

PEMANFAATAN CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI CITRA AWAN MENGGUNAKAN RASPBERRY PI

Rahdriawan Prisma; Endang Wahyu pamungkas

Program Studi Teknik Informatika, Fakultas Komunikasi dan Infomatika, Universitas Muhammadiyah Surakarta

Abstrak

Musim hujan dapat menimbulkan permasalahan seperti kelupaan terhadap barang yang dijemur. Penelitian ini bertujuan untuk membangun model pengenalan gambar awan hujan dengan memanfaatkan visi komputer dan menerapkan teknologi CNN dan sistem tanam *Raspberry Pi*. Aspek penting dalam penelitian ini adalah klasifikasi awan, dengan menerapkan CNN proses pemilihan fitur akan dilakukan secara oleh computer dan tidak memerlukan campur tangan manusia. *Dataset* citra awan menggunakan *Multi-Class Weather Dataset* yang diambil dari platform Kaggle. Dalam membangun arsitektur akan menggunakan arsitektur *EfficientNet B3* untuk mengeluarkan dan menghasilkan fitur dari gambar, dan memakai layer *Fully Connected Layer* yang dibuat terpisah mengikuti spesifikasi *Dataset* yang digunakan pada penelitian ini. Model CNN yang telah dilatih diterapkan pada *Raspberry Pi* untuk pengujian dengan gambar di lapangan. Hasil evaluasi menunjukkan hasil akurasi model mencapai angka 96% pada data validasi dan loss berada di angka 0.557 di data yang sama. Perhitungan metrik *Confusion Matrix*, Ini setelah menunjukkan hasil evaluasi dan performa yang bagus, proses selanjutnya adalah memasang dan menjalankan model di perangkat tanam *Raspberry Pi*. Perangkat *Raspberry Pi* mampu menjalankan inferensi *Deep Learning* dengan baik dan mengeluarkan *output* berupa lampu LED untuk mengetahui hasil prediksi adalah hujan atau tidak, jika terdeteksi sebagai hujan maka Lampu LED hijau akan hidup dan sebaliknya jika tidak hujan LED merah menyala.

Kata Kunci: CNN, *EfficientNet B3*, Klasifikasi awan, *Raspberry Pi*

Abstract

The rainy season can cause problems such as forgetfulness of items that are dried in the sun. This research aims to build a rain cloud image recognition model by utilizing computer vision and applying CNN technology and the Raspberry Pi embedded system. An important aspect in this research is cloud classification, by applying CNN the feature selection process will be done by computer and does not require human intervention. The cloud image Dataset uses the Multi-Class Weather Dataset taken from the Kaggle platform. In building the architecture, we will use the EfficientNet B3 architecture to extract and generate features from images, and use a Fully Connected Layer that is made separately following the specifications of the Dataset used in this study. The CNN model that has been trained is applied to Raspberry Pi for testing with images in the field. The evaluation results show that the accuracy of the model reaches 96% on the validation data and the loss is 0.557 on the same data. Confusion Matrix metric calculation, This after showing good evaluation results and performance, the next process is to install and run the model on the Raspberry Pi planting device. The Raspberry Pi device is able to run Deep Learning inference properly and issue output in the form of LED lights to find out the prediction results are rain or not, if it is detected as rain then the green LED light will be on and vice versa if it is not raining the red LED is on.

Keyword: CNN, Classification Cloud, EfficientNet B3

1. PENDAHULUAN

Negara Indonesia terkenal dengan iklim tropisnya yang hanya memiliki 2 musim selama satu kali revolusi bumi yaitu musim hujan dan musim kemarau. Dibidang pertanian para petani padi sangat bergantung pada musim hujan atau biasa disebut penanaman *off-season* yang dilakukan diluar musim kemarau, Teknik penanaman seperti ini diharapkan mencegah fluktuasi harga dan produksi cabai (Arifin, Listiana, & Jaya, 2022). Penelitian sebelumnya telah melakukan solusi untuk menjaga tanaman cabai dengan cara membuat pupuk yang memenuhi kebutuhan unsur hara tanaman, sehingga pada penelitian ini akan menggunakan teknologi kecerdasan buatan dan *embedded system* yaitu *Convolutional Neural Network* dan *Raspberry Pi* yang akan menutup dan melindungi lahan pertanian cabai dari hujan jika mendeteksi ada cuaca hujan dengan pengolahan citra.

Pada penelitian sebelumnya tentang penggunaan teknologi *Internet Of Things* untuk membuat kalibrasi sensor curah hujan piezoelektrik berbiaya rendah melalui pembelajaran mesin yang memerlukan modul sensor hujan untuk mengukur tingkat presipitasi hujan (Antonini, et al., 2022). Agar sensor hujan ini bekerja, memerlukan setetes air yang mengenai bagian atas dari sensor, artinya ada beban berupa air yang dianggap sebagai hujan, menandakan tingkat presipitasi hujan tinggi. Sistem pada penelitian ini dapat menggantikan modul sensor hujan sebagai alat yang memantau menghadap atas dengan *Raspberry Pi* dengan kamera kecil yang sudah ditanamkan algoritma pendeteksi citra awan dengan *Convolution Neural Network*, dengan menggunakan sistem seperti ini diharapkan dapat membantu sistem dalam melakukan persiapan sistem untuk melakukan perhitungan tingkat presipitasi hujan dengan pengenalan awan hujan terlebih dahulu.

Masalah selanjutnya adalah tentang menjemur pakaian di musim hujan, terkadang lupa untuk mengambil jemuran tersebut karena sibuk bekerja atau saat sedang diluar rumah. Menjemur pakaian di dalam ruangan seperti di dalam rumah juga bukan merupakan solusi yang tepat, pakaian yang dijemur di dalam ruangan menyebabkan pengeringan menjadi kurang sempurna dan bau yang kurang sedap muncul (Firmansyah, 2015), ada lagi penduduk di tempat desa yang tempat menjemur pakaiannya berada agak jauh di luar rumah juga akan kerepotan jika telah turun hujan dan baru mengambil satu persatu pakaian yang dijemur. Permasalahan tersebut memang sepele, tetapi jika terlalu sering dan bila ada pakaian yang dibutuhkan di keesokan harinya, tentu akan menimbulkan masalah lain. Penelitian sebelumnya yang serupa untuk menangani masalah yang sama menggunakan sensor hujan dan automasi menggunakan aplikasi mobile. Hal tersebut masih membutuhkan interaksi dengan manusia, yang dimana

pengguna harus menekan tombol melalui aplikasi di gawai untuk dapat menghidupkan alat pengaman jemuran (Asmiddin, 2023). Akhirnya muncul pemikiran yang lain dari penelitian tersebut yaitu mengganti peran sensor dan sistem pengendali *mobile* dengan hanya menggunakan kamera dan kecerdasan buatan.

Kembali populernya cabang ilmu komputer kecerdasan buatan khususnya dibidang *Deep Learning* karena banyak produk *Generative AI* di internet yang menggunakan ilmu tersebut seperti *Chat GPT* dan *Bard*, model *generative AI* untuk gambar seperti *DALL-E*, *Stable Diffusion*. Pada penelitian ini menggabungkan *Internet of Things* dan teknologi *Deep Learning* menggunakan model jaringan syaraf dari *Convolution neural network* (Putra, 2016). Menggunakan metode *machine learning* sederhana seperti *Support Vector Machine* tidak memberikan computer kemampuan untuk mengklasifikasikan gambar dengan akurasi yang bahkan tidak mencapai 70%, karena data yang digunakan untuk pelatihan adalah gambar yang pola nya *heterogen* atau tidak serupa. CNN mengambil konsep kerja dari korteks visual dari manusia untuk mengenali citra, kata konvolusi dari namanya karena struktur jaringan saraf tersebut melakukan proses konvolusi dua dimensi terlebih dahulu sebelum masuk ke *Multilayer Perceptron*, itulah yang membuat model jaringan syaraf CNN bagus untuk *Dataset* citra. Salah satu juga yang menjadi motivasi dibalik penelitian ini adalah mendapatkan pengetahuan dari eksperimen yang memanfaatkan CNN atau kecerdasan buatan yang dapat menyelesaikan masalah yang sama yaitu membuat alat untuk mendeteksi hujan dengan menggantikan peran sensor sebagai perangkat pasif ke mesin yang cerdas pada penelitian yang dilakukan oleh Asmiddin (2023) sebelumnya.

1.1 Batasan Penelitian

Selama proses penelitian berlangsung menemukan beberapa Batasan yang membuat hasil dari penelitian kurang maksimal. Diantaranya sebagai berikut:

1. Batasan data pada analisis *Dataset* citra awan CCSN dan tambahan gambar untuk test data lapangan yang diambil dari kamera gawai yang berlokasi di Sragen, Karanganyar dan Surakarta. Keadaan awan saat pengambilan gambar adalah saat akan terjadi hujan, atau sudah tujun gerimis dengan intensitas rendah.
2. Penelitian ini membatasi penerapan arsitektur model CNN yang sudah terlatih dengan memanfaatkan *framework* Keras.
3. Penelitian ini dilakukan tidak menggunakan perangkat kamera pada sistem tanam *Raspberry Pi* sebagai sumber masukan citra awan, tetapi mengandalkan masukan

gambar secara manual berupa berkas gambar awan yang diperoleh dari pengambilan gambar dari gawai.

4. Penggunaan metrik ukur performa model adalah fungsi loss, Akurasi, *Confusion Matrix*, dan laporan klasifikasi yang berisi *Recall*, *Precision*, *F1-score*,
5. Jumlah total *Dataset* yang digunakan saat melakukan penelitian ini sebagai data latih adalah 1125 gambar.
6. Kode inferensi *Deep Learning* yang dijalankan di perangkat *Raspberry Pi*, dilakukan pada lingkungan laboratorium terkontrol, dan terisolasi dari jaringan nirkabel. Beberapa Tindakan yang telah dilakukan untuk menambah keamanan menjalankan kode inferensi, seperti: mengupdate system, mengetikkan *password* setiap masuk akun *super user*, mematikan *login as root* di SSH, mengatur *firewall* dan menggunakan alat *anti brute-force* seperti *Fail2ban*.

2. METODE

Penelitian ini menerapkan metode ORM (Metode penelitian online) untuk mendapatkan *Dataset* yang akan digunakan untuk membangun model klasifikasi. Tujuan penelitian ini adalah mengetahui performa dari teknologi *deep learning* dan visi komputer dalam mengidentifikasi awan hujan pada alat pengaman jemuran di penelitian yang dilakukan oleh Asmiddin (2023) dan mengimplementasikannya pada sistem tanam yang lebih cepat dan hemat daya yaitu *Raspberry Pi*. Langkah-langkah yang akan dilakukan dalam penelitian ini dapat dilihat pada diagram pada Gambar 1.



Gambar 1. Diagram Langkah Penelitian.

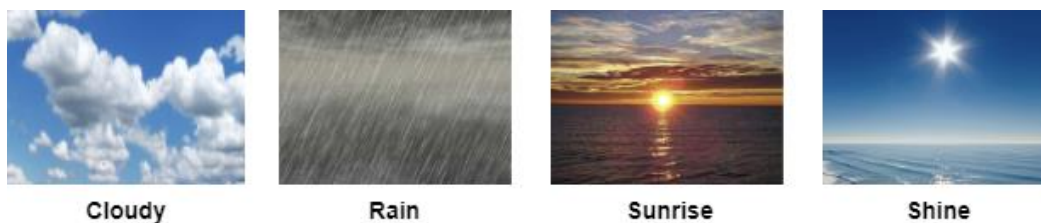
2.1 Dataset

Data yang digunakan pada penelitian ini adalah *Multi-class Weather Dataset Database* (Ajayi, 2018) yang diambil dari platform Kaggle. *Dataset* ini berisi Kumpulan citra awan yang dipotret dari luar ruangan dengan total 1,125 dalam berbagai ukuran dan berformat JPG. Gambar dikelompokkan menjadi 4 kategori berbeda: *Sunrise*, *Shine*, *Rain*, *Cloudy* seperti yang terlihat pada Tabel 1.

Tabel 1. Kategori Awan Pada *Dataset Multi-Class Weather*.

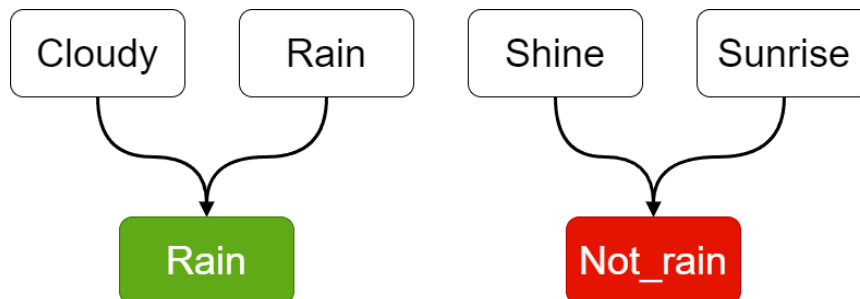
Kategori	Keterangan	Jumlah Gambar
<i>Sunrise</i>	<i>Alto cumulus</i>	357
<i>Shine</i>	<i>Strato cumulus</i>	253
<i>Rain</i>	<i>Nimbostratus</i>	215
<i>Cloudy</i>	<i>Cumulus</i>	300

Kategori *Sunrise* berupa gambar langit yang memiliki banyak awan, berwarna putih atau berwarna gelap seperti mendung, kemudian ada kategori *Rain* berupa gambar rintikan air hujan turun, kemudian kategori *Sunrise* berupa gambar saat matahari terbit dengan dominan warna jingga, yang terakhir kategori *Shine* berisi gambar sinar matahari dengan kondisi langit yang cerah atau berawan. Dapat dilihat pada Gambar 2.



Gambar 2. Sampel Dari *Dataset*.

Dataset Multi-class Weather pada penelitian ini akan dikelompokkan lagi menjadi 2 kategori yaitu “*Rain*” dan “*Not_Rain*” sebagai hasil prediksi dengan tujuan untuk menyamakan jumlah lampu LED pada tahap Penanaman model. Klasifikasi kategori hujan atau label “*Rain*” akan mengambil kelas *Rain* dan *Cloudy* (Ramadhan & Mulya, 2022), sedangkan kategori *Not_Rain* akan mengambil kelas *Sunrise* dan *Shine*, pembagian data yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar 3.



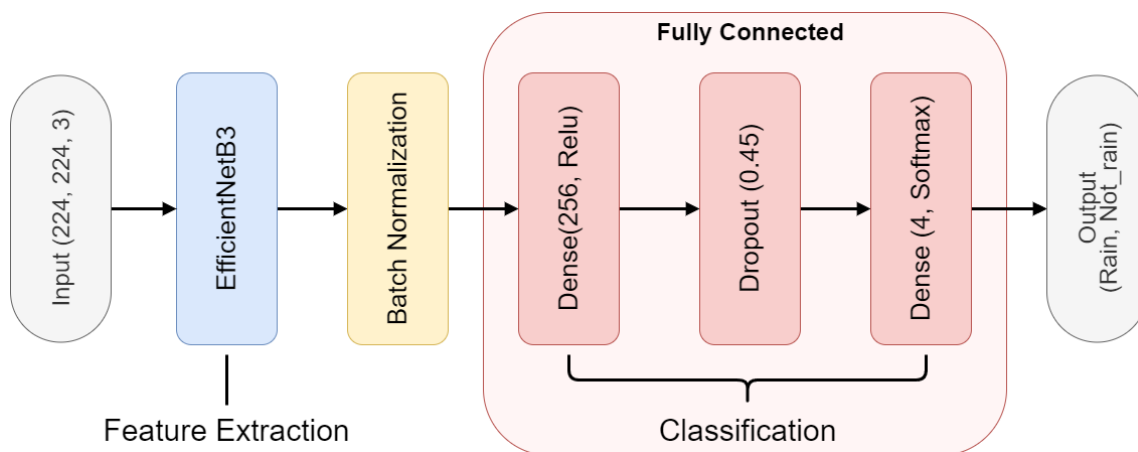
Gambar 3. Diagram Pembagian Kategori Label *Dataset* Menjadi Label Prediksi.

Kelas prediksi akhir menghasilkan 2 kategori saja karena berkaitan dengan awan mendung yang lebih dekat dengan label *Cloudy* pada *Dataset*, tidak mungkin menggunakan

label *Rain* karena model baru mendeteksi hujan dalam kondisi saat sudah turun hujan seperti sampel pada Gambar 1.2, maka gerbang hujan akan terlambat menutup dan melindungi dari basah terkena air hujan.

2.2 Membuat dan Melatih Model

Pada proses pembuatan model akan menggunakan bahasa pemrograman *Python* dan *Google Colab* sebagai alat untuk melatih model CNN. Model konvolusi dan syaraf buatan akan mengambil ide dari penelitian *Efficient Net: Rethinking Model Scaling for Convolutional Neural Networks* sebagai base model model dari *Efficient Net B3*, sebelum ke layer *Fully Connected* dilakukan normalisasi per batch, kemudian layer koneksi penuh yang akan terdiri dari dense layer pertama yang memiliki 256 *node*, *Dense Layer* kedua memiliki 4 *node* dengan fungsi aktivasi *Softmax*, kemudian pada *output layer* akan di program menggunakan percabangan *if*. Gambaran arsitektur dapat dilihat pada Gambar 4.



Gambar 4. Skema Arsitektur CNN Dengan Model Dasar Efficient Net B3.

Alur lebih lengkap dari Gambar 1.4 sebagai berikut:

2.2.1 Input

Dari skema arsitektur pada Gambar 1.3 dijelaskan input gambar memakai 3 kanal warna dari merah, hijau dan biru dan gambar akan diperkecil menjadi ukuran 224×224 pixel.

2.2.2 EfficientNetB3 Model

Lapisan ini adalah arsitektur dari *Efficient Net B3* yang berfungsi mengekstrak fitur dari gambar sebelum proses klasifikasi pada layer *Fully Connected*. Pada

penelitian sebelumnya yang dilakukan oleh Anggiratih (2021) untuk klasifikasi penyakit pada tanaman padi, model dengan arsitektur *Efficient Net B3* mendapatkan akurasi yang tinggi 99% terhadap arsitektur Mobilenet V3 yang disandingkan, Ini membuktikan bahwa *Efficient Net B3* adalah salah satu arsitektur CNN yang unggul dalam klasifikasi gambar. Pada penelitian yang sama dibahas juga mengenai arsitektur model *EfficientNet B3* memiliki lapisan dasar dan lapisan kompleks yang memiliki komposisi yang terdiri dari *Convolutional layer*, *Mobile Bottleneck Convolutional Layer*, *Pooling layer* (Anggiratih, Siswanti, Octaviani, & Sari, 2021). Parameter yang dikonfigurasi untuk penelitian ini adalah menggunakan bobot terlatih dari *ImageNet* dan untuk lapisan *Fully Connected* tidak menggunakan pengaturan bawaan tetapi menyesuaikan dengan penelitian ini yang akan dijelaskan pada tahap selanjutnya.

2.2.3 *Batch Normalization*

Lapisan ini berfungsi sebagai normalisasi nilai dengan menyamakan skala yang dilakukan pada sekumpulan data (*batch*). Pada tahapan ini nilai sebelum, tahapan ini memungkinkan proses pelatihan *Dataset* dapat menggunakan *Learning Rate* yang tinggi dapat menimbulkan permasalahan di bagian jaringan saraf seperti distribusi bobot yang tidak merata setelah nilai dihitung menggunakan fungsi aktivasi (Yao, Cao, Zheng, Huang, & Lin, 2021), atau dalam kasus penelitian ini adalah setelah keluar dari model dasar *Efficient Net B3*.

2.2.4 *Fully Connected Layer*

Lapisan koneksi penuh terdiri dari 2 *Dense Layer* dan *Dropout layer*. *Dense Layer* pertama memiliki 256 node dengan fungsi aktivasi *ReLU*. Kemudian dilanjutkan ke tahap *Dropout Layer*, Proses ini merupakan teknik regularisasi dengan cara membuat *node* dalam suatu lapisan tidak dapat menerima dan melakukan pelatihan pada data secara acak, *Dropout* memiliki peran penting dalam mengurangi *overfitting* yang sering terjadi pada model dengan jumlah data dan parameter yang besar (Ali & Mallaiah, 2022). Lapisan terakhir adalah *Dense Layer* dengan 4 node yang diberi aktivasi fungsi Softmax. Aktivasi softmax digunakan untuk menghitung probabilitas dari perkiraan model untuk

setiap kategorinya, Output berupa nilai probabilitas dari 0-1 untuk setiap kategori pada *Dataset*. Softmax dapat dihitung dengan fungsi persamaan berikut

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Dengan $e = \text{eksponen}$, $i = 1, \dots, K$ dan $z = (z_1, \dots, z_K) \in R^K$ (Denata, Rismawan, & Ruslianto, 2021).

2.2.5 Output

Pada bagian *output* menambahkan program percabangan terhadap hasil prediksi dari model. Jika hasil prediksi model masuk ke kategori hujan maka hasil *output* adalah *Rain* dan jika hasil prediksi model masuk kategori tidak hujan, maka output adalah *Not_Rain*. Kode program dapat dilihat pada Gambar 5.

```
if (classes2[0][0] == max(classes2[0]) or classes2[0][1] == max(classes2[0])):
    print("Prediction: Rain")
else:
    print("Prediction: Not_rain")

1/1 [=====] - 0s 87ms/step
[[0.04342393 0.8640811 0.06379646 0.02869855]]
Prediction: Rain
```

Gambar 5. Percabangan Pada Output Prediksi.

Konvolusi pertama menghasilkan 96 fitur baru dan dengan kernel 11 x 11, ukuran gambar berkurang menjadi 55x55. Masuk ke proses *pooling* yang bertujuan untuk menurunkan sampel dari pemetaan fitur dari proses konvolusi dan mempelajari fitur yang didapat dari gambar berskala besar. Proses ini adalah agregasi dari setiap wilayah dari fitur hasil konvolusi, metode agregasi dapat mengambil rata-rata atau nilai maksimal dari wilayah *pooling* (Nirthika, Manivannan, Ramanan, & Wang, 2022). Penelitian ini akan menggunakan metode *Max Pooling*. Selesai blok konvolusi pertama, selanjutnya gambar akan diproses di blok konvolusi selanjutnya dengan cara yang sama, menghasilkan fitur sebanyak 256 dengan ukuran gambar 27x27. Blok konvolusi ke-3 tanpa melakukan *pooling* menghasilkan fitur sebanyak 384 fitur dengan ukuran gambar 13x13 *pixel*. Blok konvolusi terakhir adalah sama dengan Langkah konvolusi pertama dan pooling yang menghasilkan fitur terakhir sebanyak 256 dengan ukuran gambar 6x6 yang kemudian akan di *flatten* atau dijadikan data 1 dimensi.

Data 1 dimensi yang telah di *flatten* tadi di proses ke *Fully Connected Layer* yang akan memiliki 2 lapisan tersembunyi. Lapisan tersembunyi pertama berjumlah 9216 node, memberikan aktivasi *ReLU* pada setiap bobot dari *node* sebelum melakukan *dropout* sebesar 0.5%. Aktivasi fungsi *ReLU* atau *Rectified Linear Unit* bukan termasuk *non-linear*, dengan formula sebagai berikut:

$$\begin{aligned} \text{ReLU}(x) &= \max(0, x) && (\text{Akil, 2023}) \\ &= \begin{cases} x, & x > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

Keterangan:

x = input neuron

2.3 Evaluasi Model

Setelah melakukan proses latihan pada model CNN, tahapan berikutnya adalah melakukan tes performa pada data tes. Metode yang digunakan untuk mengukur akurasi pada tes adalah fungsi *Cross-Entropy Loss* dan *Accuracy*. *Loss function* adalah pengukuran yang digunakan pada jaringan syaraf buatan untuk menemukan perbedaan antara hasil probabilitas yang diprediksi oleh jaringan syaraf cerdas dengan label kebenaran (Noel, Banerjee, D, & Muthiah-Nakarajan, 2023), persamaan *Cross-Entropy Loss* dapat dilihat seperti dibawah ini

$$H(p, q) = - \sum p(x) \log q(x)$$

Dengan p adalah probabilitas dari label, dan q adalah probabilitas dari model.¹ Pada penelitian ini akan menggunakan *loss function Categorical Crossentropy* karena klasifikasi menggunakan 4 label.

2.5 Penanaman Model

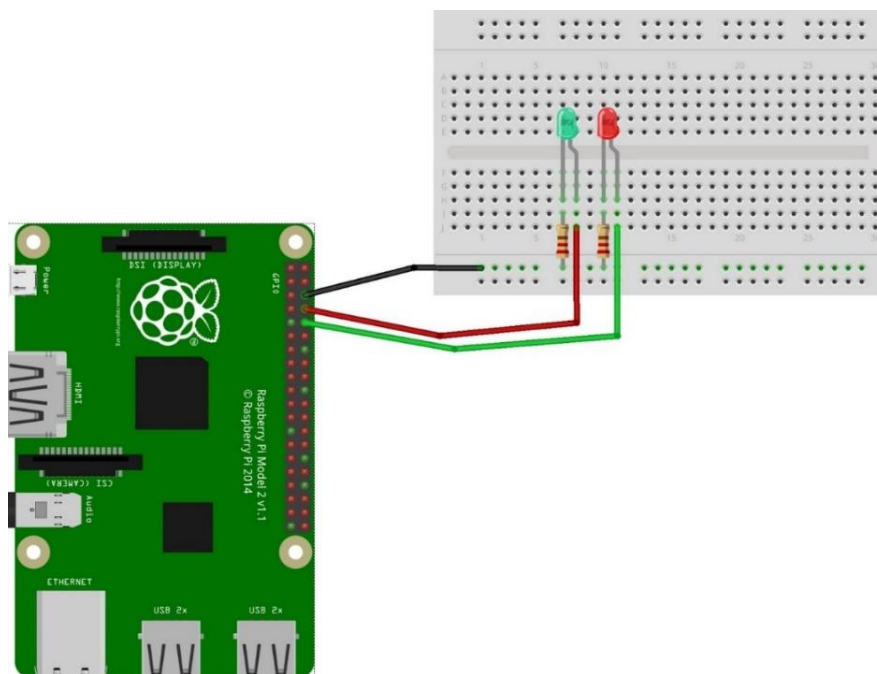
Penggunaan model *TensorFlow Lite* dimaksudkan agar dapat berjalan pada perangkat tepi, seperti pada penelitian *The Development of Android for Indonesian Sign Language Using Tensorflow Lite and CNN: An Initial Study* yang menjalankan deteksi bahasa isyarat untuk bahasa Indonesia di perangkat Android (Fadlilah, Mahamad, & Handaga, 2021). Proses penanaman adalah proses terakhir yang dilakukan dalam penelitian ini yaitu menjalankan model yang telah dilatih ke perangkat *Raspberry Pi*. Dalam tahap ini model yang sudah

¹ <https://www.v7labs.com/blog/cross-entropy-loss-guide>, diakses pada tanggal 3 Juni 2024

dievaluasi akan dikonversi menjadi format TFLite yang dapat dijalankan di perangkat sistem tanam seperti *Raspberry Pi*. Penanaman model akan dilakukan dengan tahapan berikut:

1. Memperbarui sistem pada *Raspberry Pi*.
2. Memasang modul *TensorFlow*, *Numpy*, *Pandas*, *Matplotlib* dan *Seaborn* untuk *Python*.
3. Mempersiapkan model dan melakukan inferensi dari masukan berupa berkas citra awan.

Untuk mengetahui hasil dari prediksi hujan atau tidak hujan, adalah dengan menambahkan lampu LED 2 warna, hijau dan merah. 2 LED akan disambungkan di GPIO pada *Raspberry Pi*, dengan menggunakan *breadboard* yang terlihat pada gambar 6.



Gambar 6. Diagram Kabel Pada Raspberry Pi.

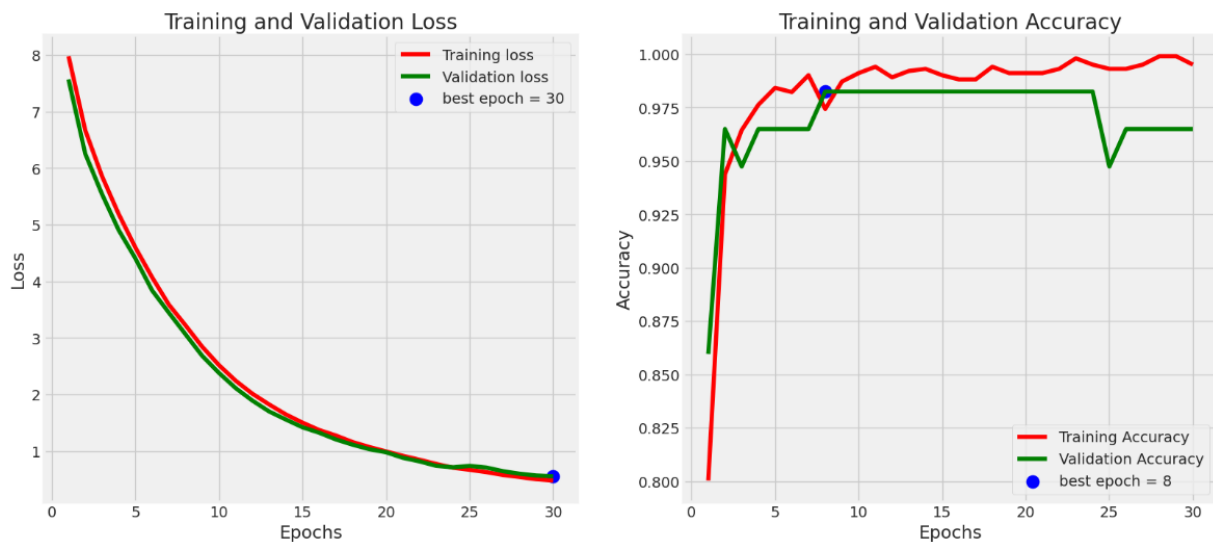
Perangkat *Raspberry Pi* pada Gambar 1.5 akan dioperasikan menggunakan antarmuka baris perintah, saat masukan gambar dibaca oleh perangkat *Raspberry Pi*, model akan melakukan tugas mengenali gambar dan program akan menhidupkan LED berwarna hijau jika gambar terprediksi kategori *Rain* oleh model atau bisa dibilang positif hujan, dan LED berwarna merah menyala jika gambar terprediksi kategori *Not_Rain*.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini penambahan *Dataset* yang diperlukan untuk melakukan test data lapangan diambil dari gawai dengan tujuan untuk menyesuaikan dengan wilayah Indonesia khususnya Jawa Tengah, lebih tepatnya pada daerah Kota Sragen dan Kota Surakarta. Gambar dari gawai diambil dengan cara mengarahkan kamera ke atas dalam jangkauan sudut 30-90 derajat, dalam kondisi awan sudah diketahui akan terjadi hujan, atau saat gerimis.

3.1 Pembuatan dan Evaluasi Model

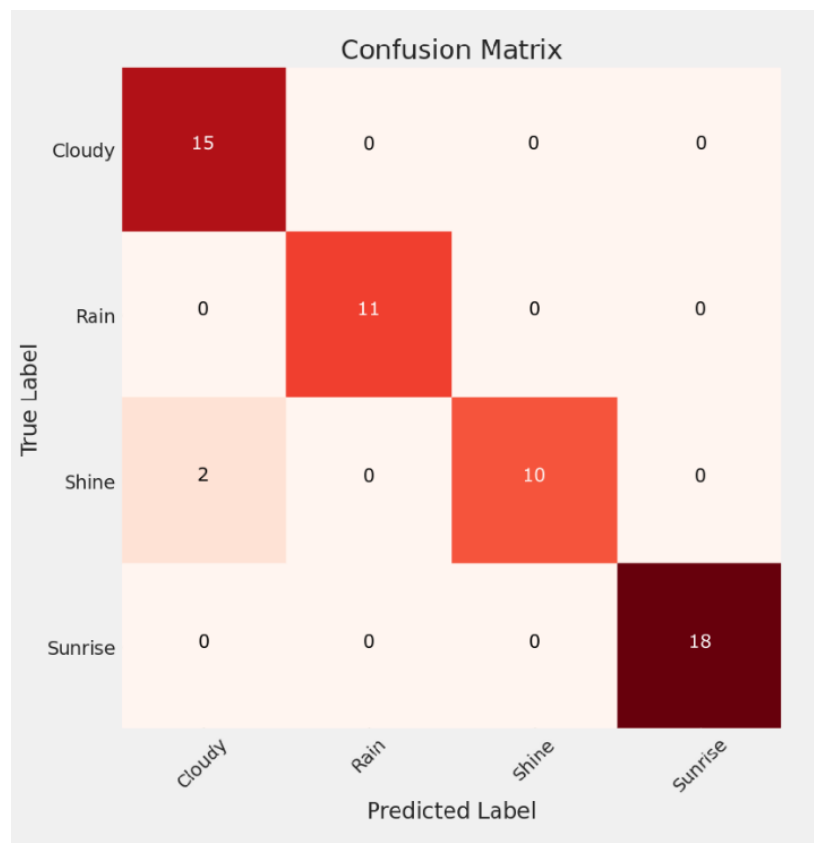
Proses pelatihan model dilakukan sesuai dengan spesifikasi yang tertulis di metodologi dan menghasilkan performa yang cukup baik. Pelatihan model dilakukan menggunakan platform *Google Colab* dengan spesifikasi RAM 12.7 *GigaByte*, menggunakan akselerasi *hardware* jenis GPU T4 dengan RAM sebesar 16 *GigaByte*, penulisan kode program *Python* dibantu dengan *framework TensorFlow* untuk memudahkan dan mempercepat pembuatan arsitektur model, versi *TensorFlow* yang digunakan saat eksperimen berlangsung adalah 2.15.0. pada pelatihan data dilakukan sebanyak 30 *epoch* (atau 1 siklus pelatihan total data latih) dan menggunakan *callback Early Stopping*, setelah melakukan pelatihan kemudian menampilkan perhitungan loss atau error dan akurasi model pada data train dan validasi yang didapat selama proses pelatihan, sehingga didapat informasi seperti pada Gambar 7.



Gambar 7. Diagram *Loss* dan *Accuracy*.

Pada 1.6 diperlihatkan bahwa akurasi pada data train terbaik adalah 1.0 atau 100%, akurasi pada data validasi adalah 0.9649122953414917 atau 96% dengan *epoch* terbaik pada

epoch ke-30, *Loss* pada data *train* terbaik adalah 0.422505646944046, dan *Loss* terbaik pada data validasi adalah 0.5571275949478149, hal ini menunjukkan bahwa kinerja model sudah bagus. Untuk benar-benar membuktikan bahwa model memiliki kinerja bagus akan menggunakan pengukuran metrik *Confusion Matrix*. *Confusion matrix* adalah informasi yang diperoleh dari pengukuran evaluasi dengan menghitung persentase prediksi benar, dan prediksi salah (Anggoro & Mukti, 2021). Diagram *confusion matrix* dapat dilihat pada Gambar 8.



Gambar 8. Hasil Confusion Matrix.

Dari diagram *confusion matrix* yang dilakukan pada data test menunjukkan bahwa kategori *Shine* terdapat kesalahan prediksi yaitu ada 2 gambar yang dianggap oleh model sebagai *Cloudy*, pada kategori lainnya terprediksi dengan benar dan tidak ada kesalahan tafsir oleh model CNN, yang menandakan model sudah cukup bagus. Dari hasil *confusion matrix* dapat dihitung laporan klasifikasi yang mengandung metrik pengukuran lain seperti *Precision*, *Recall*, *F1-score* yang dapat dilihat pada Tabel 2.

Tabel 2. Hasil Perhitungan *Precision*, *Recall*, *F1-Score*

Kategori	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Support</i>
<i>Cloudy</i>	0.88	1.00	0.94	15

<i>Rain</i>	1.00	1.00	1.00	11
<i>Shine</i>	1.00	0.83	0.91	12
<i>Sunrise</i>	1.00	1.00	1.00	18

Pada penelitian ini model yang dibuat tidak boleh terlalu banyak salah prediksi pada kategori *Cloudy* dan *Rain*, maka hasil dari laporan klasifikasi pada tabel 2 yang paling dilihat adalah *Recall*. *Recall* adalah banyaknya sampel yang diprediksi benar oleh model atau bisa disebut Tingkat Positif Sebenarnya, nilai *Recall* didapatkan dari rumus dibawah bersumber dari *confusion matrix* (Chen, et al., 2023).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + True\ Negative}$$

Pada kategori *Cloudy* dan *Rain* memiliki nilai *Recall* masing-masing sebesar 1.00 atau 100% yang menunjukkan tidak ada *False Negative* pada data test di kategori tersebut. Setelah berhasil melakukan pelatihan pada *Dataset*, model siap di dijalankan pada *Raspberry Pi*.

Nilai *F1-Score* didapatkan dari perhitungan rata-rata dari hasil Presisi dan nilai hasil *Recall*, semakin tinggi nilai *F1-Score* menunjukkan hasil klasifikasi model semakin efektif. Rumus *F1-Score* ditulis seperti berikut.

$$F1 - Score = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$$

Nilai akurasi dapat diperoleh dari melakukan perhitungan total jumlah benar prediksi dibagi dengan total jumlah data, atau dapat ditulis dengan persamaan berikut ini

$$Accuraccy = \frac{True\ positive + True\ Negative}{True\ positive + True\ Negative + False\ positive + False\ Negative}$$

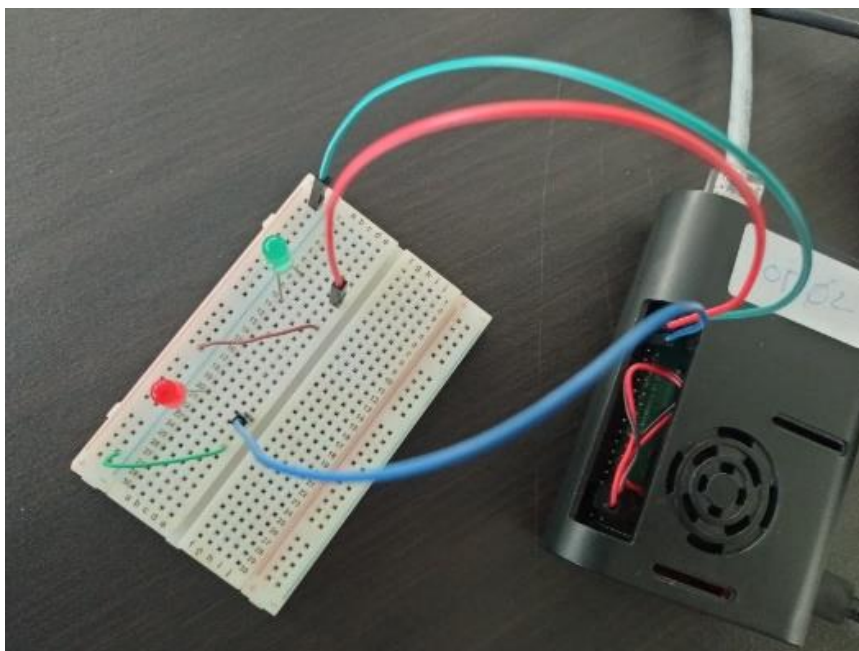
3.2 Penanaman Model Ke *Raspberry Pi*

Sebelum melakukan penanaman model ke perangkat *Raspberry Pi*, model harus dikonversi menjadi format *TFLite*. Konversi model dilakukan dengan tujuan mengubah format model seperti pada Gambar 9, yang bisa dibaca oleh sistem tanam seperti *Raspberry Pi*, konversi model juga memberikan keringanan beban kerja karena ukuran model juga akan berkurang.

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
open("/content/drive/MyDrive/DATASET/Trained_model/ccsn_v4.tflite", "wb").write(tflite_model)
```

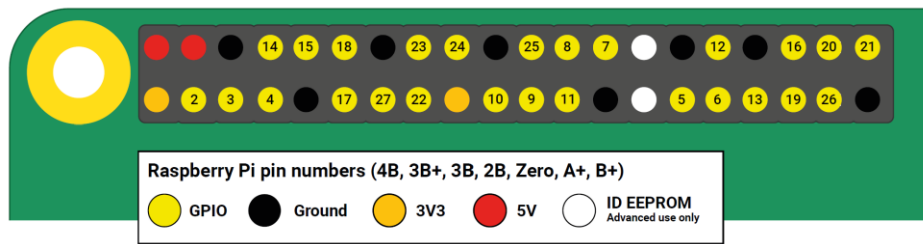
Gambar 9. Konversi model menjadi format *TFLite*.

Percobaan yang dilakukan pada penelitian ini menggunakan perangkat *Raspberry Pi 4 B* dengan RAM 4 *GigaByte*. Membuat rangkaian seperti pada gambar 1.5, sehingga terlihat pada Gambar 10.



Gambar 10. Rangkaian Pada Alat Percobaan Deteksi Hujan Dengan *Raspberry Pi*.

Ada 2 lampu LED berwarna merah dan hijau yang masing-masing anoda terhubung dengan *Ground*, dan katoda terhubung dengan pin GPIO 16 untuk LED merah dan pin GPIO 20 untuk LED hijau, pin yang dipakai ditandai dengan kotak warna merah pada Gambar 1.10.



Gambar 11. Denah Pin Pada *Raspberry Pi* 4 B.

Selesai membuat rangkaian, memindahkan semua berkas gambar yang digunakan untuk prediksi, berkas model berformat TFLite, berkas naskah kode *Python* untuk menjalankan inferensi model. Proses inferensi dilakukan dengan mengetikkan perintah pada aplikasi terminal di perangkat *Raspberry Pi* yang terlihat pada Gambar 11.

```
pi1337@raspberrypi:~/CCSN_TEST $ python3 rain_predict.py sampleN.jpg
Using tensorflow version 2.16.1
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

Prediction: Not_Rain

Initialize to open rain gate
```

Gambar 12. Inferensi Model Pada Perangkat *Raspberry Pi* 4.

Program dijalankan dengan memanggil *python3* diikuti naskah kode untuk inferensi model, naskah kode menerima 1 argumen berupa gambar. Pada gambar 12 hasil prediksi dari masukan gambar menghasilkan informasi bahwa model memprediksi gambar masuk kategori tidak hujan atau label “*Not_Rain*” dan lampu LED warna merah menyala, tulisan “*Initialize to open rain gate*” hanya sebagai perumpamaan jika menerapkan gerbang pelindung hujan.

4. PENUTUP

Selama melakukan penelitian ini, menunjukkan hasil kinerja model *Convolutional Neural Network* sudah akurat dalam mengenali citra gambar awan hujan pada *Dataset Multi-class Weather Dataset* dan *Dataset* tes lapangan yang diambil sendiri. Hasil akurasi mencapai angka 0.9649122953414917 atau 96% pada data validasi dan fungsi *loss* mencapai 0.5571275949478149 pada *Dataset* validasi yang menunjukkan bahwa performa model baik, dan berdasarkan perhitungan metrik performa *confusion matrix* menunjukkan bahwa pada data test di kategori *Rain*, *Sunrise* dan *Cloudy* dapat benar diprediksi oleh model, tetapi pada

kategori *Shine* ada kesalahan prediksi. Dalam penelitian ini juga menunjukkan model *TensorFlow Lite* yang mampu berjalan di perangkat *Raspberry Pi 4* yang memiliki daya rendah dan melakukan inferensi secara lokal terhadap masukan citra awan. Penerapan CNN dalam belum bisa memberikan pendeteksian dengan akurasi yang sangat tinggi tetapi dapat membantu dalam menyelesaikan masalah klasifikasi citra awan, melalui eksperimen ini dapat memberikan landasan yang kuat kedepannya untuk penerapan teknologi kecerdasan *Deep Learning* dan *Computer Vision* pada sistem tanam seperti *Raspberry Pi*, bahkan sistem tanam yang lebih rendah daya.

Penelitian selanjutnya yang berkaitan dengan topik yang sama, menggunakan *Dataset* citra awan yang lebih banyak dan relevan dengan wilayah yang sesuai dengan lingkungan penelitian. Menerapkan arsitektur CNN yang lainnya atau metode *Deep Learning* yang berbeda atau bahkan mencoba metode transfer-learning yang lebih menarik. Menambahkan perangkat keras berupa kamera untuk menyempurnakan proses masukan gambar secara langsung ke model, atau menambahkan perangkat sensor untuk membantu alat pendeteksi hujan agar dapat menambah fungsional dan membantu model untuk melakukan aksi terhadap datangnya hujan dengan waktu yang lebih tepat.

DAFTAR PUSTAKA

- Ajayi, G. (2018). Multi-class Weather *Dataset* for Image Classification. Mendeley Data. doi:10.17632/4drtyfjtfy.1
- Akil, I. (2023). Komparasi Fungsi Aktivasi Neural Network Pada Data Time Series. *INTI Nusa Mandiri*, 18(1), Page 78-83. doi:10.33480/inti.v18i1.4288
- Ali, A. A., & Mallaiah, S. (2022). Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout. *Journal of King Saud University - Computer and Information Sciences*, Volume 34(Issue 6), Pages 3294-3300. doi:doi.org/10.1016/j.jksuci.2021.01.012
- Anggiratih, E., Siswanti, S., Octaviani, S. K., & Sari, A. (2021). Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning Efficient Net B3 Dengan Transfer Learning. *Jurnal Ilmiah Sinus (JIS)*, 19(1). doi:https://doi.org/10.30646/sinus.v19i1.526
- Anggoro, D. A., & Mukti, S. S. (2021). Performance Comparison of Grid Search and Random Search Methods for. *International Journal of Intelligent Engineering and Systems*, 14(6), 198-207. doi:10.22266/ijies2021.1231.19
- Antonini, A., Melani, S., Mazza, A., Baldini, L., Adirosi, E., & Ortolani, A. (2022). Development and Calibration of a Low-Cost, Piezoelectric Rainfall Sensor through Machine Learning.

- Arifin, M. Z., Listiana, B. E., & Jaya, K. D. (2022). Pengaruh Beberapa Pupuk Daun Terhadap Pertumbuhan Dan Hasil Tanaman Cabai Rawit Yang Ditanaman Di Luar Musim. *Jurnal Ilmiah Mahasiswa AGROKOMPLEK*, 1(2), 132-139. doi:10.29303/jima.v1i2.1433
- Asmaddin, A. M. (2023). Rancang Bangun Jemuran Pakaian Otomatis Berbasis Internet of Things. *Jurnal Informatika*, 12, 50-59. doi:https://doi.org/10.55340/jiu.v12i1.1309
- Bochkovskiy, A., Wang, C. Y., & Liao, H. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*, 10. doi:https://doi.org/10.48550/arXiv.2004.10934
- Chen, H., Wang, N., Du, X., Mei, K., Zhou, Y., & Cai, G. (2023). Classification prediction of breast cancer based on machine learning. *Computational intelligence and neuroscience*, 2023, 6530719. doi:https://doi.org/10.1155%2F2023%2F6530719
- Denata, I., Rismawan, T., & Ruslianto, I. (2021). Implementation of Deep Learning for Classification Type of Orange Using. *Telematika: Jurnal Informatika dan Teknologi Informasi*, 297-307.
- Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503, 92-108. doi:https://doi.org/10.1016/j.neucom.2022.06.111
- Fadlilah, U., Mahamad, A. K., & Handaga, B. (2021). The Development of Android for Indonesian Sign Language Using Tensorflow Lite and CNN: An initial Study. *Journal of Physics: Conference Series*, 1858(1), 012085. doi:10.1088
- Firmansyah, R. (2015). Perancangan Fasilitas Pengerian Pakaian Saat Turun Hujan(Studi Kasus Wilayah Bandung-kampung Lengkong). *eProceedings of Art & Design*, II, 1284.
- Nadir, R. A., & Sukmana, R. N. (2023). Sistem Prediksi Harga Emas Berdasarkan Data Time Series Menggunakan Metode Artificial Neural Network (ANN). *Digital Transformation Technology*, 3(2), 426--437.
- Nirthika, R., Manivannan, S., Ramanan, A., & Wang, R. (2022). Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. *Neural Computing and Applications*, 34(34), 5321-5347. doi:https://doi.org/10.1007%2Fs00521-022-06953-8
- Noel, M. M., Banerjee, A., D, G. B., & Muthiah-Nakarajan, V. (2023). Alternate Loss Functions for Classification and Robust. 2. doi:10.48550/arXiv.2303.09935
- Putra, W. S. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101. *Jurnal Teknik ITS*, V(1). doi:http://dx.doi.org/10.12962/j23373539.v5i1.15696

- Ramadhan, I. A., & Mulya, A. (2022). Estimasi Curah Hujan Menggunakan Metode Convective Stratiform Technique(Cst) Dan Modified Convective Stratiform Terchnique(Mcst) (Studi Kasus: Jombang, 2 Januari 2021). *II*(1), 207. Retrieved from <https://ejournal.unibabwi.ac.id/index.php/knmipa/article/view/1752>
- Tan, M., & Le, Q. (2019). Efficient Net: Rethinking model scaling for convolutional neural networks. *International conference on machine learning*, 6015-6114. doi:<http://doi.org/10.48550>
- Yamashita, T., & Iwabuchi, H. (2022). Diurnal variations of different types of cloud over the Baiu–Meiyu frontal zone using retrieved cloud properties: Implication for the rainfall process. *Atmospheric Research*, 271, 2. doi:<https://doi.org/10.1016/j.atmosres.2022.106139>
- Yao, Z., Cao, Y., Zheng, S., Huang, G., & Lin, S. (2021). Cross-Iteration Batch Normalization. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12331-12340. doi:<https://doi.org/10.1109/CVPR46437.2021.01215>
- Zain, Sirodj, M. N., & Nuriani, D. A. (2022). Algoritma Artificial Neural Network dalam Klasifikasi Chest X-Rays Pasien COVID-19. *Jurnal Riset Statistika*, 2(2), 137--144. doi:[10.29313/jrs.v2i2.1426](https://doi.org/10.29313/jrs.v2i2.1426)
- Zhang, J., Liu, P., Zhang, F., & Song, Q. (2018). CloudNet: Ground-Based Cloud Classification With Deep Convolutional Neural Network. *Geophysical Research Letters*, 45(16), 8667. doi:<https://doi.org/10.1029/2018GL077787>