

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/363924413>

# Klasifikasi Cats dan Dogs

Article · September 2022

---

CITATIONS

0

---

READS

17

2 authors, including:



[La Ode Ansyarullah S. Sagala](#)

Bandung Institute of Technology

1 PUBLICATION 0 CITATIONS

SEE PROFILE

# Klasifikasi *Cats* dan *Dogs* dengan Metode CNN dalam Fungsi Aktivasi *relu*, *sigmoid*, *softmax*, *softplus*, *softsign*, dan *selu*

La Ode Ansyarullah S. Sagala  
Teknik Elektro  
Institut Teknologi Bandung  
Bandung, Indonesia  
23221066@mahasiswa.itb.ac.id

**Abstract**—Dalam proyek kali ini, tugas kami untuk melakukan klasifikasi sebuah gambar Kucing vs Anjing dengan menggunakan metode *Convolutional Neural Network* (CNN). Metode CNN mampu melakukan keseluruhan skala informasi klasifikasi pada objek tanpa kehilangan keakuratannya. Tujuan dari tugas ini adalah mampu memberikan sebuah komputer kemampuan untuk mengenali jenis binatang, kemudian mampu memprediksi jenis binatang berdasarkan gambar yang dimasukkan dan mengetahui hasil saat menambahkan layar fungsi aktivasi pada algoritma CNN.

**Index Terms**—CNN, Layer Activation Function

## I. PENDAHULUAN

Pada era teknologi yang maju saat ini, dimana teknologi dalam pengenalan sebuah citra menjadi salah satu inputan yang banyak diterapkan pada berbagai bidang. Salah satu potret fungsi dari sebuah fitur pengenalan citra adalah untuk mengenali dan memprediksi jenis sebuah objek dalam citra.

*Artificial Intelligence* (AI) merupakan sebuah bidang keilmuan yang dapat membuat komputer menirukan kebiasaan manusia. Dapat diartikan pula sebagai bagian dari ilmu komputer yang berfokus pada mesin dengan kemampuan kecerdasan yang dapat berinteraksi dan/atau bekerja seperti manusia [1].

Pada bidang *image processing* terdapat beberapa algoritma yang dapat digunakan diantaranya adalah *Neural Network*. *Neural Network* yang dikembangkan berdasarkan cara kerja jaringan saraf pada otak manusia. Salah satu pengembangan *deep learning* adalah *Convolutional Neural Network*.

Metode *deep learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan sebuah citra adalah *Convolutional Neural Network* (CNN) [2]. Hal tersebut dikarenakan CNN berusaha meniru sebuah sistem pengenalan pada citra pada visual cortex manusia [3] sehingga memiliki kemampuan untuk mengelola informasi pada citra. Namun CNN memiliki kelemahan yaitu proses pelatihan model yang lama, tetapi dengan perkembangan perangkat keras hal tersebut dapat diatasi dengan teknologi GPU (*Graphical Processing Unit*).

Pengujian terhadap implementasi metode CNN ini diharapkan dapat membedakan *cats fake* dengan kucing asli serta *dogs fake* dengan anjing asli. Hasil kajian ini diharapkan

bermanfaat sebagai pendekatan untuk menentukan objek. Sehingga diperoleh efisiensi dalam melakukan klasifikasi citra.

## II. LANDASAN TEORI

### A. Convolutional Neural Network

*Convolutional Neural Network* (CNN) termasuk dalam jenis *deep learning* dikarenakan kedalaman terhadap jaringannya. *Deep Learning* adalah cabang dari *machine learning* yang dapat mengajarkan komputer untuk melakukan pekerjaan yang selayaknya manusia, seperti komputer dapat belajar dari proses training. CNN merupakan operasi konvolusi yang menggabungkan beberapa lapisan pemrosesan dengan menggunakan beberapa elemen yang beroperasi secara paralel dan terinspirasi sistem saraf biologis [4].

### B. Konsep CNN

Cara kerja CNN memiliki kesamaan dengan MLP, tetapi dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.

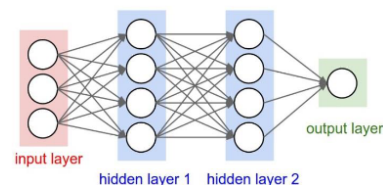


Fig. 1. Arsitektur MLP Sederhana

Seperti pada Gambar 1 memiliki  $i$  layer (kotak merah dan biru) dimana masing-masing layer berisi  $j_i$  neuron (lingkaran putih). MLP menerima sebuah input data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan output. Setiap hubungan antar neuron pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode.

Pada CNN data yang digunakan untuk propagasikan pada jaringan yaitu data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi

satu dimensi lagi, namun berbentuk empat dimensi yang merupakan kumpulan-kumpulan kernel konvolusi seperti pada Gambar 2. Dimensi bobot untuk CNN adalah:

$$\text{neuron input} * \text{neuron output} * \text{tinggi} * \text{lebar}$$

dikarenakan sifat suatu proses konvolusi maka CNN hanya dapat digunakan pada sebuah data yang memiliki sebuah struktur dua dimensi seperti citra dan suara.

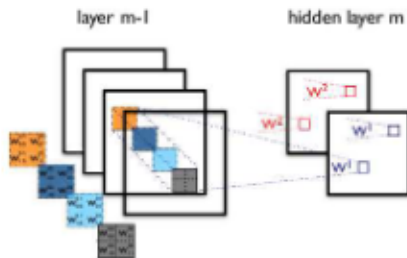


Fig. 2. Proses Konvolusi pada CNN

### C. Arsitektur Jaringan CNN

*Convolutional Neural Network* memiliki *input layer*, *output layer*, dan *hidden layer*. *Hidden layer* terdiri dari *convolutional layer*, *flattened layer*, dan *fully connected layer*. Seperti yang terlihat pada Gambar 3. menunjukkan arsitektur CNN.

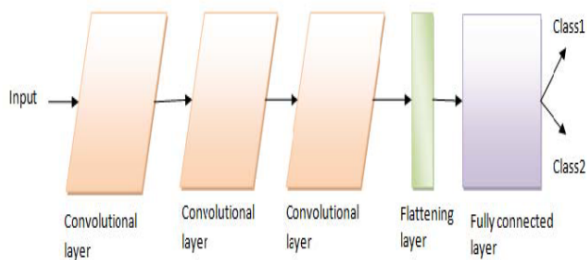


Fig. 3. Convolutional Neural Network

- 1) *Convolutional Layer* merupakan sebuah blok bangunan CNN, dimana tugas utama *convolutional layer* adalah mengekstrak sebuah fitur. Ini terdiri dari satu atau lebih lapisan konvolusi yang diikuti oleh satu atau lebih lapisan yang terhubung. Konvolusi akan menghasilkan sebuah transformasi *linear* dari data input sesuai informasi spasial pada data.
- 2) *Pooling Layer* Pooling atau subsampling adalah pengurangan ukuran matriks. Terdapat dua macam pooling yang sering digunakan yaitu average pooling dan max pooling [5]. Nilai yang diambil pada average pooling adalah nilai rata-rata sedangkan pada max pooling adalah nilai maksimal [6], seperti yang terlihat pada Gambar 5.
- 3) *Flattening Layer* merupakan sebuah *layer* yang digunakan dalam mengubahnya menjadi tensor satu dimensi. Ini dilakukan agar ketika menggunakan operasi pembentukan ulang dari kerangka kerja tensorflow, untuk

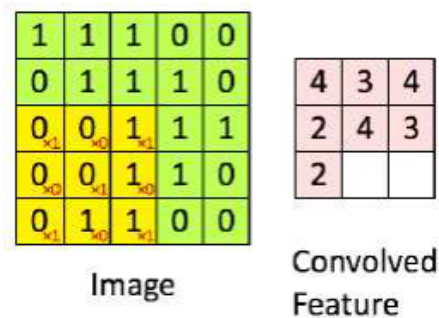


Fig. 4. Operasi Konvolusi

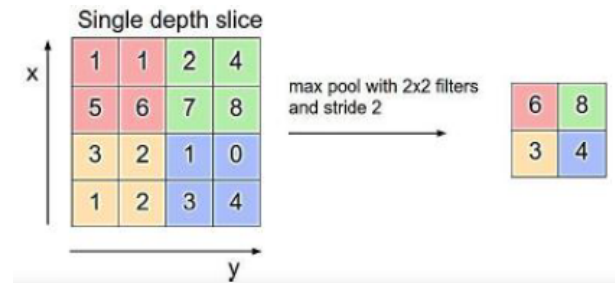


Fig. 5. Operasi Max Pooling

mendapatkan *output* dari lapisan konvolusi sebelumnya dan meratakan strukturnya dapat membuat vektor fitur tunggal dapat digunakan oleh lapisan yang terhubung penuh untuk melakukan klasifikasi.

- 4) *Fully Connected Layer* Lapisan fully connected layer merupakan kumpulan dari proses konvolusi [7]. Lapisan ini mendapatkan input dari proses sebelumnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari lapisan ini adalah untuk menyatukan semua node menjadi satu dimensi [8].

### D. Fungsi Aktivasi

Fungsi aktivasi pada *neural network* merupakan persamaan matematika yang menentukan sebuah *output* dari *neural network*. Fungsi aktivasi tersebut akan diimplementasikan pada setiap *neuron* dalam sebuah jaringan, dan dapat menentukan apakah *neuron* tersebut harus diaktifkan atau tidak berdasarkan relevansi setiap *input neuron* untuk memprediksi sebuah model.

Fungsi aktivasi harus efisien secara komputasi dikarenakan harus menghitung banyak *neuron* untuk setiap sampel data. Fungsi aktivasi juga dapat membantu proses normalisasi *output* pada setiap *neuron* untuk memiliki nilai dalam *range* antara 1 dan 0, atau antara -1 dan 1. Terdapat berbagai jenis fungsi aktivasi, sebagai berikut.

- 1) ReLU memiliki kepanjangan, yaitu *Rectified Linear Unit* dimana memiliki keuntungan terhadap fungsi aktivasi *sigmoid* dan *tanh* karena tidak pernah menjadi *saturated* (jenuh) terhadap nilai *x* yang besar. Dimana rata-rata dari fungsi ini tidak bernilai nol.

- 2) Sigmoid Fungsi aktivasi *sigmoid* akan mengembalikan *output value* dengan *range* antara 0 hingga 1. Fungsi aktivasi *sigmoid* akan menghasilkan kurva dengan bentuk "S".

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

- 3) Softmax memiliki keuntungan utama adalah rentang probabilitas *output* dengan nilai 0 hingga 1, dan jumlah semua probabilitasnya akan sama dengan satu. Fungsi *softmax* digunakan untuk sebuah model multi-klasifikasi yang akan mengembalikan sebuah peluang dari masing-masing kelas dan kelas *target* akan memiliki probabilitas yang tinggi.
- 4) Softplus memiliki kemiripan dengan *ReLU* bahwa mendekati nilai 0 *softplus* akan lebih halus dan *differentiable*. Fungsi aktivasi ini akan menghasilkan sebuah nilai *output* dengan *range* nilai antara 0 hingga positif tidak terbatas.
- 5) Softsign sebuah fungsi dari *nonlinearity* yang dapat dianggap sebagai alternatif dari fungsi aktivasi *tanh* karena tidak mudah *saturated* (jenuh).
- 6) Selu atau biasa dikenal dengan *Scaled Exponential Linear Units* merupakan sebuah fungsi aktivasi yang menginduksi normalisasi diri.

$$f(x) = \lambda x \quad (2)$$

$$f(x) = \lambda \alpha (e^x - 1) \quad (3)$$

### III. METODOLOGI

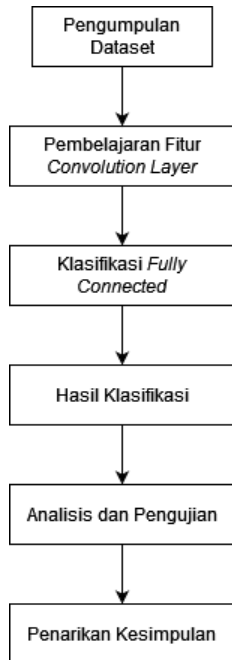


Fig. 6. Diagram Alur

### A. Dataset

Dataset yang digunakan pada tugas kali ini merupakan dataset *cats* dan *dogs*. Untuk dataset *cats* berjumlah 400 citra dan untuk *dogs* berjumlah 400 citra. Seluruh citra dataset *cats* dan *dogs* memiliki ukuran 150x150 piksel dengan total jumlah citra sebesar 800 citra.

### B. Convolutional Neural Network

CNN adalah salah satu jenis *neural network* yang ada pada *deep learning*. Berbeda dengan ANN dan RNN, CNN biasanya digunakan untuk mengolah data dalam bentuk citra. CNN menggunakan *kernel* dimana akan mengekstrak fitur dari input menggunakan operasi konvolusi.

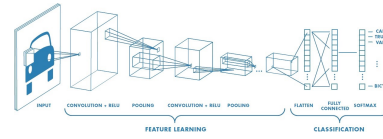


Fig. 7. Arsitektur CNN

### C. Instrumentasi

Instrumen yang digunakan untuk tugas ini adalah sebagai berikut :

#### 1) Perangkat Lunak

Visual Studi Code (VS Code) merupakan sebuah software yang sangat ringan. VS Code dengan dukungan untuk *JavaScript*, naskah dan memiliki *array* beragam ekstensi yang tersedia untuk bahasa lain termasuk C, C++, *Python*, dan PHP.

#### 2) Perangkat Keras

Laptop Acer dengan spesifikasi : Intel Core i5-10210U quad-core 1,6GHz TurboBoost 4,2GHz.

### IV. HASIL DAN PEMBAHASAN

Untuk Pengklasifikasi dengan model CNN dimana untuk tugas kali ini dapat ditampilkan sebuah fungsi *summary()* untuk bisa melihat detail arsitektur. Arsitektur ini hampir sama digunakan untuk semua fungsi aktivasi. Seperti pada Gambar 8.

#### A. ReLU

Setelah melakukan proses pelatihan dan validasi dengan fungsi aktivasi *relu*, dengan menggunakan 50 *epochs* diperoleh grafik. Seperti pada Gambar 9 dan Gambar 10.

#### B. Sigmoid

Setelah melakukan proses pelatihan dan validasi dengan fungsi aktivasi *sigmoid*, dengan menggunakan 50 *epochs* diperoleh grafik. Seperti pada Gambar 11 dan Gambar 12.

#### C. Softmax

Setelah melakukan proses pelatihan dan validasi dengan fungsi aktivasi *softmax*, dengan menggunakan 50 *epochs* diperoleh grafik. Seperti pada Gambar 13 dan Gambar 14.

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_4 (MaxPooling 2D)	(None, 74, 74, 32)	0
conv2d_5 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_5 (MaxPooling 2D)	(None, 36, 36, 64)	0
conv2d_6 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_6 (MaxPooling 2D)	(None, 17, 17, 128)	0
conv2d_7 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_7 (MaxPooling 2D)	(None, 7, 7, 128)	0
flatten_1 (Flatten)	(None, 6272)	0
...		
Total params: 3,453,121		
Trainable params: 3,453,121		
Non-trainable params: 0		

Fig. 8. Model CNN

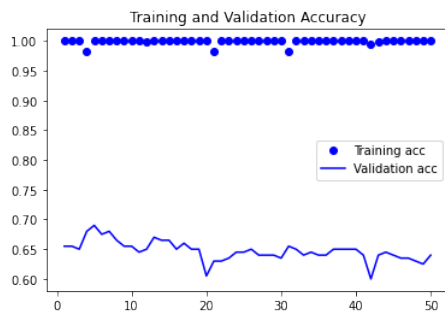


Fig. 9. Grafik acc data training dan validasi

#### D. Softplus

Setelah melakukan proses pelatihan dan validasi dengan fungsi aktivasi *softplus*, dengan menggunakan 50 *epochs* diperoleh grafik. Seperti pada Gambar 15 dan Gambar 16.

#### E. Softsign

Setelah melakukan proses pelatihan dan validasi dengan fungsi aktivasi *softsign*, dengan menggunakan 50 *epochs* diperoleh grafik. Seperti pada Gambar 17 dan Gambar 18.

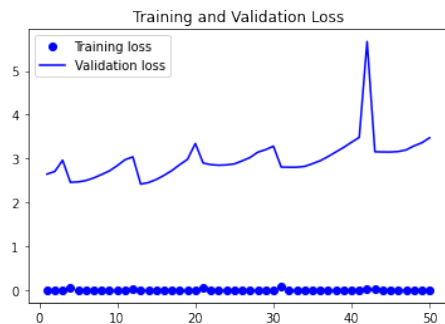


Fig. 10. Grafik loss data training dan validasi

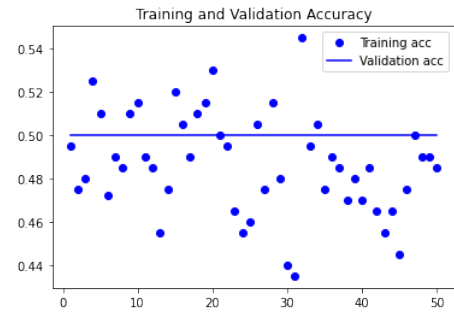


Fig. 11. Grafik acc data training dan validasi

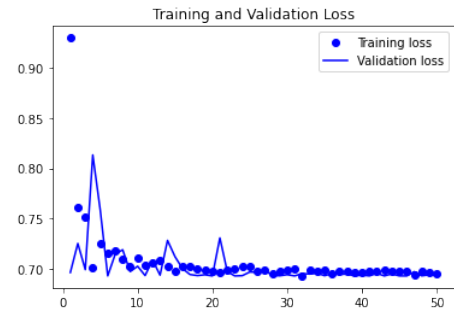


Fig. 12. Grafik loss data training dan validasi

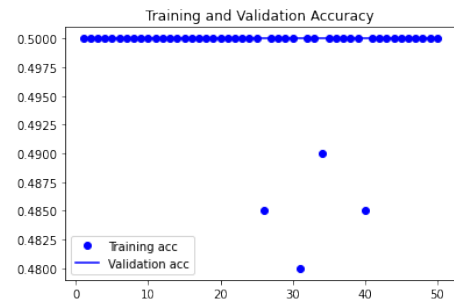


Fig. 13. Grafik acc data training dan validasi

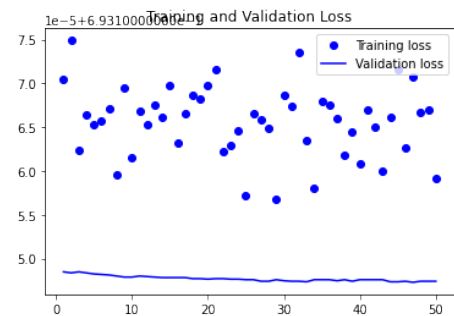


Fig. 14. Grafik loss data training dan validasi

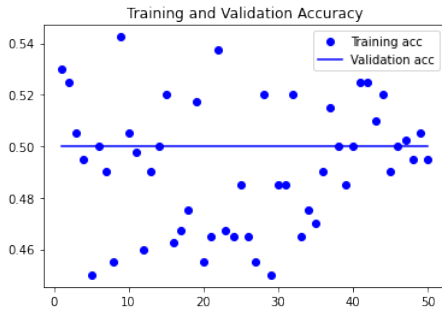


Fig. 15. Grafik acc data training dan validasi

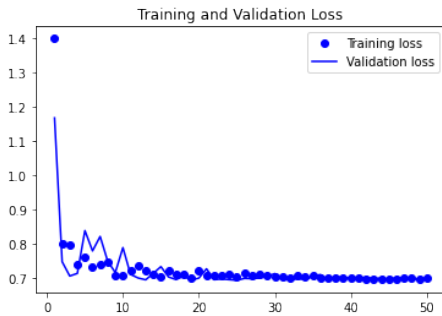


Fig. 16. Grafik loss data training dan validasi

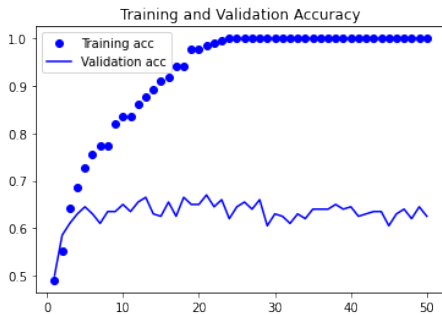


Fig. 17. Grafik acc data training dan validasi

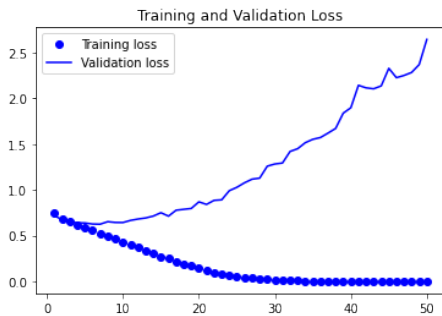


Fig. 18. Grafik loss data training dan validasi

## F. Selu

Setelah melakukan proses pelatihan dan validasi dengan fungsi aktivasi *selu*, dengan menggunakan 50 *epochs* diperoleh grafik. Seperti pada Gambar 19 dan Gambar 20.

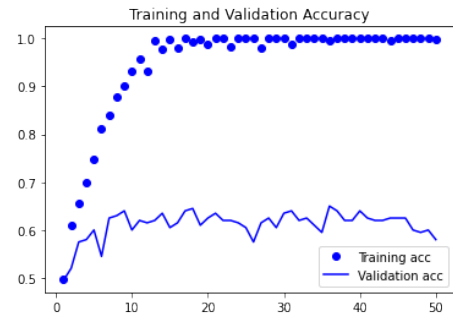


Fig. 19. Grafik acc data training dan validasi

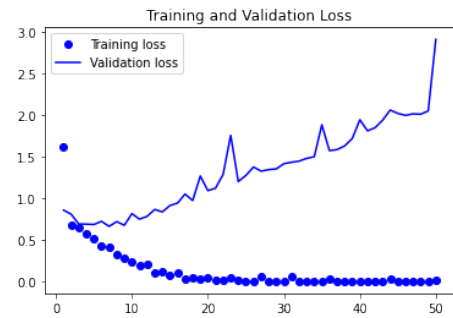


Fig. 20. Grafik loss data training dan validasi

## G. Pembahasan

Pada proses pelatihan kata "*loss*" pada gambar yang didapatkan merupakan besarnya kerugian saat melakukan pelatihan dalam setiap *epoch*. Nilai tersebut akan diminimalkan dengan fungsi optimasi RMSProp, dimana semakin kecil nilai *loss*, maka semakin baik pula hasil yang didapatkan. Sedangkan kata "*accuracy*" mengacu pada rasio antara prediksi yang benar dengan total prediksi pada saat proses pelatihan. Dimana semakin tinggi nilai akurasi, maka akan semakin baik. Begitu pula dengan *val\_loss* dan *val\_accuracy*, mempunyai konsep yang sama tetapi data yang diproses merupakan data dari validasi.

Apabila saat memproses sebuah algoritma komputasi nilai akurasi data latih dan valid memiliki selisih yang besar, hal ini dikarenakan model mengalami *overfitting*, sehingga model tidak dapat digunakan dalam memprediksi sebuah data citra. Dari hasil eksperimen yang dilakukan didapatkan sebuah grafik akurasi data training dan validasi pada tiap-tiap fungsi aktivasi.

Untuk fungsi aktivasi *relu* data latih menghasilkan akurasi sebesar 1.0 dan pada data validasi 0.63. Untuk fungsi aktivasi *sigmoid* data latih menghasilkan akurasi sebesar 0.5 dan pada data validasi 0.5.

Untuk fungsi aktivasi *softmax* data latih menghasilkan akurasi sebesar 0.5 dan pada data validasi 0.5. Untuk fungsi aktivasi *softplus* data latih menghasilkan akurasi sebesar 0.5 dan pada data validasi 0.5.

Untuk fungsi aktivasi *softsign* data latih menghasilkan akurasi sebesar 1.0 dan pada data validasi 0.625. Untuk fungsi aktivasi *selu* data latih menghasilkan akurasi sebesar 0.89 dan pada data validasi 0.058.

## V. KESIMPULAN

Hasil dari implementasi metode CNN ini cukup baik saat melakukan proses sebuah algoritma. Namun untuk penggunaan berbagai fungsi aktivasi mendapatkan hasil yang cukup baik sesuai hasil plotting grafik yang didapatkan. Diketahui juga bahwa jumlah sebuah data training dapat mempengaruhi akurasi jaringan. Semakin banyak data training maka jaringan akan semakin banyak belajar sehingga ketelitian akan semakin baik.

## REFERENCES

- [1] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020, doi: 10.26740/jinacs.v1n02.p104-108.
- [2] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.
- [3] K. Zakka, "CS231n Convolutional Neural Networks for Visual Recognition," Stanford Univ., pp. 1–23, 2021, [Online]. Available: <https://cs231n.github.io/convolutional-network/>.
- [4] F. Hu, G. S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015, doi: 10.3390/rs71114680.
- [5] Bejiga, M., Zeggada, A., Nouffidj, A., & Melgani, F. (2017). A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery. *Remote Sensing*, 9(2), 100. doi:10.3390/rs9020100
- [6] Zhi, T., Duan, L.-Y., Wang, Y., & Huang, T. (2016). Two-stage pooling of deep convolutional features for image retrieval. 2016 IEEE International Conference on Image Processing (ICIP). doi:10.1109/icip.2016.7532802
- [7] Hijazi, S., Kumar, R., & Rowen, C. (2015). Image Recognition Using Convolutional Neural Networks. *Cadence Whitepaper*, 1–12
- [8] Albelwi, S., & Mahmood, A. (2017). A Framework for Designing the Architectures of Deep Convolutional Neural Networks. *Entropy*, 19(6), 242. doi:10.3390/e19060242