakan pada penelitian ini, bentuk dan corak pada candi pada citra di dataset kebanyakan memiliki pola yang hampir sama. Kesamaan pola antar kategori yang akan diklasifikasi dapat menyebabkan kesalahan dalam memprediksi, karena fitur yang dipelajari membuat model tidak dapat melakukan prediksi dengan baik. Model pembelajaran untuk kasus seperti ini akan lebih

baik dengan menerapkan konsep jaringan syaraf tiruan dengan banyak lapisan, dan dengan model tersebut maka akan lebih baik lagi memanfaatkan performa komputasi yang cepat. Disinilah dimana kita perlu menggunakan teknik *Machine Learning*.

Teknologi *machine learning* (ML) adalah mesin yang dikembangkan untuk bisa belajar dengan sendirinya tanpa arahan dari penggunanya. Machine learning adalah pendekatan prediktif yang mampu memprediksi luaran suatu sistem berbasis pada pola yang sudah ada [1]. Pembelajaran mesin dikembangkan berdasarkan disiplin ilmu lainnya seperti statistika, matematika dan *data mining* sehingga mesin dapat belajar dengan menganalisa data tanpa perlu di program ulang atau diperintah.

2. METODOLOGI PENELITIAN

Pada penelitian ini metode analisis data yang digunakan adalah metode *Convolutional Neural Network* yang bertujuan untuk mengklasifikasikan citra mata. *Convolutional Neural Network* (CNN) merupakan salah satu metode deep learning yang konsep kerjanya terinspirasi oleh jaringan syaraf alami manusia [2]. Sedangkan *software* yang digunakan pada penelitian ini adalah *software* Python 3.6.2. Python merupakan bahsa pemprograman yang sedang berkembang dan merupakan bahasa pemprograman yang tidak memiliki editor bawaan [3].Metode pengumpulan data merupakan langkah yangpaling strategis, sesuai tujuan utama dari penelitian yaitu mendapatkan data. Dalam penyusunan penelitian ini penulis mengumpulkan data yang dibutuhkan dalam pengujian pengenalan huruf menggunakan metode pengumpulan data sebagai berikut :

1. Pengamatan (*Observation*), peneliti melakukan peninjauan pada objek penelitian.
2. Studi kepustakaan (*Literature*), data diperoleh melalui studi kepustaka (*literature*) yaitu dengan mencari bahan dari internet, jurnal dan perpustakaan serta buku yang sesuai dengan objek yang akan diteliti.

3. HASIL DAN PEMBAHASAN

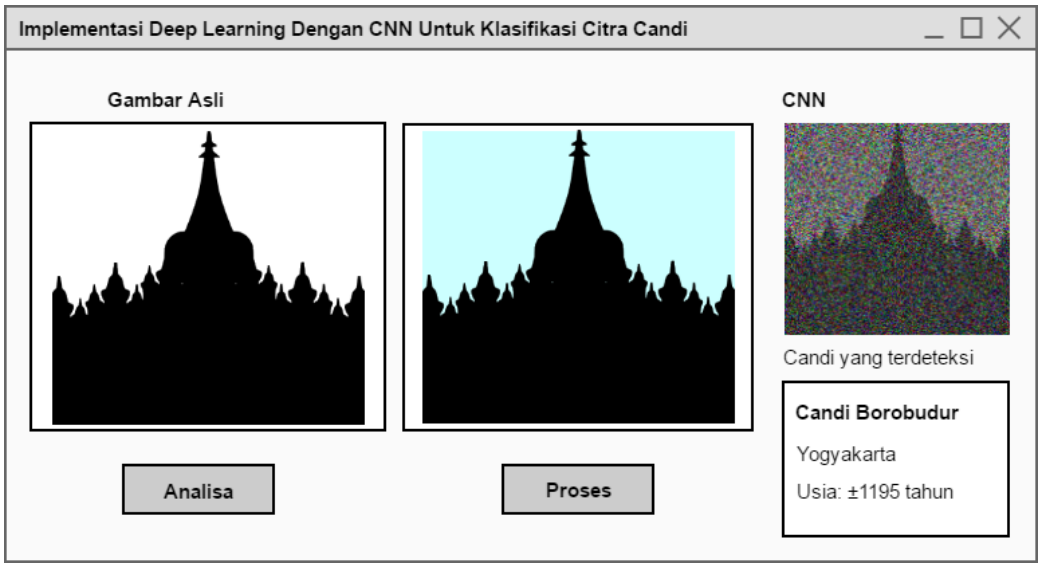
3.1 Analisa

Proses pada penelitian ini yaitu mengimplementasi metode *algoritma convolutional neural network* (*CNN*) melalui *Machine learning* dalam klasifikasi citra candi. Karena untuk mengatasi masalah melalui penelitian ini yang bertujuan agar dengan kemajuan teknologi masyarakat juga dapat mempelajari candi yang merupakan peninggalan budaya bangsa. Maka dari itu melalui penelitian ini menghadirkan solusi yaitu membangun mengimplementasikan citra candi untuk diklasifikasi melalui algoritma *convolutional neural network* (*CNN*). Berdasarkan permasalahan tersebut, maka penulis membatasi masalah dalam penelitian ini, untuk klasifikasi citra yang lebih efektif dan lebih mudah dalam mengenali dan mengolah data candi. Sedangkan, dalam tahapan komunikasi penulis melakukan observasi yang di lakukan pada penelitian implementasi *Machine learning* dengan menggunakan *CNN (convolutional neural network)*untuk klasifikasi citra candi ini, hasil dari observasi tersebut adalah aplikasi yang dapat membantu dalam proses klasifikasi citra candi. Dimana pada tahap ini *software engineer*, *user* bertemu dan mendefinisikan tujuan perangkat lunak yang diinginkan secara *overall* dan membahas mengenai penentuan sistemsecara ringkas dan fitur-fitur yang nantinya akan disediakan sistem serta konsep aplikasi yang akan dibuat.

3.2 Perancangan

Setelah melakukan analisa melalui tahap komunikasi, peneliti melakukan tahap perancangan yaitu untuk merumuskan rencana sistem yang akan dibuat, yaitu mengimplementasikan *algoritma convolutional neural network* (*CNN*) dalam klasifikasi citra candi, membuat perancangan sementara yang berpusat pada penyajian kepada pengguna. Kemudian, untuk memperjelas rancangan, dilakukan pemodelan perancangan secara cepat. Evaluasi ini dilakukan oleh pengguna apakah metode yang sudah dibangun sudah sesuai dengan keinginan pengguna. Jika sudah sesuai maka langkah keempat akan diambil.

Pada rancangan struktur menu ini menampilkan form untuk menganalisa citra candi. Kemudian, citra tersebut diproses dengan algoritma *convolutional neural network*dan dapat menampilkan citra yang terklasifikasi. Berikut adalah tampilan rancangan struktur menu yaitu:



Gambar 1. Tampilan rancangan struktur menu

Penggunaan dataset pada metode CNN yaitu berupa data gambar. Model CNN akan berjalan dengan baik ketika menggunakan data train gambar yang banyak. Sehingga sebuah model dapat belajar mengenali gambar tersebut. Dataset yang digunakan dalam penelitian ini berupa gambar yang dikumpulkan melalui github. Data gambar yang digunakan kali ini adalah gambar mata normal, katarak, disease dan galucoma. Pengumpulan dataset jika dilakukan secara manual akan memakan waktu yang cukup lama. Sehingga peneliti menggunakan metode *java script* dan python. Program *javascript* bertujuan untuk pengambilan *URL* gambar yang terdapat pada *google* dan program python yang melakukan eksekusi untuk men*download* gambar tersebut. Setelah melakukan proses pembuatan data dalam bentuk *.txt*, kemudian untuk melakukanproses download gambar kali ini menggunakan program python Berikut adalah kode yang dimasukan kedalam program *python* :

```
(cnnimage) C:\Users\User\Desktop\CNN Binary Image Classification Riki>python ./CnnBinaryImageClassification.py
2021-03-25 18:24:29.162899: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Libr
ary (oneDNN)to use the following CPU instructions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Found 1 images belonging to 1 classes.
Found 1 images belonging to 1 classes.
2021-03-25 18:24:30.383855: I tensorflow/core/profiler/lib/profiler_session.cc:164] Profiler session started.
WARNING:tensorflow:From ./CnnBinaryImageClassification.py:88: Model.fit_generator (from tensorflow.python.keras.engine.training) is deprecated and will b
e removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/25
1/1 [==============================] - 1s 609ms/step - loss: 0.7505 - accuracy: 0.0000e+00 - val_loss: 0.0655 - val_accuracy: 1.0000
Epoch 2/25
2021-03-25 18:24:35.677210: I tensorflow/core/profiler/lib/profiler_session.cc:164] Profiler session started.
WARNING:tensorflow:From C:\Users\User\.conda\envs\cnnimage\lib\site-packages\tensorflow\python\ops\summary_ops_v2.py:1277: stop (from tensorflow.python.e
ager.profiler) is deprecated and will be removed after 2020-07-01.
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
2021-03-25 18:24:36.185185: I tensorflow/core/profiler/rpc/client/save_profile.cc:176] Creating directory: logs/Foo_2021-03-25T18-24-30\train\plugins\prof
file\2021_03_25_11_24_35
2021-03-25 18:24:36.242661: I tensorflow/core/profiler/rpc/client/save_profile.cc:182] Dumped gzipped tool data for trace.json.gz to logs/Foo_2021-03-25T
18-24-30\train\plugins\profile\2021_03_25_11_24_35\FROM-ZERO.trace.json.gz
2021-03-25 18:24:36.363654: I tensorflow/core/profiler/rpc/client/save_profile.cc:176] Creating directory: logs/Foo_2021-03-25T18-24-30\train\plugins\prof
file\2021_03_25_11_24_35
2021-03-25 18:24:36.485044: I tensorflow/core/profiler/rpc/client/save_profile.cc:182] Dumped gzipped tool data for memory_profile.json.gz to logs/Foo_20
21-03-25T18-24-30\train\plugins\profile\2021_03_25_11_24_35\FROM-ZERO.memory_profile.json.gz
2021-03-25 18:24:36.709121: I tensorflow/python/profiler/internal/profiler_wrapper.cc:111] Creating directory: logs/Foo_2021-03-25T18-24-30\train\plugins
\profile\2021_03_25_11_24_35Dumped tool data for xplane.pb to logs/Foo_2021-03-25T18-24-30\train\plugins\profile\2021_03_25_11_24_35\FROM-ZERO.xplane.pb
Dumped tool data for overview_page.pb to logs/Foo_2021-03-25T18-24-30\train\plugins\profile\2021_03_25_11_24_35\FROM-ZERO.overview_page.pb
Dumped tool data for input_pipeline.pb to logs/Foo_2021-03-25T18-24-30\train\plugins\profile\2021_03_25_11_24_35\FROM-ZERO.input_pipeline.pb
Dumped tool data for tensorflow_stats.pb to logs/Foo_2021-03-25T18-24-30\train\plugins\profile\2021_03_25_11_24_35\FROM-ZERO.tensorflow_stats.pb
Dumped tool data for kernel_stats.pb to logs/Foo_2021-03-25T18-24-30\train\plugins\profile\2021_03_25_11_24_35\FROM-ZERO.kernel_stats.pb

1/1 [==============================] - 0s 141ms/step - loss: 0.1235 - accuracy: 1.0000 - val_loss: 0.0055 - val_accuracy: 1.0000
Epoch 3/25
1/1 [==============================] - 0s 109ms/step - loss: 0.0169 - accuracy: 1.0000 - val_loss: 3.2004e-04 - val_accuracy: 1.0000
Epoch 4/25
1/1 [==============================] - 0s 234ms/step - loss: 0.0021 - accuracy: 1.0000 - val_loss: 1.6264e-05 - val_accuracy: 1.0000
Epoch 5/25
1/1 [==============================] - 0s 172ms/step - loss: 2.8544e-04 - accuracy: 1.0000 - val_loss: 8.4343e-07 - val_accuracy: 1.0000
```

Gambar 2. Kode Program Python

Setelah dilakukan pembuatan data, langkah selanjutnya adalah melakukan pelatihan model CNN. Umumnya dalam CNN memiliki 2 tahapan, yaitu tahap *feature learning* dan *classification.* Input gambar pada model CNN menggunakan citra yang berukuran 64x64x3. Angka tiga yang dimaksud adalah sebuah citra yang memiliki 3 *channel* yaitu *Red*, *Green*, dan *Blue* (RGB) Citra masukan kemudian akan diproses terlebih dahulu melalui proses konvolusi dan proses pooling pada tahapan *feature learning.* Jumlah proses konvolusi pada rancangan ini memiliki dua lapisan konvolusi. Setiap konvolusi memiliki jumlah filter dan ukuran kernel yang berbeda. Kemudian dilakukan proses *flatten* atau proses mengubah *feature map* hasil *pooling layer* kedalam bentuk vector. Proses ini biasa disebut dengan tahap *fully Connected layer*.

Pengujian ini dilakukan ntuk melakukan evaluasi terhadap model yang dihasilkan oleh CNN. Pengujian ini dilakukan dua tahap yaitu tahapan training dan *testing*. Tahap training adalah tahap dimana model CNN diuji dengan data latih yang sudah disediakan. Pada tahap ini model di uji dengan gambar yang berbeda dengan tujuan menguji apakah model sudah menghasilkan performa yang baik dalam mengklasifikasikan sebuah gambar. Tahap selanjutnya adalah melakukan pelatihan model dijalankan dengan membuat directory dengan nama folder “data” terlebih dahulu yang bertujuan untuk penyimpanan data. Dalam directory tersebut dibuat dua folder yaitu data train dan validation. Setelah pembuatan *directory* selanjutnya adalah penentuan parameter dari model CNN.Input gambar pada model ini adalah 64x64 dan 150x150 . Untuk *batch sizenya* berukuran 32. *Batch size* adalah jumlah sampel yang disebarkan ke dalam arsitektur *neural network*. *Sampleper epoch* adalah jumlah sampel yang digunakan dalam tahap pelatihan. Jumlah sampel yang digunakan sebanyak 240 data. *Validation step* adalah jumlah data validasi dibagi dengan nilai *batch size*. Jumlah data validasi yaitu 600 gambar. Kemudian terdapat *number of filter* yang dimasukkan kedalam proses konvolusi nya. Pada tahap konvolusi pertama digunakan jumlah filter sebanyak 32 dan pada konvolusi kedua digunakan jumlah filter sebanyak 64. Kemudian untuk ukuran kernel nya menggunakan dua ukuran kernel yaitu 3x3 dan 5x5. Kernel adalah sebuah matriks untuk menghitung dan mendeteksi suatu pola yang digunakan pada saat proses*convolution. Pooling sizenya* diberikan

nilai 2. *Pooling* adalah Proses Mengurangi dimensi dari *feature map* (*downsampling*). Penelitian ini menggunakan 3 kelas gambar mata. Kemudian *learning rate* nya dengan membandingkan nilai learning rate 0.001 dan 0.0001.

Penelitian ini menggunakan model CNN dengan 2 proses konvolusi dan 2 proses *pooling layer* . pada masing masing proses konvolusi digunakan aktivasi fungsi RELU. Aktivasi fungsi ini bertujuan mengubah nilai minus pada sebuah matriks dari hasil proses konvolusi. Aktivasi RELU melakukan "*treshold*" dari 0 hingga *infinity*. Kemudian pada proses konvolusi digunakan *zero padding. Zero padding* adalah Parameter jumlah piksel yang berisi nilai nol yang ditambahkan disetiap sisi input. Setelah melalui proses konvolusi hasil akhir dari max pooling akan diubah kedalam bentuk *vector* dua dimensi. Layer *dense* dengan jumlah *vector* 2300 dengan menggunakan aktifasi fungsi RELU. Nilai *Dropout* yang digunakan yaitu 0.5 dan menggunakan fungsi aktivasi *sofmax*. Fungsi aktivasi ini bertujuan untuk mengklasifikasi kedalam banyak kelas.Kemudian untuk *loss function* nya menggunakan optimasi Adam. *Tensorboard* sebuah *visualize tools* yang disediakan oleh *tensorflow*, dengan bantuan *tools* ini dapat mempermudah dalam melihat tingkat *accuracy* dan loss model dari data *train* dan *validation.* Model fit generator digunakan untuk membangkitkan data untuk setiap *epoch* sampaisemua *epoch* memenuhi jumlah sampelnya. *Save* model dalam sebuah directory dan gunakan model ini untuk mengklasifikasi dengan data baru.

3.3 Pengujian Model

Algoritma *Convolutional Neural Network* membutuhkan proses training dan *testing*.Proses *training* ini bertujuan untuk melatih algoritma CNN dalam mengenali datasetnya dan membentuk sebuah model berdasarkan pelatihan tersebut. Proses *testing* betujuan mengujisebuah model yang dibentuk pada saat proses *training*.

```
6   img_width, img_height = 64, 64
7   model_path = './models/model(epoch=50,lr=0.001,Op=adam)/model.h5'
8   model_weights_path = './models/model(epoch=50,lr=0.001,Op=adam)/weights.h5'
9   model = load_model(model_path)
10  model.load_weights(model_weights_path)
```

Gambar 3. Proses *Training*

Input image harus disamakan dengan input pada proses *train*. Jika tidak sama maka akan terjadi kesalahan pada program dan model tidak akan membaca gambar tersebut.

3.4 Hasil Penelitian

Proses utama dalam pembuatan model ini diawali dengan proses *training data*. Prosesini bertujuan untuk pembentukan model yang akan digunakan untukpengujian data *testing.*

Parameter untuk mengukur tingkat keberhasilan model adalah nilai akurasi. Nilai akurasi model dapat ditentukan dengan melakukan pengujian menggunakan data *testing*. Proses *training* menggunakan *packages Keras* pada *python* dengan *back-end tensorflow.* Penelitian ini menggunakan input gambar dengan ukuran 64x64x3, tujuannya adalah untuk membandingkan nilai akurasi berdasarkan ukuran gambarnya. Proses Konvolusi pertama digunakan kernel berukuran 3x3 dan jumlah filter sebanyak 32 filter , proses konvolusi ini adalah proses kombinasi antara dua buah matriks yang berbeda untuk menghasilkan suatu nilai matriks yang baru. Setelahproses konvolusi, maka ditambahkan sebuah aktivasi fungsi yaitu RELU (*Retrified Linear Unit*).

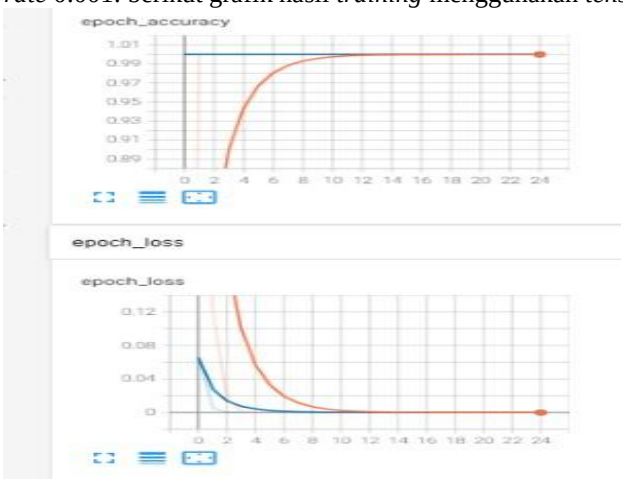
Fungsi aktivasi ini bertujuan untuk mengubah nilai negative menjadi nol(menghilangkan nilai

7*negative* dalam sebuah matriks hasil konvolusi). Hasil konvolusi ini memiliki ukuran yang sama yakni 64x64 karena pada saat proses konvolusi digunakan nilai *padding* 0.

1. Proses *pooling*, *pooling* merupakan proses mereduksi ukuran sebuah data citra dimana adanya proses pengurangan ukuran matriks dengan menggunakan operasi *pooling* [4]. Pada dasarnya pooling layer terdiri dari sebuah filter denganukuran dan tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Penelitian ini menggunakan *maxpooling* untuk mendapatkan nilai matriks yang baru hasil dari proses *pooling*. Berdasarkan hasil pooling menghasilkan matriks baru berukuran 32x32 dengan menggunakan kernel *pooling* 2x2. Cara kerja *max-pooling* adalah mengambil nilai paling maksimum berdasarkan pergeseran kernelnya sebnayak nilai *stride*nya yaitu 2.
2. Proses Kovolusi kedua yaitu meneruskan hasil dari proses *pooling* pertama yakni dengan input matriks gambar sebesar 32x32 dengan jumlah filter sebanyak 64 filter dan dengan ukuran kernel 3x3. Proses konvolsi kedua ini sama-sama menggunakan fungsi aktivasi RELU.
3. Proses selanjutnya masuk ke proses *pooling* yang kedua, proses ini hampir sama dengan proses *pooling* hang pertama, namun ada perbedaan pada nilai *output* akhir dari matriksnya. *Output* yang dihasilkan memiliki ukuran gambar 16x16.
4. Proses *Flatten* atau *fully connected*. *Flatten* adalah proses membentuk ulang fitur (reshape feature map) menjadi sebuah vektor agar bisa digunakan sebagai input dari fully – connected layer [5]. Pada tahap ini digunakan hanya satu *hidden layer* pada jaringan MLP (*Multi Layer Perceptron*). *Flatten* disini mengubah *output pooling* layer menjadi sebuah vector. Sebelum melakukan proses klasifikasi atau memprediksi gambar, pada proses ini digunakan nilai *Dropout*. *Dropout* adalah sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain neuron-neuron tersebut dibuang secara acak. Tujuan dariproses ini yaitu mengurangi *overfitting* pada saat proses *training.*
5. Proses terakhir adalah menggunakan aktivasi fungsi *Softmax*. Fungsi ini secara sepsifiknya fungsi ini biasa digunakan pada metode klasifikasi *multinomial logistic regression* dan *multiclass linear discriminant analisys.*

3.5 Model Hasil Training

Setelah melalu beberapa proses dalam algoritma *Convolutional Neural Network* (CNN) didapatkan hasil *training dan validation*. Proses ini menggunakan jumlah 20 *epoch*, nilai *learning rate* 0.001. berikut grafik hasil *training* menggunakan *tensorboard* :



Gambar 4. Hasil *Training*

3.6 Epoch

Epoch adalah ketika seluruh *dataset* sudah melalui proses *training* pada *Neural Network* sampai dikembalikan ke awal dalam satu putaran. Dalam *Neural Network* satu*epoch* itu terlalu besar dalam proses pelatihan karena seluruh data diikutkan kedalam proses training sehingga akan membutuhkan waktu cukup lama. Untuk mempermudahdan mempercepta proses *training* biasanya, data sate dibagi per *batch* (*Batch Size*). Penentuan nilai dari *batch size* biasanya tergantung peneliti dengan melihat banyaksampel. Berikut adalah hasil perbandingan epoch dari hasil *training*.

Tabel 1. *Accuracy Based*

Epoch	***Lost Validation***	***Time (Seconds)***
3/25	*0.1235*	*141*
4/25	*0.0169*	*109*
5/25	*0.0021*	*234*
1/1	*2.8544*	*172*

4. KESIMPULAN

Berdasarkan hasil analisis yang telah dilakukan, diperoleh beberapa kesimpulan yaitu :

1. Hasil dari grafik *tenserboard* ialah jumlah *epcoh* sebanyak 20 *epoch*, nilai *learning rate* 0.001 sehingga model CNN pada penelitian ini menggunakan input shape berukuran 64x64. Menghasilkan tingkat akurasi training dan testing dalam melakukan klasifikasi gambar citra.
2. Skenario penggunaan nilai epoch didapatkan tingkat akurasi terbaik menggunakan nilai epoch sebesar 100, dengan akurasi 92 %.