

Ekstraksi Data Digital Menggunakan Teknik *Max Pooling* dan *Average Pooling*

Puspa Meliuwati, Eti Kurniati*

Prodi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Bandung, Indonesia.

ARTICLE INFO

Article history :

Received : 13/8/2022
Revised : 15/12/2022
Published : 21/12/2022



Creative Commons Attribution-
NonCommercial-ShareAlike 4.0
International License.

Volume : 2
No. : 2
Halaman : 137-144
Terbitan : **Desember 2022**

ABSTRAK

Pengolahan data digital membutuhkan algoritma yang dapat memproses data secara optimal. Pemrosesan data jumlah besar membutuhkan jumlah parameter yang besar pula untuk menghasilkan output yang akurat. Maka akurasi klasifikasi pada titik tertentu akan menurun, oleh karena itu dibutuhkan suatu proses yang dapat mengekstraksi jumlah parameter yang diperlukan salah satunya menggunakan proses *pooling*. *Pooling* adalah proses melakukan seleksi untuk mengurangi resolusi pada peta ciri yang tidak signifikan. Dua metode dalam *pooling* yaitu *max pooling* dan *average pooling*, bekerja dengan membagi layer menjadi beberapa grid kecil lalu mengambil nilai terbesar atau nilai rata-rata setiap grid untuk Menyusun matriks data digital yang telah direduksi. Proses *pooling* memerlukan padding untuk menyesuaikan matriks filter dengan mengisi bagian-bagian kosong matriks dengan angka 0. Ukuran matriks filter tidak mempengaruhi ukuran matriks *pooling*. semakin kecil ukuran matriks filter maka proses *pooling* akan semakin rinci. Perpindahan matriks filter pada matriks data ditentukan oleh besarnya stride. Semakin kecil ukuran stride akan semakin rinci akan tetapi ukuran matriks *pooling* akan semakin besar.

Kata Kunci : deep learning; pooling; citra digital.

ABSTRACT

Digital data processing requires algorithms that can process data optimally. Processing large amounts of data requires a large number of parameters to produce accurate output. Then the classification accuracy at a certain point will decrease, therefore a process is needed that can extract the required number of parameters, one of which is using the pooling process. Pooling is a selection process to reduce the resolution on an insignificant feature map. Two methods in pooling, namely max pooling and average pooling, work by dividing the layer into several small grids and then taking the largest value or average value of each grid to compose the reduced digital data matrix. The pooling process requires padding to adjust the filter matrix by filling the empty parts of the matrix with the number 0. The size of the filter matrix does not affect the size of the pooling matrix. The smaller the filter matrix size, the more detailed the pooling process will be. The displacement of the filter matrix in the data matrix is determined by the magnitude of the stride. The smaller the stride size, the more detailed it will be, but the larger the pooling matrix will be.

Keywords : deep learning; pooling; digital image.

@ 2022 Jurnal Riset Matematika Unisba Press. All rights reserved.

A. Pendahuluan

Salah satu problem dalam visi komputer yang telah lama dicari solusinya adalah klasifikasi objek secara umum. Problem ini adalah bagaimana menduplikasi kemampuan manusia dalam memahami informasi data digital, agar komputer dapat mengenali objek pada data digital selayaknya manusia. Proses pengenalan objek yang digunakan pada umumnya sangat terbatas dimana hanya dapat berlaku pada dataset tertentu saja tanpa kemampuan generalisasi pada jenis data digital apapun [1].

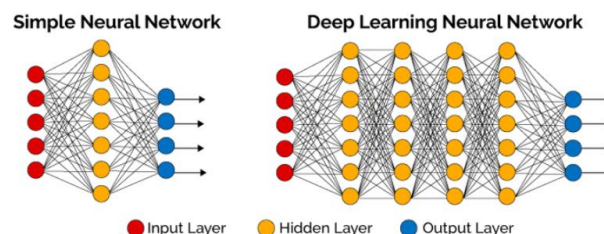
Algoritma deep learning membutuhkan hardware yang powerful karena memproses data dalam jumlah besar dan menggunakan beberapa perhitungan matematis yang kompleks. Algoritma deep learning menggunakan data dalam jumlah besar untuk menghasilkan hasil yang akurat, akan tetapi jika jumlah atribut terus bertambah, sedangkan jumlah sampel data terbatas, maka akurasi klasifikasi ini pada titik tertentu akan menurun. Fenomena inilah yang disebut “*Curse of dimensionality*”[2]. Permasalahan yang timbul adalah data digital memiliki ukuran yang sangat besar, oleh karena itu dibutuhkan suatu proses yang dapat mengekstraksi jumlah parameter yang perlu diperhitungkan, salah satunya menggunakan proses *pooling*.

Pooling adalah proses melakukan seleksi atau mengurangi resolusi *feature maps* yang tidak terpakai. Setiap peta fitur yang dikumpulkan sesuai dengan satu peta fitur dari lapisan sebelumnya. *Pooling* ini memiliki kelebihan untuk mengekstraksi fitur dominan sehingga proses pelatihan model lebih efektif tetapi tetap menjaga informasi yang penting.

B. Metode Penelitian

Deep Learning

Deep learning merupakan sebuah algoritma *neural network* yang menggunakan metadata sebagai input dan mengolah input tersebut menggunakan sekumpulan fungsi transformasi non-linier yang ditata berlapis-lapis dan mendalam [3][4]. Pada *deep learning* terdapat *hidden layer* (lapisan tersembunyi) yang bertugas untuk melatih serangkaian fitur unik berdasarkan output dari jaringan sebelumnya [5]. Berikut adalah ilustrasi perbedaan antara *neural network* sederhana yang hanya menggunakan satu atau dua *hidden layer* dengan *deep learning* yang menggunakan banyak *hidden layer*.



Gambar 1. Deep Learning

Seperti yang tampak pada pada Gambar 1. *neural network* sederhana yang menggunakan jaringan *feed forward* hanya mempunyai satu atau dua hidden layer. Sedangkan pada *deep learning neural network* terdapat *hidden layer* yang jumlahnya diatas dua layer keatas yang bahkan bisa sampai ratusan layer.

Perkembangan ilmu *Neural Network* sudah ada sejak tahun 1943 ketika Warren McCulloch dan Walter Pitts memperkenalkan perhitungan model *Neural Network* yang pertama kalinya. Mereka sama sama melakukan kombinasi beberapa *processing unit* sederhana bersama-sama yang mampu memberikan peningkatan secara keseluruhan pada kekuatan komputasi[6].

Citra Digital

Pada sebuah komputer, citra digital dipetakan menjadi elemen *pixel* berbetuk matriks dua dimensi dan bentuk grid. Pada citra digital dapat diwakili oleh sebuah matriks yang terdiri dari N baris dan M kolom, dimana perpotongan anatar baris dan kolom disebut dengan pixel [7]. *Pixel* kependekan dari *picture element* yaitu

elemen terkecil dari citra. *Pixel* memiliki dua parameter yaitu koordinat dan intensitas atau warna dari *pixel* pada titik itu. Oleh karena itu, citra dapat dituliskan kedalam sebuah matriks:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M - 1) \\ f(1,0) & \dots & \dots & f(1, M - 1) \\ \dots & \dots & \dots & \dots \\ f(N - 1,0) & f(N - 1,1) & \dots & f(N - 1, M - 1) \end{bmatrix}$$

Berdasarkan rumus diatas, suatu citra $f(x,y)$ dapat dituliskan kedalam fungsi matematis seperti dibawah ini :

$$\begin{aligned} 0 &\leq x \leq M - 1 \\ 0 &\leq y \leq N - 1 \\ 0 &\leq f(x, y) \leq G - 1 \end{aligned}$$

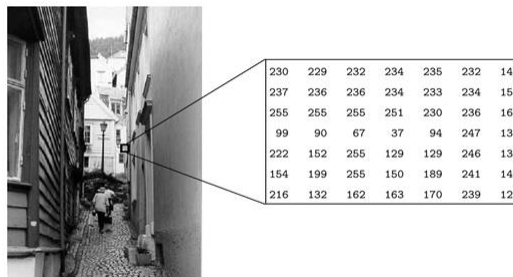
Dimana:

M = Jumlah baris pada citra

N = Jumlah kolom pada citra

G = Nilai skala keabuan.

Citra atau gambar dalam bahasa latin *imago* adalah suatu representasi, kemiripan, atau imitasi dari suatu obyek atau benda. Di antara jenis citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer.



Gambar 2. Citra Digital

Ekstraksi Ciri

Ekstraksi ciri merupakan proses pengindeksan suatu database data input. Secara matematik, setiap ekstraksi ciri merupakan encode dari vektor n dimensi yang disebut dengan vektor ciri. Komponen vektor ciri dihitung dengan pemrosesan data dan teknik analisis serta digunakan untuk membandingkan data input yang satu dengan data input yang lain. Ekstraksi ciri diklasifikasikan ke dalam 3 jenis yaitu *low-level*, *middle-level*, dan *high-level*.

Feature Map

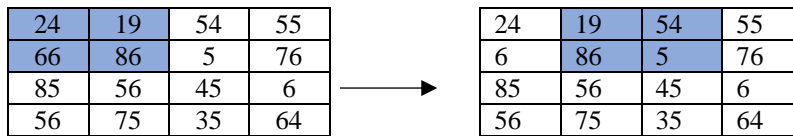
Feature Map atau *Activation Map* merupakan *output* dari proses konvolusi. Proses konvolusi merupakan proses perkalian antara nilai *input* dengan *filter*. *Feature map* ini bisa menganalisis klasifikasi berdasarkan wilayah dari berbagai aspek, yang orthogonal terhadap investigasi fitur [8].

Pooling

Proses *Pooling* adalah salah satu proses untuk pengurangan ukuran matriks data. *Pooling layer* adalah lapisan yang berfungsi untuk mengekstraksi ukuran spasial dari fitur konvolusi sehingga dapat mengurangi sumber daya komputasi yang dibutuhkan untuk memproses data melalui pengurangan dimensi dari *feature map* sehingga mempercepat komputasi karena parameter yang diperbarui semakin sedikit [9].

Stride

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai *stride* adalah 1, maka *filter* akan bergeser sebanyak 1 pixels secara horizontal lalu vertical seperti pada Gambar 3. Namun perlu diperhatikan bahwa dengan menggunakan *stride* yang kecil kita tidak selalu akan mendapatkan performa yang bagus [10].



Gambar 3. Contoh Stide

Padding

Sedangkan *Padding* atau *Zero Padding* adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari *input*.

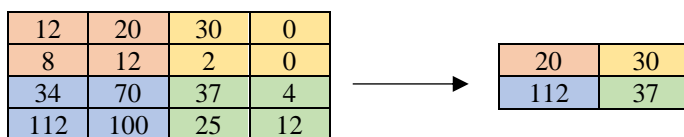


Gambar 4. Contoh Padding

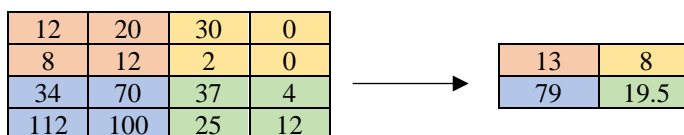
Operasi Pooling

Inti dari lapisan ini adalah pengambilan nilai-nilai penting dari setiap *feature map* dengan mengambil nilai yang maksimum. Terdapat beberapa metode untuk proses *pooling*, dalam beberapa literatur operasi *pooling* yang sering digunakan adalah *Max Pooling* dan *Average Pooling*.

Sebagai contoh jika kita menggunakan *Max Pooling 2x2* dengan *stride 2*, maka pada setiap pergeseran filter, nilai *maximum* pada area *2x2* pixel tersebut yang akan dipilih, sedangkan *Average Pooling* akan memilih nilai rata-ratanya.



Gambar 5. Max pooling



Gambar 6. Avarage Pooling

C. Hasil dan Pembahasan

Pooling adalah Teknik untuk mengekstraksi sejumlah nilai pixel menjadi pixel yang lebih kecil. Tetapi dengan dilakukannya *pooling* ini ukuran dari citra tetap sama, tidak berubah. Telah diketahui bahwa hasil *sampling* dan kuantisasi dari sebuah citra adalah bilangan real yang membentuk sebuah matriks *M* baris dan *N* kolom, ini berarti ukuran citra adalah $m \times n$.

Input Proses

Matriks citra, ukuran matriks yang digunakan pada penelitian ini adalah matriks citra derajat keabuan dengan skala 0-255 yang berukuran 10×10

Tabel 1. Matriks Citra.

62	195	179	179	179	179	179	179	166	109
127	133	92	91	91	90	89	89	79	0
97	82	60	55	49	44	38	33	76	14
83	64	115	111	97	83	81	45	75	84
126	87	111	85	27	3.5	7.1	13	43	84
119	143	76	10	0	11	8.9	70	119	63
117	59	38	0	24	23	13	74	127	28
120	53	17	63	46	35	33	110	162	28
124	108	90	138	117	131	131	115	187	74
204	126	127	128	129	130	126	114	143	63

Proses Pooling

Pada proses *pooling* ini terdapat 2 metode yaitu *max pooling* dan *avarege pooling*. Dalam eksperimen kali ini terdapat matriks 10 x 10 dan filter yang digunakan berukuran 2 x 2 dan 3 x 3 dengan stride 1 dan 2

Max Pooling, pada Max Pooling ini, nilai yang diambil adalah nilai terbesar dari setiap filter. Fiter yang digunakan 2x2 dengan stride 2.

Tabel 2. Max Pooling Fiter 2x2 dengan stride 2

62	195	179	179	179	179	179	179	166	109
127	133	92	91	91	90	89	89	79	0
97	82	60	55	49	44	38	33	76	14
83	64	115	111	97	83	81	45	75	84
126	87	111	85	27	3.5	7.1	13	43	84
119	143	76	10	0	11	8.9	70	119	63
117	59	38	0	24	23	13	74	127	28
120	53	17	63	46	35	33	110	162	28
124	108	90	138	117	131	131	115	187	74
204	126	127	128	129	130	126	114	143	63

Pada nilai matriks 10x10 yang akan di lakukan operasi *max pooling* dengan filter 2x2 dengan stride 2, pada kotak berwarna biru terdapat nilai [62, 195, 127, 133] lakukan dengan operasi *max pooling* dengan filter 2x2 ambil nilai matriks yang paling terbesar dari kotak matriks biru tersebut dan hasilnya mendapatkan nilai 195. Selanjutnya pada kotak berwarna oren terdapat nilai [179, 179, 92, 91] lakukan dengan operasi *max pooling* dengan filter 2x2 ambil nilai matriks yang paling terbesar dari kotak matriks oren tersebut dan hasilnya mendapatkan nilai 179. Selanjutnya pada kotak berwarna ungu terdapat nilai [179, 179, 91, 90] lakukan dengan operasi *max pooling* dengan filter 2x2 ambil nilai matriks yang paling terbesar dari kotak matriks ungu tersebut dan hasilnya mendapatkan nilai 179. Lakukan operasi *max pooling* pada setiap kotak dengan cara yang sama.

Tabel 3. Hasil Max Pooling filter 2x2 stride 2

195	179	179	179	166
97	115	97	81	84
143	111	27	70	119
120	63	46	110	162
204	138	130	131	143

Hingga mendapatkan hasil akhir dari *max pooling* dengan nilai matriks seperti pada Gambar 14.

Pada *Max Pooling* ini, nilai yang diambil adalah nilai terbesar dari setiap filter. Fiter yang digunakan 2x2 dengan stride 1

Tabel 3. Max Pooling filter 2x2 stride 1

62	195	179	179	179	179	179	179	166	109
127	133	92	91	91	90	89	89	79	0
97	82	60	55	49	44	38	33	76	14
83	64	115	111	97	83	81	45	75	84
126	87	111	85	27	3.5	7.1	13	43	84
119	143	76	10	0	11	8.9	70	119	63
117	59	38	0	24	23	13	74	127	28
120	53	17	63	46	35	33	110	162	28
124	108	90	138	117	131	131	115	187	74
204	126	127	128	129	130	126	114	143	63

Pada nilai matriks 10x10 yang akan di lakukan operasi *max pooling* dengan filter 2x2 dengan stride 1, pada kotak berwarna biru terdapat nilai [62, 195, 127, 133] lakukan dengan operasi *max pooling* dengan filter 2x2 ambil nilai matriks yang paling terbesar dari kotak matriks biru tersebut dan hasilnya mendapatkan nilai 195. Selanjutnya pada kotak berwarna oren terdapat nilai [195, 179, 133, 92] lakukan dengan operasi *max pooling* dengan filter 2x2 ambil nilai matriks yang paling terbesar dari kotak matriks oren tersebut dan hasilnya mendapatkan nilai 195. Selanjutnya pada kotak berwarna abu terdapat nilai [179, 179, 92, 91] lakukan dengan operasi *max pooling* dengan filter 2x2 ambil nilai matriks yang paling terbesar dari kotak matriks abu tersebut dan hasilnya mendapatkan nilai 179. Lakukan operasi *max pooling* pada setiap kotak dengan cara yang sama.

Tabel 4. Hasil Max Pooling filter 2x2 stride 1

195	195	179	179	179	179	179	179	166	
133	133	92	91	91	90	89	89	79	
97	115	115	111	97	83	81	76	84	
126	115	115	111	97	83	81	75	84	
143	143	111	85	27	11	70	119	119	
143	143	76	24	24	23	74	127	127	
120	59	63	63	46	35	110	162	162	
124	108	138	138	131	131	131	187	187	
204	127	138	138	131	131	131	187	187	

Hingga mendapatkan hasil akhir dari *max pooling* dengan nilai matriks seperti pada Gambar 18.

Average Pooling, pada Avarage Pooling ini, nilai yang diambil adalah nilai rata-rata dari setiap filter. Fiter yang digunakan 2x2 dengan stride 2.

Tabel 5. Avarage Pooling filter 2x2 stride 2

62	195	179	179	179	179	179	179	166	109
127	133	92	91	91	90	89	89	79	0
97	82	60	55	49	44	38	33	76	14
83	64	115	111	97	83	81	45	75	84
126	87	111	85	27	3.5	7.1	13	43	84
119	143	76	10	0	11	8.9	70	119	63
117	59	38	0	24	23	13	74	127	28
120	53	17	63	46	35	33	110	162	28
124	108	90	138	117	131	131	115	187	74
204	126	127	128	129	130	126	114	143	63

Pada nilai matriks 10x10 yang akan di lakukan operasi *avarage pooling* dengan filter 2x2 dengan stride 2, pada kotak berwarna biru terdapat nilai [62, 195, 127, 133] lakukan dengan operasi *average pooling* dengan filter 2x2 ambil nilai matriks rata-rata dari kotak matriks biru tersebut dan hasilnya mendapatkan nilai 129. Selanjutnya pada kotak berwarna oren terdapat nilai [179, 179, 92, 91] lakukan dengan operasi *avarage pooling* dengan filter 2x2 ambil nilai matriks rata-rata dari kotak matriks oren tersebut dan hasilnya mendapatkan nilai 135. Selanjutnya pada kotak berwarna ungu terdapat nilai [179, 179, 91, 90] lakukan dengan operasi *avarage pooling* dengan filter 2x2 ambil nilai rata-rata terbesar dari kotak matriks ungu tersebut dan hasilnya mendapatkan nilai 135. Lakukan operasi *avarage pooling* pada setiap kotak dengan cara yang sama.

Tabel 6. Hasil Avarage Pooling filter 2x2 stride 2

129	135	135	134	89
81	85	68	49	62
119	71	10	23	77
87	29	32	57	86
140	120	126	121	116

Hingga mendapatkan hasil akhir dari *avarage pooling* dengan nilai matriks seperti pada Gambar 22.

Tabel 7. Avarage Pooling filter 2x2 stride 1

62	195	179	179	179	179	179	179	166	109
127	133	92	91	91	90	89	89	79	0
97	82	60	55	49	44	38	33	76	14
83	64	115	111	97	83	81	45	75	84
126	87	111	85	27	3.5	7.1	13	43	84
119	143	76	10	0	11	8.9	70	119	63
117	59	38	0	24	23	13	74	127	28
120	53	17	63	46	35	33	110	162	28
124	108	90	138	117	131	131	115	187	74
204	126	127	128	129	130	126	114	143	63

Pada nilai matriks 10x10 yang akan di lakukan operasi *avarage pooling* dengan filter 2x2 dengan stride 1, pada kotak berwarna biru terdapat nilai [62, 195, 127, 133] lakukan dengan operasi *avarage pooling* dengan filter 2x2 ambil nilai matriks rata-rata dari kotak matriks biru tersebut dan hasilnya mendapatkan nilai 129,25. Selanjutnya pada kotak berwarna oren terdapat nilai [195, 179, 133, 92] lakukan dengan operasi *avarage pooling* dengan filter 2x2 ambil nilai matriks rata-rata dari kotak matriks oren tersebut dan hasilnya mendapatkan nilai 149,75. Lakukan operasi *avarage pooling* pada setiap kotak dengan cara yang sama.

Tabel 8. Hasil Avarage Pooling filter 2x2 stride 1

129.25	149.75	135.25	135	134.75	134.25	134	1228.75	88.5
109.75	91.75	74.5	71.5	68.5	65.25	62.5	69.25	42.25
86.5	80.25	85.25	78	68.25	61.5	49.25	57.25	62.25
90	94.25	105.5	80	52.62	43.65	36.25	44	71.5
118.75	104.25	70.5	30.5	10.37	7.62	24.75	76.25	77.25
109.5	79	31	8.5	14.5	13.97	41.47	97.5	84.25
87.25	41.75	29.5	33.25	32	26	57.5	118.25	86.25
101.25	67	77	91	82.25	82.5	97.25	143.5	112.75
140.5	112.75	120.5	28	126.75	129.5	121.5	139.75	116.75

Hingga mendapatkan hasil akhir dari *average pooling* dengan nilai matriks seperti pada Gambar 26.

D. Kesimpulan

Teknik *pooling* Teknik yang digunakan untuk mereduksi ukuran matriks data digital yang biasanya berukuran besar. Metoda *pooling* yang biasa digunakan adalah *max pooling* dan *average pooling*. metode *max pooling* mengambil nilai paling besar yang ditunjukkan oleh matriks filter pada matriks data digital. Sedangkan metode *average pooling* mengambil nilai rata-rata yang ditunjukkan oleh matriks filter pada matriks data digital.

Teknik *pooling* memiliki tiga tahapan yang harus dilalui dalam prosesnya yaitu ukuran matriks filter, *stride*, dan *padding*. Matriks filter adalah ukuran matriks yang akan digunakan untuk memproses nilai *max* atau nilai rata-rata dari matriks data. Semakin kecil ukuran matriks filter maka proses *pooling* akan semakin rinci. Perpindahan matriks filter pada matriks data ditentukan oleh besarnya *stride*. Semakin kecil ukuran *stride* maka proses *pooling* akan semakin rinci tetapi ukuran matriks hasil *pooling* akan semakin besar.

Ukuran matriks filter tidak mempengaruhi ukuran matriks hasil *pooling*. Studi kasus menunjukkan bahwa penggunaan ukuran matriks filter yang berbeda menghasilkan ukuran matriks *output* yang sama. Nilai *stride* 1 mereduksi ukuran matriks *output* sebesar 10%, sedangkan nilai *stride* 2 mereduksi ukuran matriks *output* sebesar 50%.

Daftar Pustaka

- [1] W. S. Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.
- [2] 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.
- [3] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, pp. 49–56, 2018.
- [4] A. Ahmad, *Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning*. 2017.
- [5] K. P. Danukusumo, "Convolutional neural network untuk mendeteksi bangunan," pp. 10–22, 2017.
- [6] D. G/Tsadik *et al.*, "Rancangan Bangun Sistem Klasifikasi Mineral dan Batuan Menggunakan Tensorflow.js," 2020.
- [7] M. I. Mardiyah, "Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (Cnn) Pada Tugas Akhir," 2020.
- [8] S. Ren, "Object Detection Networks on Convolutional Feature Maps," 2017.
- [9] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algoritma*, vol. 1, no. 1, pp. 45–56, 2020, doi: 10.35957/algoritme.v1i1.434.
- [10] C. Geraldly and C. Lubis, "Pendeteksian Dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once Dan Convolutional Neural Network," pp. 197–199, 2020.