

ANALISIS OPTIMASI KLASIFIKASI CITRA AWAN BERDASARKAN NILAI HYPERPARAMETER PADA *TEACHABLE MACHINE* UNTUK PENGEMBANGAN APLIKASI *WEB MOBILE*

Muhammad Indra Bendi^{*1}, Edwin Ariesto Umbu Malahina²

¹BMKG Stasiun Klimatologi Nusa Tenggara Timur, Kota Kupang, ²STIKOM Uyelindo Kupang, Kota Kupang
Email: ¹muhammad.indra@bmkg.go.id, ²edwinariesto@gmail.com

^{*}Penulis Korespondensi

(Naskah masuk: 14 Desember 2024, diterima untuk diterbitkan: 09 Desember 2025)

Abstrak

Pengamatan cuaca menjadi aspek penting dalam berbagai bidang seperti meteorologi, penelitian lingkungan dan penerbangan. Identifikasi jenis awan memainkan peran kunci dalam memprediksi perubahan cuaca dan mengevaluasi dampak lingkungan. Tujuan dari penelitian ini adalah untuk mengembangkan sebuah aplikasi *web mobile* sistem cerdas yang mampu membantu masyarakat dalam mendeteksi jenis awan secara mandiri di sekitar, memberikan edukasi tentang jenis awan dan yang paling penting adalah mencari nilai optimasi *hyperparameter epoch*, *batch size* dan *learning rate* dalam *Teachable Machine*. Penelitian ini menggunakan nilai untuk parameter-parameter yang diteliti, yaitu nilai *epoch* yang bervariasi antara 10, 50, 100, 250, 750 dan 1000. Kemudian nilai *batch size* yang bervariasi antara 16, 32, 64, 128, 256 dan 512 serta *learning rate* yang bervariasi antara 0,00001; 0,0001; 0,001; 0,01; 0,1 dan 1. Total *dataset* sebanyak 4.000 sampel data latih (400 sampel dalam 10 kelas) digunakan dalam *Teachable Machine*. Metode yang digunakan adalah dengan memanfaatkan *framework TensorFlow* pada layanan *Teachable Machine* untuk melatih data citra atau gambar. *Framework* ini menyediakan algoritma *Convolutional Neural Networks (CNN)* yang dapat melakukan klasifikasi citra atau gambar dengan tingkat akurasi yang tinggi. Hasil pengujian menunjukkan bahwa nilai optimal tertinggi tercapai pada nilai *epoch* ke-50, dengan nilai *batch size* 16 dan *learning rate* 0,00001 yang menghasilkan tingkat akurasi antara 70% hingga 98%. Aplikasi *web mobile* ini diharapkan dapat diimplementasikan secara luas untuk kepentingan masyarakat agar mengenali jenis awan yang menyebabkan potensi hujan.

Kata kunci: *teachable machine, deteksi objek, jenis awan, convolutional neural networks, optimalisasi citra*

OPTIMIZATION ANALYSIS OF CLOUD IMAGE CLASSIFICATION BASED ON HYPERPARAMETER VALUE ON *TEACHABLE MACHINE* FOR MOBILE WEB APPLICATION DEVELOPMENT

Abstract

Weather observation is becoming an increasingly important aspect in various fields, such as Meteorology, Environmental Research, and aviation. The identification of cloud types plays a key role in predicting weather changes and evaluating environmental impacts. The purpose of this study is to develop a mobile web application intelligent system that is able to help people detect the type of cloud independently around, provide education about the type of cloud, and most importantly, find the value of optimization hyperparameter epoch, batch size and learning rate in *Teachable Machine*. This study uses the values for the parameters studied, namely the epoch values that vary between 10, 50, 100, 250, 750, and 1000. Then the value of batch size varies between 16, 32, 64, 128, 256, and 512, and the learning rate varies between 0.00001; 0.0001; 0.001; 0.01; 0.1, and 1. A total of 4,000 training data samples (400 samples per class) were used in the *Teachable Machine*. The method used is to utilize the *TensorFlow* framework in the *Teachable Machine Service* to train image or image data. This *Framework* provides *Convolutional Neural Networks (CNN)* algorithms that can classify images with a high degree of accuracy. The test results showed that the highest optimal value was reached at the 50th epoch value, with a batch size value of 16 and a learning rate of 0.00001 which resulted in an accuracy rate of 70% to 98%. This application is expected to be widely implemented for the benefit of the community in order to recognize the type of cloud that causes the potential for rain.

Keywords: *teachable machine, object detection, cloud types, convolutional neural networks, image optimization*

1. PENDAHULUAN

Convolutional Neural Network (CNN) telah merevolusi klasifikasi dan deteksi objek dalam citra digital (Ajit et al., 2020). Dengan arsitektur yang terinspirasi dari cara manusia memproses visual, CNN mampu mengidentifikasi pola kompleks dalam citra (Lei et al., 2020). Metode deteksi objek bertujuan untuk mengidentifikasi semua objek target dalam gambar target dan menentukan kategori dan informasi posisi untuk mencapai pemahaman visi mesin (Zhang et al., 2020). CNN memproses citra melalui serangkaian *layer* konvolusi yang mengekstrak fitur-fitur penting dari citra (Chen et al., 2021). Kemudian, hasilnya dilalui melalui *layer pooling* untuk mereduksi dimensi dan meningkatkan invariansi terhadap translasi (Singh et al., 2020). Selanjutnya, fitur-fitur tersebut dihubungkan ke dalam lapisan *fully connected* yang memungkinkan jaringan untuk mempelajari hubungan antara fitur-fitur tersebut (Basha et al., 2020). Akhirnya, dengan menggunakan *softmax* atau fungsi aktivasi lainnya, CNN memberikan *output* dalam bentuk probabilitas kelas untuk klasifikasi atau *bounding boxes* untuk deteksi objek (Spagnolo et al., 2022). Dengan ketersediaan data yang cukup dan komputasi yang memadai, CNN telah menjadi salah satu metode yang paling efektif dalam klasifikasi dan deteksi objek pada citra digital (Shen et al., 2018).

Awan merupakan elemen penting dari sistem atmosfer yang berperan signifikan dalam menyebabkan hujan (Reid et al., 2023). Pemahaman terhadap jenis awan, tinggi awan dan karakteristiknya menjadi kunci untuk menginterpretasikan kondisi cuaca (Şen, 2020). Ada berbagai jenis awan yang dapat dikenali berdasarkan bentuk, struktur dan ketinggiannya dalam atmosfer. Awan *cirrus*, *cirrocumulus*, *cirrostratus*, *altocumulus*, *altostratus*, *nimbostratus*, *stratocumulus*, *stratus*, *cumulus* dan *cumulonimbus* masing-masing menyiratkan kondisi atmosfer tertentu (Habsah et al., 2023). Ketinggian awan memiliki peran penting dalam penentuan kondisi atmosfer. Pengamatan ketinggian awan memberikan wawasan tentang perubahan tekanan udara, suhu dan kelembaban udara pada berbagai lapisan atmosfer (Rauber, 2018).

Teachable Machine merupakan suatu *platform* yang dikembangkan oleh Google yang memungkinkan pengguna untuk membuat model pembelajaran mesin tanpa perlu memiliki latar belakang pemrograman yang mendalam. *Teachable Machine* menggunakan pembelajaran *transfer*, untuk menemukan pola dan tren dalam sampel gambar atau suara dan membuat model klasifikasi yang sederhana dan mudah dalam hitungan detik. Dengan pembelajaran *transfer*, pengguna dapat menambahkan pembelajaran sendiri dan melatih ulang model di atas model yang telah dilatih sebelumnya dari kumpulan *big data* (Carney et al., 2020). *Teachable Machine* mampu menghasilkan akurasi sampai dari 90-100% (Agustian et al.,

2021)(Malahina et al., 2022). Dalam konteks deteksi citra atau gambar awan, pengguna mengumpulkan gambar dari berbagai jenis awan dan menggunakan *Teachable Machine* untuk melatih model agar dapat mengenali dan mengklasifikasikan jenis awan berdasarkan citra atau gambar awan tersebut. Proses ini dapat dioptimalkan dengan menyesuaikan *hyperparameter* seperti jumlah *epoch* (iterasi seluruh *dataset*), ukuran *batch* (jumlah sampel yang diproses) dan *learning rate* (kecepatan model belajar) (Rochmawati et al., 2021)(Fajri et al., 2022)(Salim et al., 2023a). Selanjutnya, model dapat diintegrasikan ke dalam sistem atau aplikasi *web mobile*, memungkinkan pengguna untuk mengidentifikasi jenis awan dengan cara mengambil citra atau gambar awan (objek) tersebut.

Hasil dari proses ini dapat sangat bervariasi tergantung pada kualitas data dan jumlah *dataset* yang digunakan untuk melatih model dan parameter yang dipilih untuk optimasi (Liu et al., 2023). Dengan *dataset* yang cukup besar dan penyesuaian parameter yang tepat, memungkinkan untuk mencapai tingkat akurasi yang tinggi (Mohd Safuan et al., 2022)(Chen, 2023).

Penelitian ini berkontribusi dalam mengidentifikasi nilai-nilai optimal dari parameter pelatihan guna meningkatkan efisiensi proses klasifikasi pada *dataset* citra tertentu. Oleh karena itu, analisis mendalam terhadap parameter *epoch*, *batch size*, dan *learning rate* diperlukan untuk memperoleh konfigurasi terbaik. Penemuan nilai optimal dari ketiga parameter tersebut bertujuan untuk meningkatkan akurasi deteksi citra sekaligus meminimalkan risiko *overfitting* dan menghemat waktu komputasi.

2. LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian ini merujuk pada sejumlah studi terdahulu yang menerapkan metode *Convolutional Neural Network* (CNN) dengan memanfaatkan *platform Teachable Machine* dan *framework Tensorflow*. (Arrofiqoh dan Harintaka, 2018) mengimplementasikan pendekatan klasifikasi semantik otomatis untuk membedakan jenis tanaman sebagai alternatif dalam pengenalan objek berbasis metode *deep learning*. Penelitian ini menggunakan lima kelas tanaman, yaitu padi, bawang merah, kelapa, pisang, dan cabai. Hasil pelatihan jaringan menunjukkan akurasi 100% pada data latih, 93% pada data validasi, dan 82% pada data uji.

Sementara itu, (Edwin Ariesto Umbu Malahina, 2023) menjadikan dialek Kambara sebagai studi kasus untuk pengenalan suara menggunakan *Teachable Machine* berbasis *open-source*. Penelitian ini mengintegrasikan pendekatan CNN dengan teknik *Mel Frequency Cepstral Coefficients* (MFCC) untuk meningkatkan akurasi dalam proses ekstraksi fitur suara. Hasil evaluasi empiris menunjukkan tingkat akurasi deteksi dialek berkisar antara 98,3% hingga

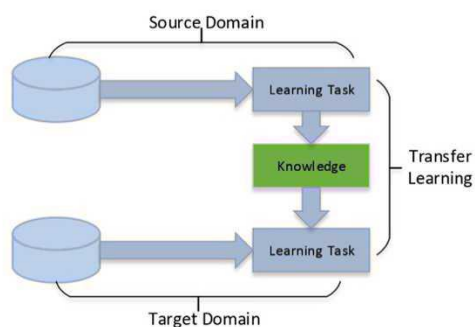
99,6%, dengan tingkat kepuasan pengguna mencapai 99,33%.

Penelitian oleh (Saitakela, 2024) berfokus pada optimasi klasifikasi citra tanaman herbal berdasarkan parameter *epoch*, *batch size*, dan *learning rate*. Tujuan utama penelitian ini adalah mengembangkan aplikasi sistem cerdas berbasis *web mobile*. Proses pelatihan menggunakan *framework Tensorflow* melalui layanan *Teachable Machine*, yang menyediakan algoritma CNN dengan tingkat akurasi klasifikasi citra yang tinggi.

2.2 Transfer Learning dan TensorflowJS

Metode dalam *machine learning* yaitu *transfer learning* adalah ketika model, yang telah dilatih pada suatu tugas, diterapkan pada tugas lain, menggunakan pengetahuan yang diperoleh selama pelatihan sebagai titik awal (Loey et al., 2020). Sebagai contoh, ketika kita melatih model sederhana untuk mengklasifikasikan bahwa gambar apakah makanan, kita bisa menggunakan pengetahuan yang kita peroleh untuk melatih model untuk mengenali minuman (Farahani et al., 2020).

Secara umum, *transfer learning* mencoba menggunakan apa yang telah dipelajari sebagai satu tugas dari perbaikan generalisasi di dalam tugas yang lain. Bobot yang telah dipelajari oleh jaringan pada 'tugas A' dapat dikembangkan ke 'tugas B' yang lain (Hussain et al., 2019). Intinya adalah pengetahuan dari model yang sudah bekerja yang telah dipelajari di luar data pelatihan yang penuh dalam tugas yang spesifik, dimana kita mempunyai data yang sangat sedikit (Williams et al., 2020). Dengan kata lain, kita tidak memulai pembelajaran dari nol tetapi dengan pola yang telah kita pelajari dari penyelesaian tugas terkait.



Gambar 1. Transfer Learning Proses

Pada gambar 1, *transfer learning* adalah teknik pembelajaran mesin yang memanfaatkan pengetahuan yang telah dipelajari oleh model pada tugas tertentu untuk meningkatkan kinerja model pada tugas lain yang berbeda. Dalam konteks ini, model yang telah terlatih dalam pengenalan suara dapat digunakan kembali untuk klasifikasi suara lainnya, menghemat waktu dan sumber daya pelatihan (Zhuang et al., 2021). Dalam *transfer learning* yang digunakan *Teachable Machine*, ada *library tensorflow.js*. *Tensorflow.js* adalah pustaka

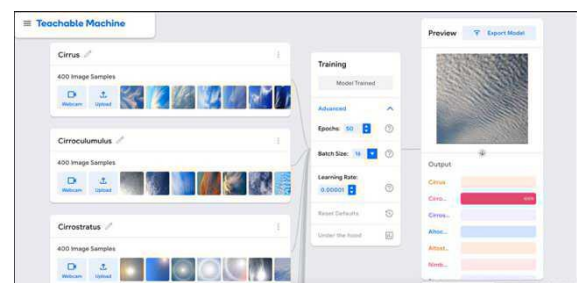
JavaScript open-source yang memungkinkan pengembang untuk melatih dan menjalankan model pembelajaran mesin di *browser* atau lingkungan *node.js* (Gerard, 2021). Dengan *tensorflow.js*, pengguna menjalankan kemampuan pembelajaran mesin di seluruh aplikasi *web*, termasuk pengenalan citra atau gambar dan suara, deteksi objek, penerjemahan bahasa dan masih banyak lagi di lingkungan *JavaScript* yang sangat sesuai (Anwar, 2022). Dengan dukungan yang dihasilkan model dan kemampuan untuk melatih model baru *online* dan *offline*, *tensorflow.js* melakukan langkah besar dalam memberikan pembelajaran mesin di *browser* untuk pengembang *web* (Rivera, 2020).

2.3 Teachable Machine

Teachable Machine adalah *web based tool* yang memungkinkan kita melatih komputer untuk mengenali citra atau gambar, suara dan pose. Kita menggunakan *file* kita atau kamera *web* untuk melatih model kita, kemudian mengeksponnya untuk digunakan dalam proyek kita sendiri, seperti situs *web* atau aplikasi. Dengan *Teachable Machine*, kita bisa menciptakan model *machine learning* tanpa harus jadi ahli atau belajar pemrograman khusus. Sistem bekerja pada *tensorflow.js* yang artinya model yang kita ciptakan dapat dijalankan pada *random platform* yang menjalankan perangkat lunak *JavaScript* (Mathew and Mahesh, 2021)(Wong and Fadzly, 2022).

Teachable Machine memungkinkan pengguna untuk hanya melatih model *machine learning* menggunakan *webcam* dan mikrofon komputer mereka. Pengguna diperbolehkan memberikan contoh kepada mesin tentang berbagai kelas atau kategori dengan melakukan tindakan yang berbeda di depan kamera atau memberikan suara yang berbeda ke mikrofon. Sebagai contoh, pengguna dapat melatih mesin untuk mengenali ekspresi wajah manusia, mendeteksi objek tertentu atau membedakan antara berbagai jenis suara (Shoukat, 2021).

Gambar 2. Tampilan Website Teachable Machine Deteksi Objek Citra/Gambar



Pada gambar 2, menunjukkan klasifikasi dan uji data objek *class* pada layanan *Teachable Machine*, pengguna dapat melakukan dengan mengunjungi *website www.teachablemachine.withgoogle.com*. Pengguna harus *login* dan masuk menggunakan akun Google, maka akan ditampilkan pilihan deteksi objek suara, citra atau gambar dan pose, maka pengguna

bisa memilih citra atau gambar untuk melakukan klasifikasi. Pengguna bisa menyiapkan data berupa gambar baik direkam langsung dalam *Teachable Machine* dengan menggunakan *webcam* atau mengunggah *file* gambar yang dimiliki dan buat nama-nama *class* nya dalam *Teachable Machine* sebagai label untuk dikenali. Jika data gambar sudah diunggah dan dibuatkan *class* masing-masing, objek gambar bisa klik tombol *training* model, tetapi sebelum itu pengguna tentukan dulu nilai *hyperparameter* seperti *epoch*, *batch size* dan *learning rate* untuk mendapatkan hasil terbaik. Jika *training* telah selesai maka pengguna akan melakukan uji data baik secara *real time* atau melakukan unggah gambar uji untuk dilatih dan dikenali dari hasil yang sudah dilatih dalam *Teachable Machine*.

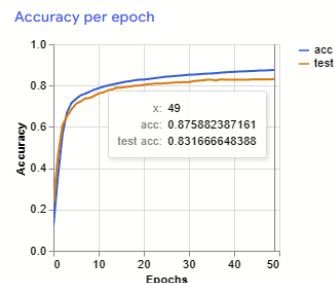


Gambar 3. Tahap Pengenalan Citra/Gambar (Teachable Machine)

Pada gambar 3, untuk hasil yang sudah memuaskan pengguna bisa *export* model ke jenis-jenis bahasa pemrograman baik berbasis *web*, *desktop* atau *mobile*. *Teachable Machine* menyediakan sumber *code* seperti *python*, *javascript* untuk pengembang bisa mengembangkan aplikasi dengan kerangka kerja *framework tensorflow* dari sumber *code* yang disediakan oleh *Teachable Machine*.

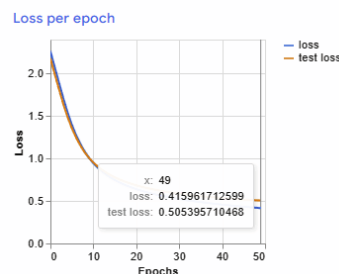
Dalam konteks mendeteksi objek menggunakan *Teachable Machine*, istilah “*accuracy per epoch*” adalah metrik yang mengukur persentase jumlah data yang diklasifikasikan dengan benar oleh model dalam satu siklus pelatihan. Dengan kata lain adalah seberapa akurat model kita dapat memberikan keluaran pada saat itu dengan data input atau seberapa baik model dapat mengenali objek berdasarkan data yang diberikan kepada mereka pada waktu itu. Secara ringkas, “*accuracy per epoch*” adalah rasio yang mengukur seberapa baik bentuk model kita dalam pelatihan untuk gambar klasifikasi objek. Semakin tinggi nilai “*accuracy per epoch*”, semakin baik model kita dalam mengenali objek secara benar sedangkan untuk “*loss per epoch*” adalah kesalahan prediksi model pada data pelatihan dari satu siklus pelatihan. Objektif dalam melatih model adalah meningkatkan *accuracy per epoch* dan menurunkan *loss per epoch*. Dan dapat dilakukan dengan mengubah parameter seperti parameter *learning rate*

untuk memastikan model dapat mempelajari dan mengklasifikasi citra atau gambar dengan lebih baik sesuai dengan apa yang kita inginkan. Berikut adalah hasil yang dapat dilihat pada gambar 4 dan 5 di bawah:



Gambar 4. Accuracy Per Epoch

Pada gambar 4, merupakan tampilan grafik *accuracy per epoch* dalam tangkapan layar pada situs *Teachable Machine* menampilkan metrik “*acc*” yang mengukur akurasi dalam pelatihan model, menunjukkan seberapa baik model memprediksi data pelatihan. Nilai “*acc*” berkisar 0,99 yang berarti model memiliki akurasi 99% terhadap data pelatihan. Sementara “*test acc*” adalah akurasi validasi yang mengukur seberapa baik model memprediksi data uji yang tidak digunakan dalam pelatihan. Nilai “*test acc*” berkisar 0,88 menunjukkan akurasi 88% terhadap data validasi.



Gambar 5. Loss Per Epoch

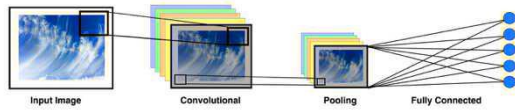
Pada gambar 5, terdapat grafik perubahan *loss per epoch* dalam tangkapan layar pada situs *Teachable Machine*, yang menggambarkan kemampuan model dalam melakukan prediksi pada seluruh *dataset* selama setiap iterasi pelatihan. “*loss*” adalah metrik yang mengukur seberapa jauh prediksi model dari nilai sebenarnya dan semakin rendah nilai *loss*, semakin baik model dalam melakukan prediksi dengan nilai *loss* antara 0,04 atau 4%).

2.4 Convolutional Neural Network

Google *Teachable Machine* mendukung untuk jaringan saraf konvolusional, menyediakan alat yang sangat mumpuni atau *tools* yang mudah digunakan dalam tugas klasifikasi objek, citra atau gambar (Toivonen et al., 2020). *Convolutional Neural Network* atau CNN adalah metode *deep learning* yang digunakan untuk mendeteksi atau mengenali objek dalam citra digital. CNN ini diklaim sebagai model

terbaik untuk melaksanakan atau memecahkan permasalahan *Object Detection* dan *Object Recognition* (Felix et al., 2020).

CNN memiliki lapisan-lapisan, yaitu *convolution layers*, *pooling layers* dan *fully connected*. Lihat arsitekturnya seperti gambar dibawah ini:

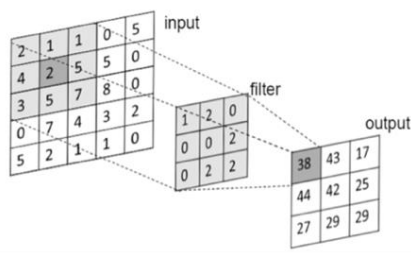


Gambar 6. Arsitektur CNN

Pada gambar 6, lapisan konvolusi adalah lapisan yang berfungsi untuk mengubah *input* gambar mentah direpresentasi menjadi lebih bermakna dan mudah dipahami oleh jaringan. Lapisan konvolusional terdiri dari satu set kernel konvolusi dimana setiap *neuron* bertindak sebagai sebuah kernel (Alabassy et al., 2020). Operasi konvolusi dapat dinyatakan sebagai berikut:

$$f_l^k(p, q) = \sum_c \sum_{x,y} i_c(x, y) \cdot e_l^k(u, v) \quad (1)$$

Dimana, $i_c(x, y)$ adalah elemen dari *tensor* gambar *input* i_c , yang merupakan elemen dikalikan dengan $e_l^k(u, v)$ indeks dari k^{th} kernel konvolusi k' dari lapisan l^{th} . Sedangkan *output* fitur-peta k^{th} operasi konvolusi dapat dinyatakan sebagai $f_i^k = [f_i^k(1,1), \dots, f_i^k(p, q), \dots, f_i^k(p, q)]$. Proses operasi konvolusi dari *input-filter-output* diilustrasikan sebagai berikut:



Gambar 7. Arsitektur CNN

Pada gambar 7, menjelaskan tentang operasi konvolusi dimana salah satu komponen *convolution layers* antara lain adalah *strides*, merupakan parameter atau batasan yang dapat menentukan jumlah pergeseran *filter*. Jika nilai dari langkahnya yaitu 1, maka *filter* konvolusional akan bergeser sebesar 1 piksel bergerak secara horizontal dan vertical (Yepez and Ko, 2020), selain itu ada *padding*, merupakan parameter yang bertujuan menentukan jumlah piksel yang berisi angka 0 yang akan ditambahkan ke setiap sisi *input* atau sebuah operasi yang meningkatkan ukuran data inputan (Wikarsa et al., 2022).

Tujuan dari *pooling layers* adalah untuk mengurangi jumlah parameter atau jumlah perhitungan pada jaringan resolusi dari fitur citra atau gambar dengan tetap mempertahankan informasi

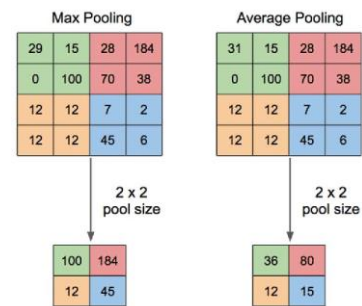
spasial yang lebih penting (Ozdemir et al., 2023), tujuan lain dari *pooling layers* adalah mencegah *overfitting* dari model itu sendiri, *overfitting* terjadi ketika model belajar terlalu detail pada data *training* (Zafar et al., 2022). *Pooling layers* ini terbagi menjadi *max pooling* dan *average pooling* seperti pada gambar 8. *Max Pooling* mengambil nilai terbesar dari area yang dipilih (Nurjannah et al., 2022), *max pooling* ditunjukkan pada persamaan berikut:

$$f_{max}(x) = \max_i \max_i \quad (2)$$

Sedangkan *average pooling* mengambil nilai rata-rata dari area yang dipilih (Lee et al., 2018), *average pooling* ditunjukkan pada persamaan berikut:

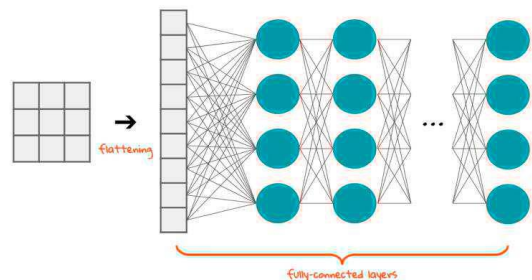
$$f_{avg} = \frac{1}{N} \sum_i N = 1^x \quad (3)$$

Dimana x adalah vektor yang mewakili aktivasi dari kotak persegi panjang dengan N adalah permutasi dalam gambar atau saluran.



Gambar 8. Max Pooling dan Average Pooling

Fully connected adalah layer yang terhubung sepenuhnya, berada di bagian akhir dari setiap jaringan *neuron-neuron* pada layer ini terhubung dengan setiap *neuron* pada layer sebelumnya (Bhugwan et al., 2019), sehingga tujuan dari lapisan *fully connected* adalah untuk mengubah *input* gambar mentah menjadi representasi yang lebih bermakna dan mudah dipahami oleh jaringan (Alzubaidi et al., 2021).



Gambar 9. Fully Connected Layer

2.5 Epoch, Learning Rate dan Batch Size

Terdapat tiga parameter utama dalam *deep learning* khususnya dalam algoritma *deep neural network* untuk mengukur pelatihan *dataset*. Pertama

adalah *epoch*, kedua adalah *batch size* dan yang terakhir *learning rate* (Shafi and Assad, 2023).

1. *Epoch* adalah jumlah kali seluruh *dataset* pelatihan diberikan ke model yang akan melatih model tersebut (Dhande and Shaikh, 2019). *Epoch* memungkinkan model mempelajari pola dalam data ulang lebih banyak. Nilai *epoch* pada *Teachable Machine* mulai dari 1 sampai tak terbatas, namun semakin banyak nilai *epoch*, maka model tersebut akan *overfitting*. *Overfitting* adalah kondisi model pelatihan menjadi terlalu spesifik untuk data latihan dan performanya akan menurun pada data yang baru (Cui et al., 2024). Secara umum, jumlah *epoch* yang lebih besar akan menyebabkan model pembelajaran mesin belajar dengan lebih akurat, tetapi lebih lama. Jumlah *epoch* yang lebih kecil akan menyebabkan model pembelajaran mesin belajar dengan lebih cepat, tetapi kurang akurat.
2. *Learning Rate* adalah parameter yang mengontrol seberapa sering jaringan saraf harus memperbaharui pengetahuan yang telah dipelajari. *Learning rate* mengukur sejauh apa langkah-langkah yang diambil sepanjang proses optimisasi dalam melatih sebuah model (Wen et al., 2021). *Learning rate* berada pada rentang 1 yang terbesar hingga 0,00001 yang terkecil. Semakin kecil *learning rate*, semakin lambat dalam belajarnya, namun dapat mencegah *overfitting* (Liu et al., 2022). Oleh karena itu, model CNN *Teachable Machine* direkomendasikan memiliki *learning rate* 0,001 hingga 0,01.
3. *Batch Size* mengontrol jumlah sampel data yang digunakan untuk sekali iterasi pelatihan model pembelajaran mesin. *Batch size* digunakan dalam model pelatihan dan bergantung dan juga dibatasi oleh kapasitas memori (Piao et al., 2023). *Batch size* model pada *Teachable Machine* berkisar dari 16 sampai 512, *batch size* yang akan mempengaruhi waktu pelatihan model, waktu per *epoch* dan kualitas. Oleh sebab itu dapat dilakukan percobaan dengan ukuran *batch size* yang sesuai agar hasilnya akurat. Jumlah *batch size* yang lebih kecil memberikan hasil pembelajaran model yang belajar dengan lebih akurat, tetapi juga menyebabkan pelatihan model pembelajaran membutuhkan waktu lebih lama (Lu et al., 2022). Pentingnya menemukan efektifitas *dataset* pelatihan dari ketiga parameter untuk mendapatkan tingkat deteksi yang optimal dan akurat (Salim et al., 2023b).

Epoch, *batch size* dan *learning rate* adalah nilai parameter yang dirangkum dari berbagai sumber untuk menggunakan nilai minimal dan maksimal untuk pengujian deteksi objek, citra atau gambar. Nilai *epoch* berkisar antara 10; 50; 100; 250; 750 dan 1000. *Epoch* yang pernah dilakukan para penelitian sebelumnya (Nurfita and Ariyanto, 2018)(Sari et al.,

2022), untuk nilai *batch size* dari 16; 32; 64; 128; 256 dan 512 yang tersedia secara *default* di *Teachable Machine*, kemudian untuk *learning rate* berdasarkan rata-rata penelitian sebelumnya berkisar dari 0,00001; 0,0001; 0,001; 0,01; 0,1 dan 1 (Johny and Madhusoodanan, 2021)(Pan, 2022)(Ni Komang Rai Mirayanti et al., 2023). Nilai ini adalah nilai minimal dan maksimal yang telah tersedia.

Selain ketiga *hyperparameter* di atas yaitu *epoch*, *batch size* dan *learning rate*, persentase dan akurasi juga akan dipengaruhi oleh jumlah *dataset* latih. Untuk itulah pada penelitian ini dilakukan pengujian eksplorasi langsung untuk mengetahui hasil persentase akurasi yang diperoleh serta melalui pengukuran dan sampel data yang diberikan lalu diujikan pada situs *Teachable Machine*.

3. METODE PENELITIAN

3.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan eksperimental kuantitatif untuk menguji pengaruh nilai *hyperparameter* terhadap performa klasifikasi citra awan. Tiga parameter utama yang dianalisis meliputi *epoch*, *batch size*, dan *learning rate*, yang diterapkan pada model klasifikasi berbasis *Convolutional Neural Network* (CNN) melalui platform *Teachable Machine* menggunakan framework *Tensorflow*.

3.2 Sumber dan Pengolahan Data

Data yang digunakan dalam penelitian ini berupa citra atau gambar jenis awan, yang dikumpulkan dari berbagai sumber daring (internet) serta koleksi pribadi peneliti. Seluruh gambar kemudian melalui proses *preprocessing* berupa:

1. Pembersihan data (removal of noise/redundant images),
2. Kategorisasi jenis awan berdasarkan lapisan ketinggian (tinggi, menengah dan rendah),
3. Penyesuaian ukuran citra sesuai dengan format *input* model *Teachable Machine*.

3.3 Klasifikasi Data

Jenis awan yang digunakan sebagai sampel klasifikasi terdiri dari 10 kelas, yaitu:

1. *Cirrus*, *Cirrocumulus*, *Cirrostratus* (awan tinggi),
2. *Alto cumulus*, *Altostratus*, *Nimbostratus* (awan menengah),
3. *Stratocumulus*, *Stratus*, *Cumulus* dan *Cumulonimbus* (awan rendah).

Masing-masing kelas memiliki 400 gambar, sehingga total data latih berjumlah 4.000 citra. Untuk validasi eksternal, peneliti juga menambahkan data uji berupa: citra awan lain di luar *dataset* latih dan objek non-awan (seperti kapas, benang, gula pasir, matahari, salju, dan sebagainya) untuk menguji kemampuan generalisasi model.

3.4 Eksperimen Pengujian Parameter

Pengujian dilakukan dengan mengatur kombinasi nilai *hyperparameter* sebagai berikut:

Tabel 1. Parameter Pengujian Nilai *Epoch*, *Batch Size* dan *Learning Rate*

<i>Epoch</i>	10	50	100	250	750	1000
<i>Learning rate</i>	0,0001	0,0001	0,0001	0,01	0,1	1
<i>Batch size</i>	16	32	64	128	256	512

Dengan kombinasi tersebut, dilakukan 216 kali pengujian untuk mengamati variasi nilai akurasi terhadap klasifikasi citra. Setiap kombinasi diuji menggunakan pembagian *dataset* menjadi dua bagian, yaitu data latih (*training data*) dan data uji (*testing data*). Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk mengevaluasi performa klasifikasi terhadap data yang belum pernah dilihat model sebelumnya.

3.5 Platform dan Implementasi

Model CNN dibangun dan dilatih menggunakan *Teachable Machine*, yang secara otomatis mengelola struktur CNN dasar untuk klasifikasi gambar. Setelah pengujian selesai, model dengan konfigurasi terbaik diekspor dalam format *Tensorflow.js* untuk

digunakan dalam pengembangan aplikasi *web mobile* berbasis edukasi.

3.6 Visualisasi dan Validasi

Seluruh hasil klasifikasi divalidasi menggunakan metrik akurasi terhadap *Original Test Class* dan *Outside Test Class*. Gambar 10 menampilkan representasi visual dari kumpulan data awan yang digunakan dalam proses pelatihan dan pengujian model.



Gambar 10. Kumpulan Gambar Jenis Awan

4. HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Model

Pengujian dilakukan menggunakan *dataset* citra awan yang terdiri dari 10 kelas dengan masing-masing 4.000 sampel, dilatih melalui *platform Teachable Machine*. Hasil terbaik diperoleh berdasarkan evaluasi pada dua kategori, yaitu *Original Test Class* dan *Outside Test Class*, sebagaimana disajikan pada tabel 2.

Tabel 2. Tingkat Pengujian Terbaik Berdasarkan *Original Test Class* dan *Outside Test Class Teachable Machine*

Dataset	Epoch	Batch Size	Learning Rate	Loss per Epoch		Accuracy per Epoch		Original Test	Outside Test
				Loss	Test Loss	Acc	Test Acc	Class (100%)	Class (%)
4.000	50	16	0,00001	0,43	0,47	0,86	0,85	70-98	33-99
4.000	100	256	0,00001	0,79	0,85	0,80	0,77	57-100	32-84
4.000	10	512	0,0001	1,22	1,66	0,75	0,75	48-100	33-45
4.000	50	128	0,00001	0,87	0,87	0,79	0,79	57-95	31-64
4.000	10	256	0,0001	0,85	0,87	0,79	0,79	55-97	60-96

Parameter utama yang diuji meliputi *epoch*, *batch size*, dan *learning rate*. Hasil analisis menunjukkan bahwa konfigurasi optimal terdapat pada *epoch* 50, *batch size* 16, dan *learning rate* 0.00001. Kombinasi ini menghasilkan tingkat akurasi tinggi pada *Original Test Class* serta nilai minimum pada *Outside Test Class*, yang menunjukkan efektivitas klasifikasi dan kemampuan model dalam membedakan objek target dari objek yang tidak relevan (seperti: kapas, matahari, benang, gula pasir, kursi, kemeja putih, dan salju).

4.2 Analisis Hyperparameter

1. Epoch

Peningkatan nilai *epoch* berpengaruh terhadap akurasi, namun berisiko menimbulkan *overfitting* jika terlalu tinggi. Pada konfigurasi *epoch* 10 dengan *batch size* 256 dan *learning rate* 0.0001, model justru mengenali objek *non-target* (misalnya kapas) dengan akurasi hingga 98%, yang menunjukkan kegagalan generalisasi. Oleh karena

itu, pemilihan nilai *epoch* yang tepat menjadi sangat penting.

2. Batch Size

Batch size yang lebih besar memberikan hasil akurasi yang lebih stabil, karena model mampu mengenali pola lebih baik pada setiap iterasi. Meski demikian, peningkatan *batch size* memerlukan waktu pelatihan lebih lama untuk mencapai konvergensi optimal.

3. Learning Rate

Nilai *learning rate* yang terlalu tinggi (misalnya 0.001 hingga 1) menyebabkan pembelajaran yang terlalu cepat dan tidak stabil, berujung pada akurasi 100% terhadap data latih namun buruk terhadap data uji (indikasi *overfitting*). Sebaliknya, *learning rate* kecil (0.00001 hingga 0.0001) memungkinkan model belajar secara bertahap dan stabil.

4.3 Pengaruh Kualitas dan Kuantitas Data

Jumlah data dan kualitas anotasi citra sangat mempengaruhi performa model. Penggunaan 4.000 sampel terdiri dari 10 kelas citra dengan masing-

masing 400 data per kelas memberikan kontribusi signifikan terhadap kemampuan model mengenali pola dan fitur dari setiap jenis awan. Pembagian *dataset* menjadi data latih dan data uji secara proporsional menjadi langkah krusial untuk mengukur keandalan model dalam menghadapi data baru.

4.4 Implementasi Model pada Aplikasi Web Mobile

Model terbaik diekspor dalam format *Tensorflow.js* melalui fitur *Export Model* di *Teachable Machine* untuk diterapkan pada aplikasi *web mobile*. Antarmuka visual dirancang agar intuitif bagi pengguna umum. Pada pengujian *real-time* (Gambar 11), sistem mampu mendeteksi jenis awan dari kamera maupun unggahan gambar, dengan akurasi antara 70% hingga 98%. Selain itu, aplikasi memberikan informasi tambahan seperti nama awan dan potensi hujan.



Gambar 11. Implementasi Dalam Web Mobile

4.5 Tantangan Kemiripan Visual

Kesalahan klasifikasi masih terjadi apabila objek uji memiliki kemiripan visual dengan awan dalam basis data, seperti kapas, benang, dan salju. Hal ini menunjukkan perlunya pengayaan *dataset* dan peningkatan representasi fitur untuk mengurangi ambiguitas klasifikasi.

5 KESIMPULAN

Penelitian ini bertujuan untuk menentukan konfigurasi *hyperparameter* yang paling optimal dalam proses klasifikasi citra awan menggunakan layanan *Teachable Machine*. Parameter yang diuji meliputi nilai *epoch* (10, 50, 100, 250, 750, dan 1000), *batch size* (16, 32, 64, 128, 256, dan 512), serta *learning rate* (0.00001; 0.0001; 0.001; 0.01; 0.1; dan 1), dengan jumlah total data latih sebanyak 4.000 sampel, terdiri dari 10 kelas citra dengan masing-masing 400 data per kelas.

Berdasarkan hasil pengujian dan evaluasi performa model, ditemukan lima konfigurasi terbaik dengan akurasi tinggi, dan di antaranya konfigurasi paling optimal terdapat pada kombinasi *epoch* ke-50, *batch size* 16, dan *learning rate* 0.00001. Kombinasi ini menghasilkan akurasi klasifikasi antara 70% hingga 98%, serta menunjukkan kemampuan generalisasi yang baik terhadap data uji.

Model terbaik kemudian diekspor dalam format *Tensorflow.js* melalui fitur *Export Model* pada platform *Teachable Machine*, dengan menggunakan parameter hasil pengujian sebagai acuan. File model tersebut selanjutnya dikembangkan menjadi aplikasi *web mobile* berbasis edukasi, yang memungkinkan pengguna untuk mendeteksi jenis awan secara *real-time* maupun melalui unggahan gambar. Aplikasi ini dapat mendukung literasi cuaca dan teknologi kecerdasan buatan secara luas di masyarakat.

Penelitian ini berkontribusi dengan menentukan nilai parameter ideal yang menghasilkan akurasi terbaik, sehingga dapat menjadi referensi bagi penelitian selanjutnya untuk efisiensi waktu dan akurasi yang optimal.

DAFTAR PUSTAKA

- AGUSTIAN, D., PERTAMA, P.P.G.P., CRISNAPATI, P.N., NOVAYANTI, P.D., 2021. Implementation of Machine Learning Using Google's Teachable Machine Based on Android, in: 2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS). IEEE, pp. 1–7. <https://doi.org/10.1109/ICORIS52787.2021.9649528>
- AJIT, A., ACHARYA, K., SAMANTA, A., 2020. A Review of Convolutional Neural Networks, in: 2020 International Conference on Emerging Trends in Information Technology and Engineering (Ic-ETITE). IEEE, pp. 1–5. <https://doi.org/10.1109/ic-ETITE47903.2020.049>
- ALABASSY, B., SAFAR, M., EL-KHARASHI, M.W., 2020. A High-Accuracy Implementation for Softmax Layer in Deep Neural Networks, in: 2020 15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS). IEEE, pp. 1–6. <https://doi.org/10.1109/DTIS48698.2020.9081313>
- ALZUBAIDI, L., ZHANG, J., HUMAIDI, A.J., AL-DUJAILI, A., DUAN, Y., AL-SHAMMA, O., SANTAMARÍA, J., FADHEL, M.A., AL-AMIDIE, M., FARHAN, L., 2021. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. J Big Data 8, 53. <https://doi.org/10.1186/s40537-021-00444-8>
- ANWAR, C., 2022. Deteksi Objek Berbasis Web Menggunakan Tensorflow Js dan Coco Dataset pada Framework React Js. Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

- 5, 1008–1015.
<https://doi.org/10.32672/jnkti.v5i6.5464>
- ARROFIQOH, E.N., HARINTAKA, H., 2018. IMPLEMENTASI METODE CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI TANAMAN PADA CITRA RESOLUSI TINGGI. GEOMATIKA 24, 61. <https://doi.org/10.24895/JIG.2018.24-2.810>
- BASHA, S.H.S., DUBEY, S.R., PULABAIGARI, V., MUKHERJEE, S., 2020. Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification. *Neurocomputing* 378, 112–119. <https://doi.org/10.1016/j.neucom.2019.10.008>
- BHUGWAN, D., RANCHOD, P., KLEIN, R., ROSMAN, B., 2019. A Comparison Between Fully Connected and Deconvolutional Layers for Road Segmentation from Satellite Imagery, in: 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA). IEEE, pp. 117–122. <https://doi.org/10.1109/RoboMech.2019.8704849>
- CARNEY, M., WEBSTER, B., ALVARADO, I., PHILLIPS, K., HOWELL, N., GRIFFITH, J., JONGEJAN, J., PITARU, A., CHEN, A., 2020. Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification, in: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems. ACM, pp. 1–8. <https://doi.org/10.1145/3334480.3382839>
- CHEN, L., LI, S., BAI, Q., YANG, J., JIANG, S., MIAO, Y., 2021. Review of Image Classification Algorithms Based on Convolutional Neural Networks. *Remote Sens (Basel)* 13, 4712. <https://doi.org/10.3390/rs13224712>
- CHEN, S., 2023. CNN Combined with Data Augmentation for Face Recognition on Small Dataset. *J Phys Conf Ser* 2634, 012040. <https://doi.org/10.1088/1742-6596/2634/1/012040>
- CUI, J., LI, XUEWEI, ZHAO, H., WANG, H., LI, B., LI, XI, 2024. Epoch-Evolving Gaussian Process Guided Learning for Classification. *IEEE Trans Neural Netw Learn Syst* 35, 326–337. <https://doi.org/10.1109/TNNLS.2022.3174207>
- DHANDE, G., SHAIKH, Z., 2019. Analysis of Epochs in Environment based Neural Networks Speech Recognition System, in: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, pp. 605–608. <https://doi.org/10.1109/ICOEI.2019.8862728>
- EDWIN ARIESTO UMBU MALAHINA, 2023. Teachable Machine: Deteksi Dialek Sumba Timur (Kambara) Menggunakan Layanan Open Source. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi* 12, 280–286. <https://doi.org/10.22146/jnteti.v12i4.8174>
- FAJRI, F.N., MALIK, K., PRATAMASUNU, G.Q.O., 2022. Metode Pengumpulan Data Pada Deteksi Pakaian Hijab Syar'I Berdasarkan Citra Digital Menggunakan Teachable machine Learning. *Justek : Jurnal Sains dan Teknologi* 5, 194. <https://doi.org/10.31764/justek.v5i2.11614>
- FARAHANI, A., POURSHOJAE, B., RASHEED, K., ARABNIA, H.R., 2020. A Concise Review of Transfer Learning, in: 2020 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, pp. 344–351. <https://doi.org/10.1109/CSCI51800.2020.00065>
- FELIX, F., WIJAYA, J., SUTRA, S.P., KOSASIH, P.W., SIRAIT, P., 2020. Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun. *Jurnal SIFO Mikroskil* 21, 1–10. <https://doi.org/10.55601/jsm.v21i1.672>
- GERARD, C., 2021. TensorFlow.js, in: Practical Machine Learning in JavaScript. Apress, Berkeley, CA, pp. 25–43. https://doi.org/10.1007/978-1-4842-6418-8_2
- HABSAH, A.P., AS'ARI, R., NINGSIH, M.P., 2023. Meningkatkan Penguasaan Konsep Materi Klasifikasi Awan Melalui Model Pembelajaran Discovery Learning. *SOSEARCH : Social Science Educational Research* 3, 81–86. <https://doi.org/10.26740/sossearch.v3n2.p81-86>
- HUSSAIN, M., BIRD, J.J., FARIA, D.R., 2019. A Study on CNN Transfer Learning for Image Classification. pp. 191–202. https://doi.org/10.1007/978-3-319-97982-3_16
- JOHNY, A., MADHUSOODANAN, K.N., 2021. Dynamic Learning Rate in Deep CNN Model for Metastasis Detection and Classification of Histopathology Images. *Comput Math Methods Med* 2021, 1–13. <https://doi.org/10.1155/2021/5557168>
- LEE, C.-Y., GALLAGHER, P., TU, Z., 2018. Generalizing Pooling Functions in CNNs: Mixed, Gated, and Tree. *IEEE Trans Pattern Anal Mach Intell* 40, 863–875. <https://doi.org/10.1109/TPAMI.2017.2703082>
- LEI, J., LUO, X., FANG, L., WANG, M., GU, Y., 2020. Region-Enhanced Convolutional Neural Network for Object Detection in Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing* 58, 5693–5702. <https://doi.org/10.1109/TGRS.2020.2968802>

- LIU, H., FU, Q., DU, L., ZHANG, T., YU, G., HAN, S., ZHANG, D., 2022. Learning Rate Perturbation: A Generic Plugin of Learning Rate Schedule towards Flatter Local Minima, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. ACM, New York, NY, USA, pp. 4234–4238.
<https://doi.org/10.1145/3511808.3557626>
- LIU, X., JIANG, S., WU, R., SHU, W., HOU, J., SUN, Y., SUN, J., CHU, D., WU, Y., Song, H., 2023. Automatic Taxonomic Identification Based on the Fossil Image Dataset (415,000 images) and Deep Convolutional Neural Networks. *Paleobiology* 49, 1–22.
<https://doi.org/10.1017/pab.2022.14>
- LOEY, M., MANOGARAN, G., KHALIFA, N.E.M., 2020. A Deep Transfer Learning Model with Classical Data Augmentation and CGAN to Detect COVID-19 from Chest CT Radiography Digital Images. *Neural Comput Appl.* <https://doi.org/10.1007/s00521-020-05437-x>
- LU, K.-W., LIU, P., HONG, D.-Y., WU, J.-J., 2022. Efficient Dual Batch Size Deep Learning for Distributed Parameter Server Systems, in: 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, pp. 630–639.
<https://doi.org/10.1109/COMPSAC54236.2022.00110>
- MALAHINA, E.A.U., HADJON, R.P., BISILISIN, F.Y., 2022. Teachable Machine: Real-Time Attendance of Students Based on Open Source System. *The IJICS (International Journal of Informatics and Computer Science)* 6, 140.
<https://doi.org/10.30865/ijics.v6i3.4928>
- MATHEW, M.P., MAHESH, T.Y., 2021. Object Detection Based on Teachable Machine. *Journal of VLSI Design and Signal Processing* 7.
<https://doi.org/10.46610/JOVDSP.2021.v07i02.003>
- MOHD SAFUAN, S.N., MD TOMARI, M.R., WAN ZAKARIA, W.N., 2022. Cross Validation Analysis of Convolutional Neural Network Variants with Various White Blood Cells Datasets for the Classification Task. *International Journal of Online and Biomedical Engineering (iJOE)* 18, 123–140.
<https://doi.org/10.3991/ijoe.v18i02.27321>
- NI KOMANG RAI MIRAYANTI, SARIYASA SARIYASA, I GEDE ARIS GUNADI, 2023. Batch Size and Learning Rate Effect in COVID-19 Classification Using CNN 7.
- NURFITA, R.D., ARIYANTO, G., 2018. Implementasi Deep Learning berbasis Tensorflow untuk Pengenalan Sidik Jari. *Emitor: Jurnal Teknik Elektro* 18, 22–27.
<https://doi.org/10.23917/emitor.v18i01.6236>
- NURJANNAH, A.F., KURNIASARI, A.S.D., SARI, Z., AZHAR, Y., 2022. Pneumonia Image Classification Using CNN with Max Pooling and Average Pooling. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 6, 330–338.
<https://doi.org/10.29207/resti.v6i2.4001>
- OZDEMIR, C., DOGAN, Y., KAYA, Y., 2023. A New Local Pooling Approach for Convolutional Neural Network: Local Binary Pattern. *Multimed Tools Appl* 83, 34137–34151. <https://doi.org/10.1007/s11042-023-17540-x>
- PAN, J., 2022. The impact of learning rate and data size on CNN for skin cancer detection, in: Yusof, Y. (Ed.), *Second International Conference on Medical Imaging and Additive Manufacturing (ICMIAM 2022)*. SPIE, p. 28.
<https://doi.org/10.1117/12.2636723>
- PIAO, X., SYNN, D., PARK, J., KIM, J.-K., 2023. Enabling Large Batch Size Training for DNN Models Beyond the Memory Limit While Maintaining Performance. *IEEE Access* 11, 102981–102990.
<https://doi.org/10.1109/ACCESS.2023.3312572>
- RAUBER, R., 2018. Clouds and Vertical Motions, in: *Radar Meteorology*. Wiley, pp. 413–435.
<https://doi.org/10.1002/9781118432662.ch17>
- REID, J.S., MARING, H.B., NARISMA, G.T., VAN DEN HEEVER, dkk. The Coupling Between Tropical Meteorology, Aerosol Lifecycle, Convection, and Radiation during the Cloud, Aerosol and Monsoon Processes Philippines Experiment (CAMP2Ex). *Bull Am Meteorol Soc* 104, E1179–E1205.
<https://doi.org/10.1175/BAMS-D-21-0285.1>
- RIVERA, J.D.D.S., 2020. *Practical TensorFlow.js*. Apress, Berkeley, CA.
<https://doi.org/10.1007/978-1-4842-6273-3>
- ROCHMAWATI, N., HIDAYATI, H.B., YAMASARI, Y., TJAHYANINGTIJAS, H.P.A., YUSTANTI, W., PRIHANTO, A., 2021. Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *Journal of Information Engineering and Educational Technology* 5, 44–48.
<https://doi.org/10.26740/jieet.v5n2.p44-48>
- SAITAKELA, M., 2024. Teachable Machine: Optimization of Herbal Plant Image Classification Based on Epoch Value, Batch Size and Learning Rate. *Journal of Applied Data Sciences* 5, 532–545.
<https://doi.org/10.47738/jads.v5i2.206>
- SALIM, S., JAMIL, M.M.A., AMBAR, R., ZAKI, W.S.W., MOHAMMAD, S., 2023a. Learning Rate Optimization for Enhanced Hand Gesture Recognition using Google Teachable Machine, in: *2023 IEEE 13th International Conference on Control System, Computing and Engineering*

- (ICCSCE). IEEE, pp. 332–337. <https://doi.org/10.1109/ICCSCE58721.2023.10237148>
- SALIM, S., JAMIL, M.M.A., AMBAR, R., ZAKI, W.S.W., MOHAMMAD, S., 2023b. Learning Rate Optimization for Enhanced Hand Gesture Recognition using Google Teachable Machine, in: 2023 IEEE 13th International Conference on Control System, Computing and Engineering (ICCSCE). IEEE, pp. 332–337. <https://doi.org/10.1109/ICCSCE58721.2023.10237148>
- SARI, Y., ARIFIN, Y.F., NOVITASARI, FAISAL, M.R., 2022. The Effect of Batch Size and Epoch on Performance of ShuffleNet-CNN Architecture for Vegetation Density Classification, in: 7th International Conference on Sustainable Information Engineering and Technology 2022. ACM, New York, NY, USA, pp. 39–46. <https://doi.org/10.1145/3568231.3568239>
- ŞEN, Z., 2020. Meteorology. pp. 7–52. https://doi.org/10.1007/978-3-030-01542-8_2
- SHAFI, S., ASSAD, A., 2023. Exploring the Relationship Between Learning Rate, Batch Size, and Epochs in Deep Learning: An Experimental Study. pp. 201–209. https://doi.org/10.1007/978-981-19-6525-8_16
- SHEN, Y., HAN, T., YANG, Q., YANG, X., WANG, Y., LI, F., WEN, H., 2018. CS-CNN: Enabling Robust and Efficient Convolutional Neural Networks Inference for Internet-of-Things Applications. IEEE Access 6, 13439–13448. <https://doi.org/10.1109/ACCESS.2018.2810264>
- SHOUKAT, S., & A.S., 2021. Fraud Detection System Using Facial Recognition Based On Google Teachable Machine for Banking Applications. Journal of Innovative Computing and Emerging Technologies 2. <https://doi.org/10.56536/jicet.v2i2.30>
- SINGH, P., RAJ, P., NAMBOODIRI, V.P., 2020. EDS Pooling Layer. Image Vis Comput 98, 103923. <https://doi.org/10.1016/j.imavis.2020.103923>
- SPAGNOLO, F., PERRI, S., CORSONELLO, P., 2022. Aggressive Approximation of the SoftMax Function for Power-Efficient Hardware Implementations. IEEE Transactions on Circuits and Systems II: Express Briefs 69, 1652–1656. <https://doi.org/10.1109/TCSII.2021.3120495>
- TOIVONEN, T., JORMANAINEN, I., KAHILA, J., TEDRE, M., VALTONEN, T., VARTIAINEN, H., 2020. Co-Designing Machine Learning Apps in K–12 With Primary School Children, in: 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT). IEEE, pp. 308–310. <https://doi.org/10.1109/ICALT49669.2020.00099>
- WEN, L., GAO, L., LI, X., ZENG, B., 2021. Convolutional Neural Network With Automatic Learning Rate Scheduler for Fault Classification. IEEE Trans Instrum Meas 70, 1–12. <https://doi.org/10.1109/TIM.2020.3048792>
- WIKARSA, L., ANGDRSEY, A., SOMBOUWADIL, T., 2022. Detection of the Types of Consumable Saltwater Fish in the Coastal Area of Likupang Uses the Convolutional Neural Network Method. Jurnal Pekommas 7.
- WILLIAMS, J., TADESSE, A., SAM, T., SUN, H., MONTAÑEZ, G.D., 2020. Limits of Transfer Learning. pp. 382–393. https://doi.org/10.1007/978-3-030-64580-9_32
- WONG, J.J.N., FADZLY, N., 2022. Development of Species Recognition Models Using Google Teachable Machine on Shorebirds and Waterbirds. Journal of Taibah University for Science 16, 1096–1111. <https://doi.org/10.1080/16583655.2022.2143627>
- YEPEZ, J., KO, S.-B., 2020. Stride 2 1-D, 2-D, and 3-D Winograd for Convolutional Neural Networks. IEEE Trans Very Large Scale Integr VLSI Syst 28, 853–863. <https://doi.org/10.1109/TVLSI.2019.2961602>
- ZAFAR, A., AAMIR, M., MOHD NAWI, N., ARSHAD, A., RIAZ, S., ALRUBAN, A., DUTTA, A.K., ALMOTAIRI, S., 2022. A Comparison of Pooling Methods for Convolutional Neural Networks. Applied Sciences 12, 8643. <https://doi.org/10.3390/app12178643>
- ZHANG, X., CHEN, G., SARUTA, K., TERATA, Y., 2020. A Guideline for Object Detection Using Convolutional Neural Networks. pp. 157–164. https://doi.org/10.1007/978-981-15-0187-6_18
- ZHUANG, F., QI, Z., DUAN, K., XI, D., ZHU, Y., ZHU, H., XIONG, H., HE, Q., 2021. A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE 109, 43–76. <https://doi.org/10.1109/JPROC.2020.3004555>