



PEMILIHAN MODEL TERBAIK ALGORITMA *CONVOLUTIONAL NEURAL NETWORK* UNTUK KLASIFIKASI JENIS BENCANA ALAM

The Best Model Selection Of Convolutional Neural Network Algorithm For Natural Disaster Classification

Aryo Anindyo Abhinowo^{*)}, R. Rizal Isnanto, Dania Eridani.
Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

How to cite: A. A. Abhinowo, R. R. Isnanto, and D. Eridani, "Pemilihan Model Terbaik Algoritma *Convolutional Neural Network* untuk Klasifikasi Jenis Bencana Alam", *Jurnal Teknik Komputer*, vol. 1, no. 4, pp. 199-208, 2023. doi: 10.14710/jtk.v1i4.37656 [Online].

Abstract – Having a proper understanding of natural disasters can greatly reduce the damage caused in terms of property loss and even save lives. By the help of deep learning we try to achieve study aims to classify four specific classes of natural disasters: forest fires, hurricanes, floods, and earthquakes by using *Convolutional Neural Network (CNN)*. Various scenarios will be tested to proposed a good model, these include the use of dropouts, regularization, different combinations of pooling, convolutions, filter size, kernel size, and dataset volume. A total of 93 models will be trained and evaluated, with the ultimate objective being to identify the most efficient model with the smallest file size and fastest computation time. Research results show that the classification method using *CNN* is able to be implemented with an accuracy rate of 91.62%, *F1 Score* 92%, *Recall* 92%, *Precision* 92% and with the test model file is 5.1 MB in size.

Keywords – Natural Disasters; Classification; Deep Learning; *Convolutional Neural Network (CNN)*

Abstrak – Pemahaman yang baik dan benar mengenai bencana alam dapat mengurangi kerusakan dari segi properti, harta benda ataupun kehilangan nyawa. Dalam penelitian ini akan mencoba melakukan klasifikasi terhadap empat bencana alam yaitu kebakaran hutan, angin topan, banjir, dan gempa bumi dengan bantuan Algoritma *Convolutional Neural Network (CNN)*. Terdapat 8 skenario yang digunakan dalam pembuatan model seperti penggunaan dropout, regularisasi, jumlah pooling, jumlah konvolusi, ukuran tapis, ukuran kernel dan juga jumlah data yang digunakan. Terdapat 93 model yang dilatih dalam percobaan ini, hasil akhir yang diinginkan adalah menemukan model terbaik yang memiliki ukuran berkas terkecil dengan komputasi tercepat untuk dijalankan serta kemampuan akurasi model. Berdasarkan hasil penelitian menunjukkan metode klasifikasi dengan menggunakan *CNN* ini sudah

mampu untuk diimplementasikan dengan mendapatkan nilai akurasi sebesar 91,62%, *F1 Score* 92%, *Recall* 92% *Precision* 92% dan ukuran berkas pengujian sebesar 5,1 MB.

Kata Kunci – Bencana Alam; Klasifikasi; Deep Learning; *Convolutional Neural Network (CNN)*

I. PENDAHULUAN

Bencana alam adalah suatu fenomena alam yang memiliki dampak besar terhadap keberlangsungan populasi manusia[1]. Negara Indonesia sebagai negara kepulauan yang berada pada posisi secara geografis terletak dalam pertemuan tiga lempengan bumi yaitu Eurasia, Pasifik, dan Indo-Australia menyebabkan Indonesia termasuk negara yang tergolong rawan terjadi bencana seperti gempa bumi, tsunami, dan longsor. Secara geografis Indonesia juga terletak di daerah sabuk api atau yang dikenal dengan julukan *ring of fire* dimana sebanyak 500 gunung api tersebar di Indonesia dan 127 buah diantaranya merupakan gunung api aktif[2].

Begitu banyaknya potensi terjadi bencana alam yang ada harus membuat Indonesia lebih waspada baik dari segi penanggulangan ataupun evakuasi, jika pemerintah dapat mengatasi hal tersebut tentu nya dapat membantu keselamatan dari populasi yang ada. Dengan seiring adanya perkembangan teknologi, besar harapan agar pencegahan bencana alam dapat dilakukan sebelum bencana itu terjadi dengan memanfaatkan bantuan dari Kecerdasan Buatan. Penelitian ini berfokus terhadap pengujian parameter yang ada tanpa menggunakan bantuan *transfer learning* agar dapat meminimalisir penggunaan jumlah parameter yang tidak diinginkan serta didapatkan hasil komputasi tercepat dalam melakukan pembelajaran.

Dalam penelitian ini digunakan Arsitektur *Convolutional Neural Network (CNN)* sebagai bentuk klasifikasi bencana alam yang ada. Alasan dipilihnya arsitektur tersebut karena *CNN* memberikan akurasi yang lebih tinggi pada jenis basis data gambar objek benda, dibandingkan objek lain seperti basis data kata atau gelombang sinyal[3]. *CNN* itu sendiri juga sudah terkenal dan umum

^{*)} Penulis Korespondensi (A. A. Abhinowo)
Email: aryoanindyo@student.ce.undip.ac.id



untuk melakukan kasus klasifikasi dengan bantuan fitur *deep-level* yang dimilikinya[4] seperti dalam pengerjaan basis data MNIST, CIFAR10, Caltech-10 dan ImageNET. CNN sangat bergantung pada arsitektur dalam melakukan pembelajaran model dan untuk mendapatkan performa terbaik sudah terdapat arsitektur pembelajaran yang dimiliki oleh CNN seperti GoogLeNet, ResNet dan DenseNet[5].

Pengembangan penelitian ini juga dibantu dengan penggunaan Tensorflow sebagai algoritma dasar untuk pembentukan model sekuensial dari CNN, berkat peningkatan dukungan TensorFlow terhadap penggunaan *Graphic Processing Unit* (GPU). Flask dalam penelitian ini digunakan sebagai pembuatan API perantara agar model latih dapat dijalankan dalam Google Cloud App Engine sebagai kebutuhan dalam melakukan prediksi data untuk nantinya dijalankan dalam aplikasi perangkat Android.

II. KAJIAN LITERATUR

A. Kajian Penelitian Terdahulu

Pada penelitian terdahulu mengenai *deep learning* yang ditulis oleh M Anbarasan, BalaAnand Muthu, C.B Sivaparthipan, Revathi Sundarasekar, Seifedine Kadry, Sujatha Krishnamoorthy, Dinesh Jackson S R, dan A. Anthony Dasel yang membahas mengenai pemanfaatan IoT dalam pendeteksian kejadian banjir dengan bantuan *Convolutional Deep Neural Network* (CDNN) yang dimana algoritma tersebut memiliki keunggulan dalam hal ini dibanding menggunakan Jaringan Syaraf Tiruan (JST)[6]. Dalam studi komparasi parameter berupa Sensitivitas, Spesifitas, Akurasi, Presisi, Recall dan juga *F1-Score* yang digunakan dalam penelitian ini CDNN memiliki nilai paling tinggi dari data sebesar 500 dengan Akurasi sebesar 92,23%, Sensitivitas sebesar 91,41%, Spesifitas 91,56%, Presisi sebesar 92,23%, Recall sebesar 90,36%, dan F-Score sebesar 91,28%[6]. Penelitian ini nantinya akan dijadikan referensi dalam melakukan pengembangan model kasus bencana dengan menggunakan 5 lapisan neuron dengan isi dari tiap fitur parameternya dilakukan secara uji coba.

Penelitian selanjutnya tentang *deep learning* dilakukan oleh Dean Rizki Hartawan, Tito Waluyo Purboyo, dan Casi Setianingsih dengan penelitian berjudul *Disaster Victims Detection System using Convolutional Neural Network* (CNN) Method. Basis Data yang digunakan sebanyak 3.542 gambar untuk data berlabel positif dan 852 data berlabel negatif[7]. Arsitektur yang digunakan pada percobaan ini menggunakan CNN dengan perolehan akurasi sebesar 100% untuk gambar dengan jarak yang bisa dilihat menggunakan Raspberry Pi sejauh 1 hingga 4 meter. Nilai akurasi mulai mengalami penurunan signifikan setelah objek prediksi melebihi jarak dari 4 meter ke atas. Penelitian ini akan dijadikan sebagai referensi pengerjaan dengan hanya menggunakan data uji dan data latih tanpa penggunaan data validasi serta jumlah basis data yang digunakan hanya sebanyak 3.542 gambar dalam melakukan pembelajaran model.

Penelitian berikutnya yang ditulis oleh Lulwa Ahmed, Kashif Ahmad, Naina Said, Basheer Qolomany, Junaid Qadir, dan Ala Al-Fuqaha mengenai penggunaan *Active Learning* dan *Federated Learning* untuk mengklasifikasi bencana alam dan juga penggunaan limbah[8]. Dalam pengaplikasiannya, penggunaan *Federated Learning* untuk mengklasifikasi kedua hal tersebut bertujuan agar mempermudah pembacaan basis data yang memiliki kriteria baru, besar dan tidak berlabel agar bisa segera diproses dan nantinya dilakukan pelatihan data secara bersamaan dalam proses yang dilakukan secara *offline*. Penelitian ini akan dijadikan referensi pengerjaan terkait kebutuhan jumlah basis data bencana alam dalam proses pembelajaran model serta total penggunaan iterasi yang dilakukan. Dalam kasus ini jumlah basis data sebanyak 7000 gambar dengan total iterasi untuk *Active Learning* sebanyak 1500 kali.

Berikutnya terdapat penelitian mengenai dampak ukuran tapis untuk mendapatkan akurasi yang baik dengan metode CNN yang ditulis oleh Wafaa Shihab Ahmed dan Abdul amir A. Karim[9] Dari penelitian ini dilakukan beberapa percobaan dengan menggunakan tapis dan ukuran kernel yang berbeda-beda, diperoleh hasil penggunaan tapis berukuran 128 lebih baik dibandingkan dengan ukuran 32 dan 64 meskipun proses komputasi yang dilakukan akan lebih lama akhirnya. Akurasi terbaik dalam data uji didapati ketika model memiliki empat konvolusi dan juga dua lapisan *pooling* di dalamnya. Penelitian ini akan dijadikan acuan dalam membuat tapis dan ukuran kernel untuk pembelajaran model CNN dalam kasus bencana alam.

Penelitian terakhir yang diambil untuk dijadikan referensi yaitu penelitian oleh Osman Dogus Gulgun dan Prof Dr Hamza Erol[10]. Penelitian ini memanfaatkan algoritma CNN dalam mengklasifikasikan gambar berupa x-ray untuk mengetahui penyakit pneumonia. Pengujian akurasi model dalam penelitian ini menggunakan jumlah basis data sebanyak 5.840 gambar dengan data latih dibagi menjadi 5.216 gambar dan data uji sebesar 624 gambar. Hasil penelitian mendapatkan akurasi sebesar 93,4% untuk pengujian dari data uji. Dari penelitian ini disimpulkan penambahan data dengan menggunakan augmentasi gambar dapat menaikkan akurasi sebesar 13% dan pengurangan *loss* sebesar ~60%. Pada kesimpulan tersebut nantinya akan dijadikan acuan untuk melakukan augmentasi gambar pada kasus bencana alam.

B. Convolutional Neural Network

Convolutional Neural Network adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Lapisan CNN memiliki susunan neuron 3 dimensi yaitu lebar, tinggi, dan kedalaman[11]. Lebar dan tinggi adalah ukuran lapisan, sedangkan kedalaman mengacu pada jumlah lapisan. Sebuah CNN dapat memiliki puluhan hingga ratusan lapisan yang masing-masing lapisan bertujuan untuk belajar mendeteksi berbagai gambar.



C. Padding

Padding adalah parameter jaringan saraf konvolusi yang mengacu pada manipulasi jumlah piksel yang ditambahkan ke gambar ketika sedang diproses oleh kernel CNN. Dengan *padding* akan didapatkan hasil dimensi keluaran agar memiliki ukuran sama dengan dimensi masukan[12].

D. Filter dan Kernel

Tapis (*filter*) berfungsi untuk mendeteksi pola spasial seperti tepi pada gambar dengan mendeteksi perubahan nilai intensitas gambar. Definisi dari tapis ini juga mengacu kepada struktur 3 dimensi dari tumpukan kernel yang saling menyatu. Salah satu alasan untuk memilih ukuran kernel yang kecil daripada jaringan yang sepenuhnya terhubung dikarenakan hal itu mengurangi biaya komputasi dan pembagian bobot yang pada akhirnya mengarah pada bobot yang lebih rendah untuk propagasi balik. Umumnya penggunaan tapis ganjil merupakan pilihan yang optimal untuk diikuti oleh para praktisi hingga saat ini[13]. Dalam pemrosesan gambar, kernel, matriks konvolusi, atau mask ini juga bisa digunakan untuk pengaburan, penajaman, embossing, deteksi tepi, dan banyak lagi.

E. Precision, Recall dan F1 Score

Untuk mengetahui baik tidaknya suatu model yang sudah dibuat, dapat dilakukan suatu evaluasi dengan menggunakan matriks seperti *precision*, *recall* dan juga *F1 Score*. Ketiga hal ini berkaitan dengan penggunaan *confusion matrix* dan juga istilah *True Positive*, *True Negative*, *False Positive*, *False Negative*.

Hal ini dapat dikembangkan kembali dalam bentuk pelabelan yang dijelaskan seperti berikut

- True Positive* (TP) : Data yang memiliki label positif diprediksi sebagai data positif.
- True Negative* (TN) : Data yang memiliki label negatif diprediksi sebagai data negatif.
- False Positive* (FP) : Data yang memiliki label negatif diprediksi sebagai data positif.
- False Negative* (FN) : Data yang memiliki label positif diprediksi sebagai data negatif.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

F. Akurasi dan Loss

Akurasi merupakan sebuah nilai yang didapatkan dari jumlah hasil prediksi yang benar berbanding dengan jumlah semua data untuk dijadikan sebagai tingkat keberhasilan dari model yang sebelumnya sudah dibuat.

Sedangkan *loss* merupakan parameter dari tingkat keyakinan suatu sistem dalam memprediksi hasil yang salah. Fungsi *loss* melakukan perbandingan antara.

G. Regularisasi L2

Regularisasi jaringan syaraf tiruan adalah teknik yang digunakan untuk mengurangi kemungkinan model *overfitting*[14]. Regularisasi L2 bekerja dengan menambahkan nilai norm, pemberian nilai norm ini ditujukan kepada bobot, ketika nilai bobot semakin besar akan berpengaruh besar terhadap perubahan fitur masukan, maka dari itu akan diberikan sanksi terhadap bobot ini. Regularisasi L2 menimbang kesalahan dengan nilai kuadratnya, sehingga nilai yang diperoleh dapat diambil secara signifikan.

H. Dropout

Dropout merupakan salah satu metode regularisasi yang digunakan untuk mencegah terjadinya *overfitting*, dropout bekerja dengan menghapus beberapa node secara acak dan periodik dalam proses pelatihan. Dropout dapat diterapkan pada akhir dari beberapa layer dan memastikan bahwa tidak ada neuron yang terlalu bergantung pada neuron lain[15].

I. Max Pooling

Dalam cara kerja *max pooling*, diambil suatu representasi nilai yang tertinggi sebagai perwakilan dari setiap petak yang ada dari masukan. Ini berarti meski telah dilakukan sebuah reduksi informasi yang terkandung dalam gambar utama namun tetap dapat mempertahankan fitur yang paling mencolok ketika dilakukan konvolusi.

J. Adam

Adam atau *Adaptive Moment Estimation* adalah sebuah algoritma optimasi yang biasa digunakan dalam pengerjaan *neural network* untuk memperbarui bobot jaringan dengan melakukannya secara berulang pada data latih. Adam memiliki keunggulan ringan dalam penggunaan sumber daya.

III. METODE PENELITIAN

A. Data Penelitian

Dalam penelitian ini basis data diambil dari *website* *pyimagesearch* dengan judul basis data bernama *Natural Disaster Dataset*. Basis data ini dibuat oleh Gautam Kumar dengan bantuan Google Images. Basis data memiliki format *.jpg* yang masing-masing resolusi dan juga rasio ukurannya adalah beragam. Total terdapat sebanyak 4.428 gambar dengan masing-masing kelas terdiri dari:

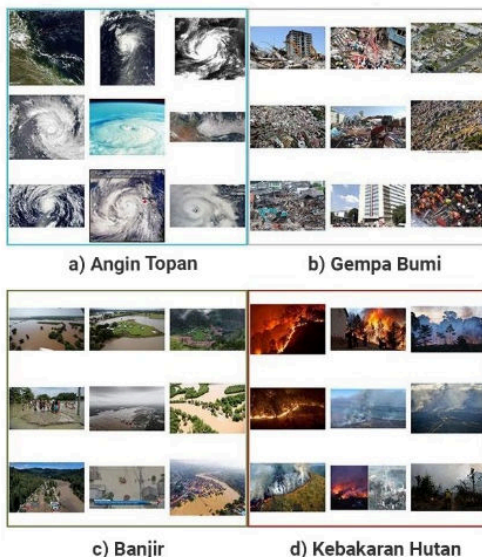
- Angin Topan: 928 gambar
- Gempa Bumi: 1350 gambar
- Banjir: 1073 gambar
- Kebakaran Hutan: 1077 gambar

Dikarenakan basis data dalam percobaan masih belum memiliki jumlah yang banyak di masing-masing kelas maka akan dilakukan sebuah proses augmentasi untuk

membantu penambahan jumlah data. Proses augmentasi yang dilakukan berupa pembesaran gambar, rotasi gambar, melakukan pengaturan pencahayaan dan yang terakhir adalah pengaturan elastisitas distorsi gaussian. Total dari gambar setelah dilakukan Augmentasi menjadi sebanyak 11.430 gambar dengan masing-masing kelas menjadi:

- a) Angin Topan: 2928 gambar
- b) Gempa Bumi: 3051 gambar
- c) Banjir: 2874 gambar
- d) Kebakaran Hutan: 2577 gambar

Berikut adalah contoh gambar basis data yang digunakan dalam penelitian yang dapat dilihat pada Gambar 1.



Gambar 1. Cuplikan contoh gambar pada basis data

B. Variabel Terikat dan Variabel Bebas

Variabel terikat yang digunakan pada model dirangkum sebagai berikut:

- Arsitektur : Convolutional Neural Network
- Input Resolution : 224 x 224 piksel[16]
- Optimizer : Adam
- Loss : Sparse Categorical Crossentropy
- Metrics : Accuracy
- Layer Activation : ReLu
- Last Dense Layer Activation : Softmax

Variabel bebas yang dibandingkan dapat dirinci sebagai berikut:

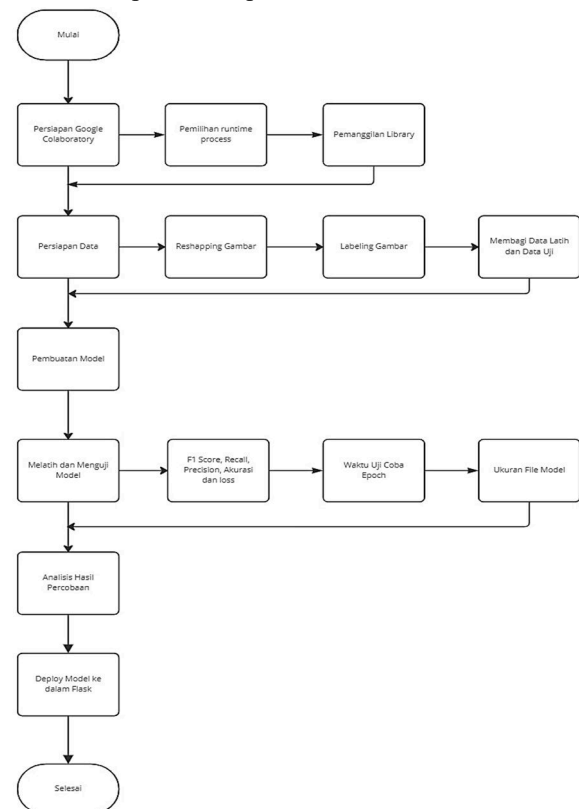
- Train Test Ratio : 70:30, 80:20, 90:10
- Training Images : 3099, 3542, 3985, 8019, 9165, 10311
- Test Images : 1329, 886, 443, 3438, 2292, 1146
- Defined Max Epochs : 15, 20
- Persentase Dropout : 50%, 60%, 70%
- Total MaxPooling : 1, 2, 3, 4
- Regularisasi : L2 10%
- Kernel size : (3,3), (5,5) dan (7,7)

Filter : 16, 32

C. Langkah Penelitian

Pada aplikasi Anti Bencana Alam Nasional (ANCANA) penggunaan *deep learning* dimanfaatkan dalam aplikasi penggunaannya dalam proses *submit* verifikasi detail laporan, hal ini bertujuan sebagai bentuk autentikasi apakah laporan tersebut benar-benar terjadi atau tidak.

Dalam melakukan penelitian ini tahap dari alur pembuatan model yang dibuat oleh penulis dengan bantuan *flowchart* dapat dilihat pada Gambar 2.



Gambar 2. Tahapan penelitian

Dilakukan sebuah tahap berupa pra pengolahan basis data agar dapat diolah untuk tahap pembuatan model, proses yang dilakukan di tahap ini adalah perubahan ukuran data, normalisasi dan juga pelabelan gambar, data diberikan label secara penomoran dengan tujuan agar bisa digunakan secara fleksibel ketika terdapat perubahan seperti penggunaan data dilakukan normalisasi untuk menghilangkan redundansi data sehingga nantinya data memiliki rentang angka dari 0 hingga 1. Ukuran data dalam gambar ini diubah menjadi ukuran sebesar 224x224 piksel.

Selanjutnya dalam tahap pembuatan model *convolutional neural network* akan dilakukan dengan membuat model yang sederhana terlebih dahulu untuk dilakukan sebagai tahap pengujian. Langkah awal dalam membuat model CNN dengan cara mendeklarasikannya secara sekuensial dengan bantuan pustaka Keras sebagai bentuk awal dari membuat kerangka *neural network* itu



sendiri, dengan tiap lapisan umumnya memiliki satu masukan tensor dan satu keluaran tensor.

Tahapan berikutnya adalah melatih dan menguji model dengan melakukan *compile* secara menyeluruh terhadap keseluruhan parameter yang sudah ada. Hal ini merupakan bagian utama dari proses *deep learning* karena pada tahap ini model dirancang untuk membaca dan mempelajari data. Model melakukan *compile* menggunakan *optimizer* adam, *loss* berupa sparse categorical crossentropy dan menggunakan *metric* akurasi.

D. Daftar Percobaan

Dalam tahapan percobaan dilakukan pelatihan dan pengujian model, model akan dilatih menggunakan basis data yang telah dipersiapkan sesuai dengan kebutuhan parameter dan nantinya model akan dievaluasi menggunakan data uji. Terdapat 93 percobaan yang dilakukan oleh pengujian dalam melakukan pengujian model dengan 13 model awal memiliki arsitektur yang dapat dilihat dalam Tabel 1.

Tabel 1. Daftar percobaan arsitektur pertama

Regu- lari- sasi	Drop- out	Tapis	Ukuran Kernel	Total Kon- volusi	Total Max- Pool- ing
-	-	16;32	5,5;3,3	2	1
-	-	16;32	5,5;3,3	2	2
-	-	16;16 ;16	7,7;5,5; 3,3	3	3
-	-	32;32 ;32	7,7;5,5; 3,3	3	3
-	-	16;16 32;32	7,7;7,7 ;5,5;5,5	4	4
-	-	16;16 32;32	5,5;5,5; 3,3; 3,3	4	4
-	-	16;16 32;32	7,7;5,5; 3,3; 3,3	4	4
-	50%	16;16 32;32	5,5;5,5; 3,3; 3,3	4	4
-	60%	16;16 32;32	5,5;5,5; 3,3; 3,3	4	4
-	70%	16;16 32;32	5,5;5,5; 3,3; 3,3	4	4
L2	50%	16;16 32;32	5,5;5,5; 3,3; 3,3	4	4
L2	70%	16;16 32;32	5,5;5,5; 3,3; 3,3	4	4
L2	70%	16,16 32,32	7,7;5,5; 3,3; 2,2	4	4

Poin utama dalam percobaan ini, model ini masih menggunakan basis data dengan total gambar sebesar 4.428. Kemudian dari 13 model tersebut dikalikan sebanyak 3 menjadi 39 model untuk dilakukan percobaan

dengan uji coba rasio data uji dan latih sesuai variabel bebas yang ada.

Dalam Tabel 2 terdapat sebanyak 18 percobaan dengan masing-masing parameter yang ada, basis data yang digunakan dalam percobaan ini sebanyak 11.430 gambar yang kemudian akan dilakukan uji coba rasio data latih dan uji sesuai variabel bebas sehingga terdapat 54 percobaan.

Tabel 2. Daftar percobaan arsitektur kedua

Drop out	Tapis	Ukuran Kernel	Total Kon- volusi	Total Max Pooling
50%	16,16, 32,32	5,5;5,5; 3,3; 3,3	4	4
50%	32,32, 64,64	5,5;5,5; 3,3; 3,3	4	4
50%	32,32, 64,64	7,7;7,7; 5,5;5,5	4	4
50%	16,16, 32,32	7,7;7,7; 5,5;5,5	4	4
50%	16,16, 32,32	5,5;5,5; 3,3; 3,3	4	4
50%	16,16, 32,32, 32	5,5;5,5; 3,3; 3,3;3,3	5	4
60%	16,16, 32,32	5,5;5,5; 3,3; 3,3	4	4
60%	32,32, 64,64	5,5;5,5; 3,3; 3,3	4	4
60%	32,32, 64,64	7,7;7,7; 5,5;5,5	4	4
60%	16,16 ,32,32	7,7;7,7; 5,5;5,5	4	4
60%	16,16, 32,32	5,5;5,5; 3,3; 3,3	4	4
60%	16,16, 32,32 ,32	5,5;5,5; 3,3; 3,3;3,3	5	4
70%	16,16, 32,32	5,5;5,5; 3,3; 3,3	4	4
70%	32,32, 64,64	5,5;5,5; 3,3; 3,3	4	4
70%	32,32, 64,64	7,7;7,7; 5,5;5,5	4	4
70%	16,16, 32,32	7,7;7,7; 5,5;5,5	4	4
70%	16,16 ,32,32	5,5;5,5; 3,3; 3,3	4	4
70%	16,16 ,32,32, 32	5,5;5,5; 3,3; 3,3;3,3	5	4



III. HASIL DAN PEMBAHASAN

A. Pengujian dan Analisis

Dari Tabel 3 dirincikan mengenai penggunaan arsitektur CNN dengan penggunaan *dropout* pada masing-masing rasio yang digunakan dalam percobaan dan model menggunakan data latih sebesar 4.428 gambar. Terdapat 4 representasi *dropout* dalam tabel yang disajikan dari tiap rasio dari 13 percobaan yang dilakukan. Pemilihan 4 representasi data berdasarkan referensi dari arsitektur yang sudah mendekati keluaran yang diinginkan dengan parameter penggunaan ditekankan pada *dropout* itu sendiri. Kesimpulan yang didapatkan dari pengujian penggunaan *dropout* dapat diketahui bahwa *dropout* berhasil mengurangi nilai *loss* yang ada, namun untuk memilih model dengan angka persentase terbaik yang digunakan masih memiliki ambiguitas dikarenakan hal tersebut harus disamakan dengan keterkaitannya terhadap data serta parameter lain yang digunakan dalam percobaan ini.

Tabel 3. Hasil evaluasi *dropout*

Rasio Data Latih:Uji	Dropout	Akurasi Uji (%)	Loss Uji (%)
70 :30	-	80,44	88,5
	50%	87,13	53,34
	60%	85,48	58,41
	70%	84,8	50,89
80:20	-	81,94	131,21
	50%	87,02	50,02
	60%	85,54	49,34
	70%	84,54	47,83
90:10	-	85,33	86,58
	50%	86,91	50,03
	60%	87,13	49,45
	70%	88,94	49,48

Dari Tabel 4 masing-masing arsitektur didapatkan akurasi yang lebih tinggi dan *loss* lebih rendah dengan implementasi *max pooling* yang lebih banyak, hal ini berlaku baik pada saat memprediksi data latih maupun data uji. Dapat disimpulkan bahwa penggunaan *max pooling* lebih cocok diterapkan pada arsitektur dan basis data yang digunakan dalam penelitian ini dikarenakan *max pooling* membantu dalam ekstraksi fitur seperti tepi, poin lebih mendetail dengan menampilkan representasi gambar yang lebih baik[17].

Tabel 4. Hasil evaluasi total *max pooling* dan total konvolusi

Rasio Data Latih :Uji	Total Max Pooling	Total Konvolusi	Akurasi Uji (%)	Loss Uji (%)
70:30	1	2	81,94	94,23
	2	2	81,94	71,86
	3	3	82,02	109,99
	4	4	85,1	49,63

80:20	1	2	83,41	89,86
	2	2	85,1	79,67
	3	3	82,84	98
	4	4	85,89	47,05
90:10	1	2	81,94	112,18
	2	2	86	91,84
	3	3	83,3	106,78
	4	4	88,94	49,48

Dari Tabel 5 penggunaan regularisasi L2 berhasil menemukan nilai akurasi pengujian yang sesuai pada rasio uji dan latih di 70:30 dan 80:20 dengan nilai ukuran kernel berada pada 7,7;5,5; 3,3;3,3 dan tapis berukuran 16;16;32;32. Namun untuk rasio 90:10 sendiri regularisasi L2 belum mendapatkan hasil akurasi terbaiknya melainkan akurasi terbaik didapatkan dari penggunaan tanpa regularisasi dengan ukuran kernel serta tapis yang sama dengan rasio 70:30 serta 80:20. Dapat disimpulkan bahwa penggunaan regularisasi dipengaruhi juga dari rasio uji latih yang ada dan juga parameter *dropout* yang digunakan sebelumnya. Regularisasi L2 tidak mempengaruhi secara besar dari nilai akurasi dan *loss* pengujian.

Tabel 5. Hasil evaluasi regularisasi L2

Rasio Latih :Uji	L2	Drop-out	Ker-nel	Tapis	Akurasi Uji (%)	Loss Uji (%)
70:30	-	50	5,5; 3,3; 3,3	16; 32; 32	87,13	53,34
	-	60	5,5; 3,3; 3,3	16; 32; 32	85,48	58,41
	-	70	5,5; 3,3; 3,3	16; 32; 32	84,8	50,89
	10%	50	5,5; 3,3; 3,3	16; 32; 32	85,55	61,02
	10%	70	5,5; 3,3; 3,3	16; 32; 32	85,55	49,72
	10%	70	7,7; 5,5; 3,3; 3,3	16; 16; 32; 32	85,1	49,63
	-	50	5,5; 3,3; 3,3	16; 32; 32	87,02	50,02
	-	60	5,5; 3,3; 3,3	16; 32; 32	85,54	49,34



		3,3;	32;		
		3,3	32		
		5,5;	16;		
		5,5;	16;		
-	70	3,3;	32;	84,54	47,83
		3,3	32		
		5,5;	16;		
10%	50	5,5;	16;	85,67	57,62
		3,3;	32;		
		3,3	32		
		5,5;	16;		
10%	70	5,5;	16;	84,54	53,89
		3,3;	32;		
		3,3	32		
		7,7;	16;		
10%	70	5,5;	16;	85,89	47,05
		3,3;	32;		
		3,3	32		
		5,5;	16;		
-	50	5,5;	16;	86,91	50,03
		3,3;	32;		
		3,3	32		
		5,5;	16;		
-	60	5,5;	16;	87,13	49,45
		3,3;	32;		
		3,3	32		
		5,5;	16;		
-	70	5,5;	16;	88,94	49,48
		3,3;	32;		
		3,3	32		
90:10		5,5;	16;		
		5,5;	16;		
10%	50	3,3;	32;	83,97	65,16
		3,3	32		
		5,5;	16;		
10%	70	5,5;	16;	85,51	53,71
		3,3;	32;		
		3,3	32		
		7,7;	16;		
10%	70	5,5;	16;	84,2	57,81
		3,3;	32;		
		3,3	32		

Dapat diperhatikan pada Tabel 6 dilakukan pengujian dari model dan hasil terbaiknya dituliskan secara tiga sampel dari masing-masing rasio yang ada. Dari hasil percobaan yang ada, dapat diketahui bahwa dari 9 model tersebut masih memiliki angka *loss* yang cukup besar bahkan ketika sudah diimplementasikan parameter berupa *dropout* maupun regularisasi L2 sebesar 10%. Maka dari itu percobaan akan kembali dilakukan dengan jumlah basis data yang lebih besar untuk mengetahui apakah model dapat belajar dengan jauh lebih baik dengan jumlah basis data yang kecil.

Tabel 6. Hasil evaluasi model pertama

Rasio Percobaan Latih:Uji	Kode Percobaan	Loss Uji	F1 Score Uji
70:30	10	50,89%	84,80%

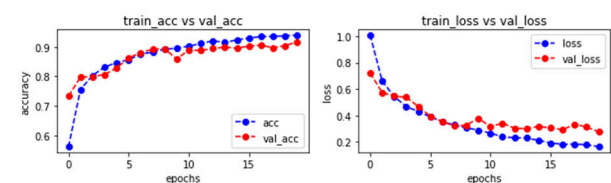
70:30	12	49,72%	85,55%
70:30	13	49,63%	85,10%
80:20	9	49,34%	85,54%
80:20	10	47,83%	84,54%
80:20	13	47,05%	85,59%
90:10	9	49,45%	87,13%
90:10	10	49,48%	88,94%
90:10	12	57,81%	85,51%

Pada Tabel 4 dapat diketahui bahwa model mengalami peningkatan dalam melakukan pembelajaran yang dapat dilihat dari nilai *loss* yang semakin kecil dan juga nilai *F1 score* yang semakin besar. Dari percobaan ini ditentukan bahwa kode percobaan 18 dengan rasio 80:20 merupakan model terbaik. Hal ini juga dapat dilihat dari ukuran bekas yang disimpan dalam bentuk .H5 sebesar 5,1 MB yang tergolong kecil untuk dilakukan pengunggahan nantinya dalam aplikasi Android.

Tabel 7. Hasil evaluasi model kedua

Rasio Percobaan Latih:Uji	Kode Percobaan	Loss Uji	F1 Score Uji	Ukuran Berkas
70:30	17	31,25%	90%	9,8 MB
70:30	18	32,49%	90%	5,1 MB
80:20	12	38,95%	89%	5,1 MB
80:20	18	27,5%	92%	5,1 MB
90:10	14	37,07%	88%	5,1 MB
90:10	17	32,33%	90%	5,1 MB

Pada Gambar 3 dilakukan proses pelatihan data sebanyak 20 *epoch* dengan *epoch* terbaik di dapat di urutan ke 20 yaitu dengan nilai akurasi sebesar 91,62% dan *loss* sebesar 27,52%.



Gambar 3. Grafik riwayat pengujian kode percobaan ke-4

Dari model dengan kode percobaan 18 dengan rasio 80:20 dilihat pada Gambar 4 terdapat total parameter sebanyak 434.916 yang dimana model CNN dimulai dengan melakukan konvolusi terlebih dahulu dengan memberikan masukan gambar sebesar 224x224. Total konvolusi yang dilakukan dalam percobaan ini sebanyak 5 kali dan total *max pooling* sebanyak 4 kali. Adapun penggunaan *dropout* yang digunakan sebesar 60%, proses *flatten* dalam pembuatan model diperlukan untuk mempermudah komputasi menjadi sebuah vector masukan yang panjang sehingga dapat diproses oleh *neural network*. Proses ini diakhiri dengan penggunaan *dense*



dengan nilai sebesar 4 sesuai dengan jumlah kelas dalam klasifikasi yang ada.

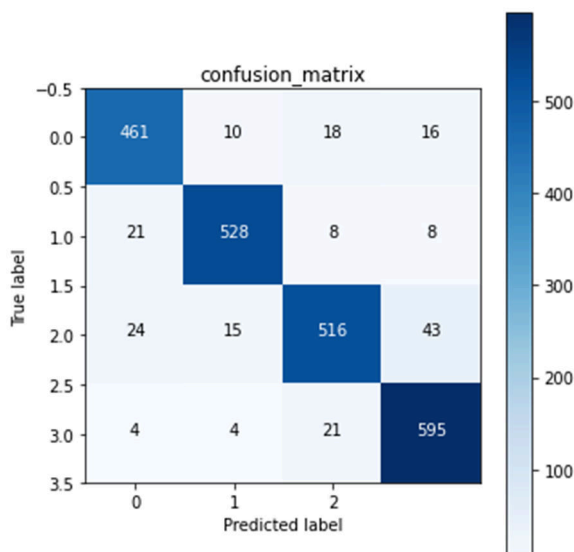
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 16)	1216
max_pooling2d (MaxPooling2D)	(None, 112, 112, 16)	0
conv2d_1 (Conv2D)	(None, 112, 112, 16)	6416
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 16)	0
conv2d_2 (Conv2D)	(None, 56, 56, 32)	4640
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 32)	0
conv2d_3 (Conv2D)	(None, 28, 28, 32)	9248
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 32)	0
dropout (Dropout)	(None, 14, 14, 32)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 128)	802944
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 32)	2080
dense_3 (Dense)	(None, 16)	528
dense_4 (Dense)	(None, 4)	68

=====
 Total params: 835,396
 Trainable params: 835,396
 Non-trainable params: 0

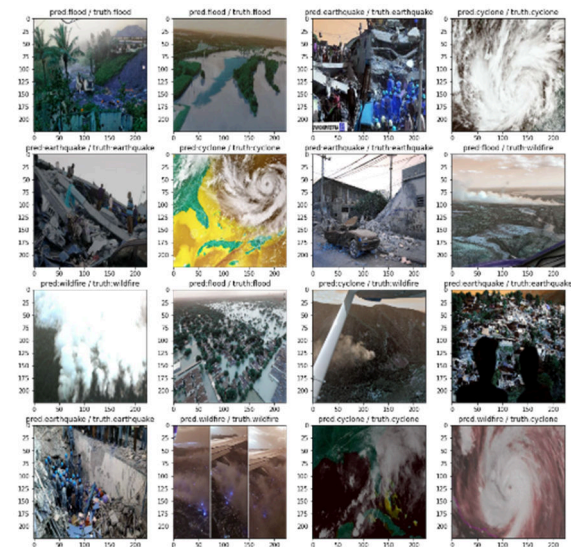
Gambar 4. Ringkasan model

Confusion matrix dari Gambar 5 digunakan sebagai media untuk memahami tingkat kemampuan model dalam melakukan prediksi. Dapat diketahui bahwa masing-masing kelas memiliki tingkat kemampuan yang kuat dalam melakukan pengklasifikasian tiap bencana alam. Nilai *Recall* dalam model ini mendapatkan angka sebesar 92%, hal ini dijadikan sebagai acuan karena penting untuk model dapat memprediksi secara benar tiap kejadian bencana yang ada sebagai bentuk pelaporan yang sempurna nantinya. Untuk Nilai *Precision* yang didapatkan dari model ini sebesar 92%.



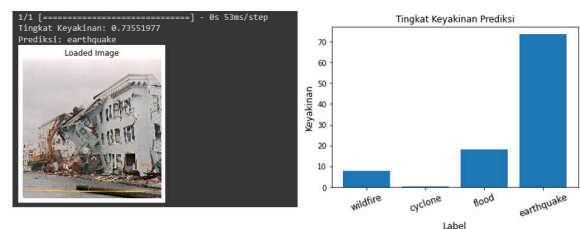
Gambar 5. Confusion matrix

Model yang sudah dipilih tadi kemudian akan dilakukan pengujian dengan menggunakan 16 gambar bencana alam secara acak yang hasilnya dapat dilihat pada Gambar 6. Pada hasil uji coba tersebut, model hampir secara baik memprediksi dengan tepat kejadian bencana alam yang ada. Namun kesalahan masih umum terjadi disaat melakukan prediksi gambar, dikarenakan pada kebanyakan gambar dalam basis data menampilkan kesamaan yang terjadi paska kejadian bencana terjadi, seperti bangunan yang hancur, asap dan awan tebal serta warna yang hampir serupa dalam satu gambar. Hal tersebut tentunya menjadi sebuah ambiguitas terhadap proses pembelajaran yang ada pada model sehingga tidak dapat meyakini suatu gambar dengan keyakinan yang besar atau sempurna.



Gambar 6. Prediksi 16 gambar acak kode percobaan ke-4

Evaluasi uji coba kedua dilakukan pada kejadian secara spesifik bencana alam gempa bumi seperti pada Gambar 7. Dalam pengevaluasian model gempa bumi memiliki tingkat keyakinan sebesar 73,55%, model juga meyakini gambar berkaitan dengan bencana banjir sebesar 18,22%, kebakaran 7,86% dan angin topan sebesar 0,36%.



Gambar 7. Tingkat keyakinan pada gambar spesifik gempa

B. Perancangan Model dalam Flask

Dalam melakukan perancangan model untuk tahap Flask diperlukan lima komponen penting yaitu app.yaml,

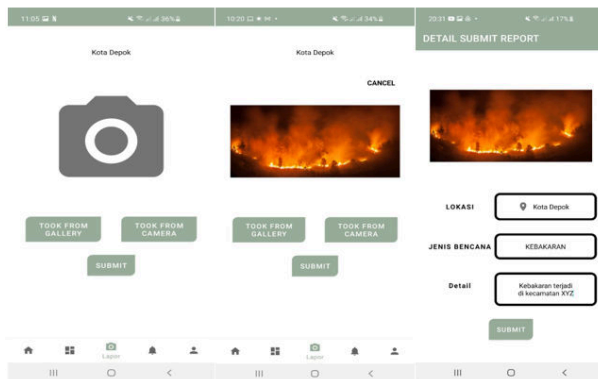


main.py, predict.py, saved model dan yang terakhir adalah requirements.txt.

App.yaml merupakan skrip yang berfungsi sebagai syarat pengaturan minimum seperti waktu penggunaan proses dan identifikasi pengenalan versi yang digunakan dalam App Engine. Kedua adalah main.py yang memiliki tugas menerima permintaan atau tanggapan ketika pengguna membuka alamat web. Ketiga adalah predict.py yaitu berkas yang berisikan baris kode untuk mendapatkan prediksi dari model. Keempat adalah requirements.txt, digunakan agar App Engine mengetahui kebutuhan instalasi baik berupa versi dan jenis pustaka yang digunakan dalam pengembangan model ini. Terakhir saved model .h5 adalah model latih yang sudah disimpan sebelumnya sebagai bentuk mempermudah komputasi dalam App Engine.

C. Implementasi Model pada Perangkat Android

Dari Gambar 8 menunjukkan tampilan dari proses melakukan prediksi yang ada dalam aplikasi ANCANA.



Gambar 8. Hasil prediksi model dalam Perangkat Android [4]

Hal pertama yang perlu dilakukan adalah memastikan gambar ingin diambil melalui kamera atau melalui galeri. Ketika sudah dipastikan pilihannya, maka langkah selanjutnya adalah menekan tombol submit yang ada. Hasil dari prediksi bencana alam pada contoh tersebut berhasil ditampilkan berupa kebakaran.

IV. KESIMPULAN [6]

Secara keseluruhan dari hasil pengujian yang telah dilakukan, model dengan arsitektur CNN berhasil melakukan pengklasifikasian bencana alam dengan empat kelas yang ada yakni gempa bumi, angin topan, banjir dan juga kebakaran hutan. Model latih memiliki ukuran 5,1 MB yang mengindikasikan bahwa model tersebut terhitung berukuran kecil. Hal ini diketahui karena model transfer learning umumnya memiliki ukuran berkisar 4 MB hingga ~700MB. Keberhasilan dari keseluruhan komputasi untuk prediksi model dijalankan dengan bantuan App Engine yang nantinya dihasilkan bentuk string json untuk keluaran label nya, sehingga mempermudah kinerja telepon genggam dalam bekerja. Minimal spesifikasi untuk menggunakan

aplikasi ini dimulai dari Android 5 (Lollipop). Dalam pengujian parameter dengan skenario yang berbeda, dapat disimpulkan bahwa penggunaan max pooling dengan jumlah yang lebih banyak dapat membantu pembelajaran model lebih baik dalam kasus klasifikasi bencana alam. Dropout terbaik yang digunakan yaitu dengan persentase sebesar 90% juga terbukti dapat mengurangi jumlah parameter dan juga nilai loss yang ada dalam suatu model. Penggunaan jumlah data sebanyak 11.430 gambar yang dijadikan sebagai referensi untuk data pengujian dan data latih dinyatakan lebih mampu untuk memperbaiki pembelajaran model dengan mengurangi nilai loss sebesar ~15% dibandingkan dengan penggunaan jumlah data sebanyak 4.428 gambar. Untuk hasil model terbaik didapatkan nilai akurasi sebesar 91,62%, F1 Score 92%, Recall 92%, dan Precision 92%.

DAFTAR PUSTAKA

- M. Gading Sadewo, A. Perdana Windarto, dan A. Wanto, "Penerapan Algoritma Clustering Dalam Mengelompokkan Banyaknya Desa/Kelurahan Menurut Upaya Antisipasi/Mitigasi Bencana Alam Menurut Provinsi Dengan K-MEANS," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 2018, [Daring]. Available: <http://ejournal.stmik-budidarma.ac.id/index.php/komik>
- H. Hadi, S. Agustina, dan A. Subhani, "Penguatan Kesiapsiagaan Stakeholder Dalam Pengurangan Risiko Bencana Gempabumi," *Geodika*, vol. 3, no. 1, hlm. 30–40, 2019.
- M. Aamir dkk., "Natural disasters intensity analysis and classification based on multispectral images using multi-layered deep convolutional neural network," *Sensors*, vol. 21, no. 8, Apr 2021, doi: 10.3390/s21082648.
- T. H. Hsieh dan J. F. Kiang, "Comparison of CNN algorithms on hyperspectral image classification in agricultural lands," *Sensors (Switzerland)*, vol. 20, no. 6, Mar 2020, doi: 10.3390/s20061734.
- Y. Sun, B. Xue, M. Zhang, G. G. Yen, dan J. Lv, "Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification," *IEEE Trans Cybern*, vol. 50, no. 9, hlm. 3840–3854, Sep 2020, doi: 10.1109/TCYB.2020.2983860.
- M. Anbarasan dkk., "Detection of flood disaster system based on IoT, big data and convolutional deep neural network," *Comput Commun*, vol. 150, hlm. 150–157, Jan 2020, doi: 10.1016/j.comcom.2019.11.022.
- D. R. Hartawan, T. W. Purboyo, dan C. Setianingsih, "Disaster Victims Detection System Using Convolutional Neural Network (CNN) Method," *Artificial Intelligence, and Communications Technology (IAICT)*, 2019.
- L. Ahmed, K. Ahmad, N. Said, B. Qolomany, J. Qadir, dan A. Al-Fuqaha, "Active Learning Based Federated Learning for Waste and Natural Disaster Image Classification," *IEEE Access*, vol. 8, hlm. 208518–208531, 2020, doi: 10.1109/ACCESS.2020.3038676.



- [9] W. S. Ahmed dan A. A. A. Karim, *The Impact of Filter Size and Number of Filters on Classification Accuracy in CNN*. 2020.
- [10] O. D. Gülgün dan Prof. Dr. H. EROL, "Classification Performance Comparisons Of Deep Learning Models In Pneumonia Diagnosis Using Chest X-Ray Images," *Turkish Journal of Engineering*, Jul 2019, doi: 10.31127/tuje.652358.
- [11] R. D. Nurfitra, "Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari Publikasi Ilmiah," 2018.
- [12] M. Dwarampudi dan N. V. S. Reddy, "Effects of padding on LSTMs and CNNs," Mar 2019, [Daring]. Available: <http://arxiv.org/abs/1903.07288>
- [13] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, dan N. A. Alajlan, "Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model with Attention," *IEEE Access*, vol. 9, hlm. 14078–14094, 2021, doi: 10.1109/ACCESS.2021.3051085.
- M. Yang, M. K. Lim, Y. Qu, X. Li, dan D. Ni, "Deep neural networks with L1 and L2 regularization for high dimensional corporate credit risk prediction," *Expert Syst Appl*, vol. 213, hlm. 118873, Mar 2023, doi: 10.1016/J.ESWA.2022.118873.
- I. Vasilev, D. Slater, G. Spacagna, P. Roelants, dan V. Zocca, *Deep Learning with Python*, Second Edi. Birmingham: Packt Publishing Ltd., 2019. doi: 10.1007/978-1-4842-2766-4.
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, dan L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, hlm. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- M. Tripathi, "Analysis of Convolutional Neural Network based Image Classification Techniques," *Journal of Innovative Image Processing*, vol. 3, no. 2, hlm. 100–117, Jun 2021, doi: 10.36548/jiip.2021.2.003.



©2023 This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).