

KOMPARASI FUNGSI AKTIVASI NEURAL NETWORK PADA DATA TIME SERIES

Ibnu Akil
Sistem Informasi
Universitas Bina Sarana Informatika
www.bsi.ac.id
ibnu.ial@bsi.ac.id



Abstract— *The sophistication and success of machine learning in solving problems in various fields of artificial intelligence cannot be separated from the neural networks that form the basis of its algorithms. Meanwhile, the essence of a neural network lies in its activation function. However because so many activation function which are merged lately, it's needed to search for proper activation function according to the model and it's dataset used. In this study, the activation functions commonly used in machine learning models will be tested, namely; ReLU, GELU and SELU, for time series data in the form of stock prices. These activation functions are implemented in python and use the TensorFlow library, as well as a model developed based on the Convolutional Neural Network (CNN). From the results of this implementation, the results obtained with the CNN model, that the GELU activation function for time series data has the smallest loss value.*

Keyword: *activation function, machine learning, neural network.*

Abstrak—Kecanggihan dan kesuksesan *machine learning* dalam memecahkan masalah pada berbagai bidang *artificial intelligence* tidak terlepas dari *neural network* yang menjadi dasar algoritmanya. Sedangkan inti dari *neural network* terletak pada *activation function* (fungsi aktivasi) yang ada padanya. Namun demikian karena banyaknya berkembang fungsi aktivasi, maka perlu diteliti fungsi aktivasi yang sesuai dengan model dan dataset yang digunakan. Pada penelitian ini akan diuji fungsi aktivasi yang umum digunakan dalam model machine learning, yaitu; ReLU, GELU dan SELU, terhadap data time series berupa stock price. Fungsi-fungsi aktivasi tersebut diimplementasikan dalam bahasa python dan menggunakan library TensorFlow, serta model yang dikembangkan berbasis Convolutional Neural Network (CNN). Dari hasil implementasi tersebut didapatkan hasil dengan model CNN, bahwa fungsi aktivasi GELU untuk data time series memiliki nilai loss terkecil yait sebesar 0.0007120146765373647.

Kata kunci: fungsi aktivasi, neural network, pembelajaran mesin.

PENDAHULUAN

Deep learning atau *deep neural network* telah sukses digunakan di dalam berbagai macam bidang yang ada sekarang ini untuk menyelesaikan masalah secara nyata yang mungkin menggunakan arsitektur *deep learning* yang semakin komplek. Untuk mencapai performa terbaik dari hal ini, arsitektur *deep learning* menggunakan fungsi aktivasi (*activation function*), untuk menjalankan beragam komputasi antara *hidden layers* dan *output layers* dari berbagai arsitektur *deep learning*. Jadi, inti dari neural network yang diterapkan dalam *machine learning* yang sederhana atau yang lebih kompleks yaitu *deep learning*, adalah fungsi aktivasi. Akan tetapi dengan menjadikan deep learning sebagai arus utama dalam penelitian,

berbagai fungsi aktivasi mulai menjamur, yang menyebabkan kebingungan baik ditataran teori maupun praktek (Lederer, 2021).

Tujuan dari penelitian ini adalah menentukan fungsi aktivasi yang terbaik untuk data time series. Pemilihan fungsi aktivasi di dalam hidden layer akan mengendalikan seberapa baik jaringan belajar dari data training. Selain itu pemilihan fungsi aktivasi yang tepat di dalam output layer akan menentukan jenis prediksi yang dapat dibuat oleh model (Brownlee, 2021). Penelitian tentang fungsi aktivasi yang umum telah banyak dilakukan diantaranya oleh Tomasz Szandata (Szandata, 2018). Selain itu juga Nwankpa telah melakukan survey pada fungsi-fungsi aktivasi yang ada yang digunakan di dalam aplikasi-aplikasi deep learning dan dalam kompetisi

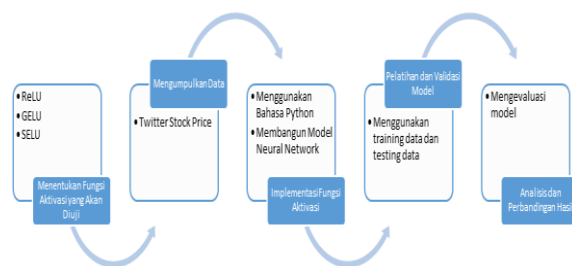
ILSVRC, yaitu diterapkan dalam berbagai kasus image classification, NLP, object detection, dan lain-lain (Nwankpa et al., 2018).

Permasalahan dari penelitian sebelumnya, Szandata melakukan perbandingan fungsi aktivasi yang umum dipakai seperti sigmoid, tanh, ReLU, dan lain-lain, masih sebatas dalam tataran teori. Sedangkan Nwankpa hanya melakukan review penerapan masing-masing fungsi yang dilakukan oleh para peneliti dan menemukan bahwa fungsi ELU sebagai fungsi aktivasi tercepat dibanding ReLU.

Pada penelitian ini akan dicoba menerapkan fungsi-fungsi aktivasi, namun untuk membatasi ruang lingkup fungsi-fungsi aktivasi yang akan diuji coba antara lain; Rectified Linear Unit (ReLU), Gaussian Error Linear Unit (GELU), dan Scaled Exponential Linear Unit (SELU), yang akan diterapkan pada data time series Twitter Stock Price. Diharapkan dengan penggunaan data real dalam ujicoba akan menunjukkan komparasi yang lebih komprehensif.

BAHAN DAN METODE

Tahapan dalam penelitian ini disusun sebagai berikut: menentukan fungsi aktivasi yang akan diuji coba yaitu: ReLU, GELU, dan SELU; mengumpulkan data yang akan digunakan dalam implementasi; implementasi fungsi aktivasi dalam model neural network; pelatihan dan validasi model; analisis dan perbandingan hasil; dan kesimpulan, sebagaimana tampak pada gambar 1 berikut:



Sumber: (Akil, 2023)

Gambar 1. Rancangan Penelitian

Data Penelitian:

Sumber data utama yang digunakan dalam penelitian ini adalah data yang bersifat time series. Data ini peneliti dapatkan dari dataset DJIA Stock Prices 2006-2017 yang didapatkan dari <https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231>, dan data yang dipilih adalah data milik Google. Dataset ini memiliki fitur time series atau range, pada dataset di tabel 1 adalah kolom "Date". Dalam penelitian ini Dataset tersebut akan dibagi menjadi tiga, 70% untuk data training dan 20% untuk validasi, dan 10% untuk test. Dataset ini berisi 2118 record. Fitur-fitur dari dataset tersebut dapat dilihat pada tabel 1 berikut ini:

<https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231>, dan data yang dipilih adalah data milik Google. Dataset ini memiliki fitur time series atau range, pada dataset di tabel 1 adalah kolom "Date". Dalam penelitian ini Dataset tersebut akan dibagi menjadi tiga, 70% untuk data training dan 20% untuk validasi, dan 10% untuk test. Dataset ini berisi 2118 record. Fitur-fitur dari dataset tersebut dapat dilihat pada tabel 1 berikut ini:

Tabel 1. Fitur Data Penelitian

	Date	Open	High	Low	Close	Volume	Name
0	2006-01-03	211.47	218.05	209.32	217.83	13137450	GOOGL
1	2006-01-04	222.17	224.70	220.09	222.84	15292353	GOOGL
2	2006-01-05	223.22	226.00	220.97	225.85	10815661	GOOGL
3	2006-01-06	228.66	235.49	226.85	233.06	17759521	GOOGL
4	2006-01-09	233.44	236.94	230.70	233.68	12795837	GOOGL

Sumber: (Akil, 2023)

Date—adalah tanggal transaksi, **Open**—adalah nilai saham diawal pembukaan, **High**—adalah nilai tertinggi pada tanggal tersebut, **Low**—adalah nilai terendah pada tanggal tersebut, **Close**—adalah nilai pada saat penutupan di tanggal tersebut, **Volume**—yang merupakan jumlah saham yang diperjual-belikan pada tanggal tersebut (Akil, 2023). Sedangkan **Name** adalah perusahaan pemilik data tersebut, yang nantinya fitur ini akan dihapus karena tidak terpakai dalam pemodelan.

Metode:

Untuk proses pemodelannya diimplementasikan pada bahasa Python dan library Tensorflow. Saat ini bahasa python dianggap pilihan terbaik untuk implementasi pembelajaran mesin salah satunya karena banyak memiliki library seperti scikit-learn, keras, Tensorflow, dan lain-lain (Ryabtsev, 2022). Begitu juga library TensorFlow yang mudah dalam membangun model dan sangat handal untuk riset experimental.

Activation Function:

Menurut LeCun, fungsi aktivasi adalah sebuah fungsi non linear yang digunakan untuk melakukan training terhadap neural network, dengan mendefinisikan output dari neuron dari satu set input yang diberikan (Gustinel, 2022). Cara kerja fungsi aktivasi di dalam neural network adalah; input diberikan kepada network dari input layer. Di dalam neuron, di layer berikutnya, jumlah bobot dari input-input dikalkulasi, dan biasa ditambahkan kepada jumlah tersebut. Jumlah ini kemudian melalui fungsi aktivasi (Shaw, 2022).

Rectified Linear Unit (ReLU):

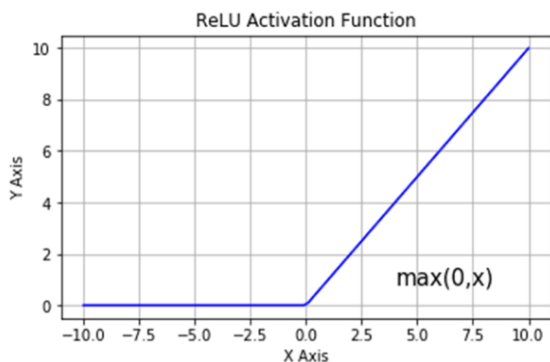
Fungsi aktivasi yang cukup terkenal adalah ReLU. ReLU diperkenalkan oleh Kunihiko Fukushima pada tahun 1969 dalam konteks ekstraksi fitur visual neural network bertingkat. Dewasa ini ReLU adalah fungsi aktivasi yang paling banyak digunakan untuk deep Convolutional Neural Network (CNN)(Varshney & Singh, 2021). Ditemukan bahwa ReLU membuat training deep neural network lebih baik. ReLU didefinisikan sebagai fungsi aktivasi yang memetakan nilai x jika x > 0, selain itu nilai negatif diubah menjadi 0. Persamaannya adalah sebagai berikut:

$$\begin{aligned}
 f(x) &= x^+ \\
 &= \max(0, x) = \frac{x + |x|}{2} \\
 &= \begin{cases} x, & \text{if } x > 0, \\ 0, & \text{otherwise.} \end{cases} \dots \dots \dots (1)
 \end{aligned}$$

Fungsi derivatif:

$$\begin{aligned}
 f'(x) &= \begin{cases} 0, & \text{for } x < 0 \\ 1, & \text{for } x \geq 0 \end{cases} \dots \dots \dots (2)
 \end{aligned}$$

Plot ReLU:



Sumber: (Pretorius et al., 2019)
Gambar 2. Plot ReLU

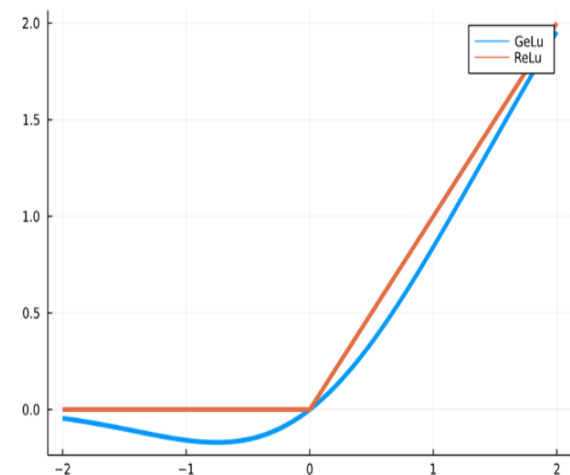
Pada persamaan (1) x adalah input neuron. Untuk fungsi derivatif-nya dapat dilihat pada persamaan (2). Sedangkan untuk plot dari ReLU dapat dilihat pada gambar 2. Dalam kasus dataset CIFAR10, Jaringan ReLU hanya handal daripada jaringan sigmoid ketika dilatih dengan normalisasi batch (Pretorius et al., 2019).

Gaussian Error Linear Unit (GELU):

GELU merupakan kombinasi dari properti dropout, zoneout, dan ReLU. Catatan pertama bahwa ReLU dan dropout, keduanya menghasilkan sebuah output neuron dengan ReLU secara deterministik mengalikan input dengan satu atau nol dan dropout secara stokastik mengalikan dengan nol. Dengan kata lain GELU adalah versi halus dari ReLU. Persamaan GELU dapat dilihat sebagai berikut:

$$\begin{aligned}
 GELU(x) &= xP(X \leq x)x\Phi(x) \\
 &= x \cdot \frac{1}{2} [1 \\
 &+ \text{erf}(\frac{x}{\sqrt{2}})] \dots \dots \dots (3)
 \end{aligned}$$

Plot GELU:



Sumber: (Ureña-Pliego et al., 2023)
Gambar 3. Plot ReLU dan GELU

Perbandingan plot antara ReLU dan GELU dapat dilihat pada gambar 3. Dapat dilihat bahwa persamaan GELU menghasilkan gelombang yang lebih halus dari ReLU.

Scaled Exponential Linear Unit (SELU):

SELU adalah fungsi aktivasi yang memasukkan self-normalization dimana hal ini tidak bisa dilakukan oleh ReLU, sigmoid atau tanh. Fungsi aktivasi SELU secara otomatis memusatkan mean

dan unit varian dengan nol (Ahmed, 2023). Persamaan fungsi matematika SELU dapat dilihat sebagai berikut:

$$f(x) = \lambda x \text{ if } x > 0 \dots \dots \dots (4)$$

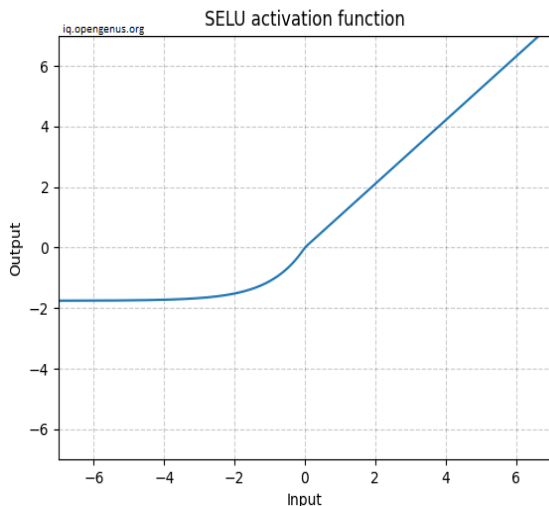
$$f(x) = \lambda \alpha (e^x - 1) \text{ if } x \leq 0 \dots \dots \dots (5)$$

Dimana λ dan α adalah nilai yang mendekati sebagai berikut:

$$\lambda = 1.0507009873554804934193349852946$$

$$\alpha = 1.6732632423543772848170429916717$$

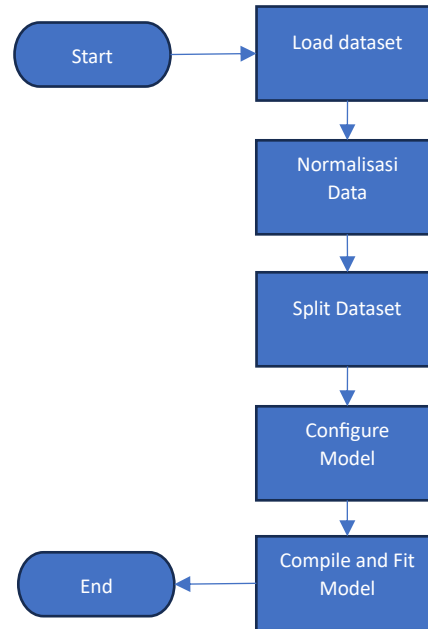
jika nilai x lebih besar dari 0, hasil outputnya adalah x dikali λ . Jika nilai input x adalah lebih kecil atau sama dengan 0, maka kita memiliki fungsi yang naik sampai ke 0, dimana output y, ketika x bernilai 0. Perhatikan gambar 4 yang merupakan plot dari SELU.



Sumber: (Ahmed, 2023)
Gambar 4. Plot SELU

HASIL DAN PEMBAHASAN

Penelitian ini diimplementasikan dengan menggunakan bahasa Python. Python terus menjadi bahasa yang paling disukai untuk komputasi saintifik, data sains, dan penerapan machine learning. Python mempercepat baik itu performa maupun kreatifitas dengan penggunaan library dan high-level API (Raschka et al., 2020). Untuk memudahkan memahami alur program perhatikan Gambar 5 yang merupakan flowchart dari program.



Sumber: (Akil, 2023)
Gambar 5. Flowchart Program

Proses yang terjadi pada flowchart di gambar 5, akan dijelaskan sebagai berikut.

Persiapan Dataset:

Pada tahap ini, dataset di load dari google drive. Kemudian dari dataset yang ada dipilih fitur-fitur data yang akan digunakan untuk dianalisa. Maka kolom "Name" yang ada pada tabel 1, dihapus. Karena kolom tersebut bertipe kategorikal yang tidak diperlukan dalam analisa ini. Dan juga dilakukan pembersihan data dari null value, atau nilai-nilai lain yang menyebabkan distorsi data.

Normalisasi Data:

Selanjutnya dilakukan proses normalisasi data dengan metode Maximum Absolute Scaling. Metode ini merubah skala data berdasarkan nilai maximum yang absolut dari nilai masing-masing fitur (Hackersrealm, 2021). Formulanya adalah sebagai berikut:

$$x_{scaled} = \frac{x}{\max(x)} \dots \dots \dots (6)$$

Normalisasi data diperlukan sebelum data tersebut diproses oleh model. Skala data yang bisa diproses oleh model adalah antara -1

sampai 1. Data yang telah dinormalisasi dapat dilihat pada tabel 2.

Tabel 2. Dataset Normal

Date	Open	High	Low	Close	Volume
2006-01-03	0.1952	0.2006	0.1952	0.2007	0.3190
2006-01-04	0.2051	0.2068	0.2052	0.2053	0.3713
2006-01-05	0.2061	0.2080	0.2060	0.2081	0.2626
2006-01-06	0.2111	0.2167	0.2115	0.2147	0.4312
2006-01-09	0.2155	0.2180	0.2151	0.2153	0.3107

Sumber: (Akil, 2023)

Gambaran data secara keseluruhan dapat dilihat pada gambar 5, dimana fitur yang dipilih adalah "Close".



Sumber: (Akil, 2023)

Gambar 6. Plot Dataset Fitur "Close"

Split Dataset:

Kemudian dataset yang ada dibagi menjadi tiga bagian, yaitu 70% untuk training, 20% untuk validasi, dan 10% untuk test.

Konfigurasi Model dan Layer:

Tahap selanjutnya adalah melakukan konfigurasi model yang meliputi *compiling* dan *fitting*. Jumlah *epoch* (periode training) adalah sebanyak 20. Sedangkan fungsi loss yang digunakan adalah

Mean Squared Error (MSE), dan optimizer yang digunakan adalah "Adam".

Kemudian konfigurasi layer di dalam network dengan menggunakan Convolutional Neural Network, yang terdiri dari tiga layer. Layer pertama adalah Conv1D, layer kedua Dense layer dengan 32 unit output dan fungsi aktivasi yang pertama diujikan adalah "ReLU". Dan terakhir adalah layer Dense dengan 1 unit output.

Compile dan Fitting Model:

Compile dan fitting adalah proses dimana dataset ditraining dengan menggunakan model yang telah dikonfigurasi sebelumnya, dan algoritma neural network melakukan penyesuaian terhadap output atau nilai prediksi sampai didapatkan nilai loss yang minimal. Perulangan dilakukan sesuai jumlah epoch yaitu sebanyak 20.

Dari hasil training dengan menggunakan fungsi aktivasi **ReLU** didapatkan hasil loss sebesar: **0.0007120146765373647**, dan accuracy sebesar: **0.0033444815780967474**.

Selanjutnya adalah konfigurasi model dengan menggunakan fungsi aktivasi "GELU". Dari hasil training tersebut, dengan fungsi aktivasi GELU didapatkan loss sebesar: **0.00013680916163139045**, dan accuracy sebesar: **0.0033444815780967474**.

Selanjutnya konfigurasi model untuk fungsi aktivasi "SELU" dari hasil trainingnya didapatkan hasil dari fungsi aktivasi **SELU**, loss sebesar: **0.0002795507898554206**, dan accuracy sebesar: **0.0033444815780967474**.

Tabel 3. Hasil Perbandingan Fungsi Aktivasi (ReLU, GELU, dan SELU)

Layer	Type	AF	Units	Loss	Accuracy
1	Conv1D	ReLU	32	0.000712	0.003344
2	Dense	ReLU	32		
3	Dense	ReLU	1		
1	Conv1D	GELU	32	0.000137	0.003344
2	Dense	GELU	32		
3	Dense	GELU	1		
1	Conv1D	SELU	32	0.00028	0.003344
2	Dense	SELU	32		
3	Dense	SELU	1		

Sumber: (Akil, 2023)

KESIMPULAN

Dari hasil percobaan yang diuraikan di atas kepada tiga fungsi aktivasi yaitu; ReLU, GELU, dan SELU, dengan menggunakan dataset time series Google Stock Price, didapatkan hasil sebagaimana dapat dilihat pada tabel 3. Disitu dapat dilihat tidak ada perbedaan dari aspek "accuracy" sedangkan loss dengan MSE terdapat perbedaan dimana hasil terkecil didapatkan oleh fungsi aktivasi GELU, sedangkan SELU nomor dua, dan terakhir adalah ReLU. Sebagaimana diketahui semakin kecil loss yang didapatkan, berarti semakin akurat fungsi aktivasi tersebut. Namun kekurangan dari penelitian ini adalah hanya terbatas pada tiga fungsi aktivasi yang diujicobakan. Sebagai saran untuk penelitian selanjutnya, perlu diujicobakan juga fungsi-fungsi aktivasi yang lain dalam kasus yang serupa.

REFERENSI

- Ahmed, G. S. (2023). *SELU (Scaled Exponential Linear Unit) Activation Function*. www.iq.opengenus.org.
<https://iq.opengenus.org/scaled-exponential-linear-unit/>
- Akil, I. (2023). Face Detection Pada Gambar Dengan Menggunakan OpenCV Haar Cascade. *INTI NUSA MANDIRI*, 17(2).
- Brownlee, J. (2021). How to Choose an Activation Function for Deep Learning. In *Machine Learning Mastery* (pp. 1-26).
<https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>
- Gustlineli, M. (2022). *A survey on recently proposed activation functions for Deep Learning*. 1-7.
<http://arxiv.org/abs/2204.02921>
- Hackersrealm. (2021). *Normalize data using Max Absolute & Min Max Scaling | Machine Learning | Python*. www.hackersrealm.net.
<https://www.hackersrealm.net/post/normalize-data-using-max-absolute-min-max-scaling>
- Lederer, J. (2021). *Activation Functions in Artificial Neural Networks: A Systematic Overview*. 1-42.
<http://arxiv.org/abs/2101.09957>
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 1-20.
<http://arxiv.org/abs/1811.03378>
- Pretorius, A. M., Barnard, E., & Davel, M. H. (2019). ReLU and sigmoidal activation functions. *CEUR Workshop Proceedings, 2540*, 37-48.
- Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information (Switzerland)*, 11(4), 1-48.
<https://doi.org/10.3390/info11040193>
- Ryabtsev, A. (2022). *8 Reasons Why Python is Good for AI and ML*. www.jangostars.com.
<https://djangostars.com/blog/why-python-is-good-for-artificial-intelligence-and-machine-learning/#:~:text=Python for machine learning is,and quickly see the results.>
- Shaw, S. (2022). *Activation Functions Compared With Experiments*. www.wandb.ai.
<https://wandb.ai/shweta/Activation-Functions/reports/Activation-Functions-Compared-With-Experiments--VmlldzoxMDQwOTQ>
- Szandała, T. (2018). *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*. Available: https://link.springer.com/chapter/10.1007/978-981-15-5495-7_11.
- Ureña-Pliego, M., Martínez-Marín, R., González-Rodrigo, B., & Marchamalo-Sacristán, M. (2023). Automatic Building Height Estimation: Machine Learning Models for Urban Image Analysis. *Applied Sciences (Switzerland)*, 13(8).
<https://doi.org/10.3390/app13085037>
- Varshney, M., & Singh, P. (2021). Optimizing nonlinear activation function for convolutional neural networks. *SIViP*, 15.