JURNAL ILMIAH TEKNOLOGI INFORMASI DAN KOMUNIKASI (JTIK) VOL 15, No.2, September 2024, pp. 360 - 370
p-ISSN : 2087-0868 (media cetak)
e-ISSN : 2598-9707 (media online)
http://ejurnal.provisi.ac.id/index.php/JTIKP

OPTIMASI MODEL DEEP LEARNING UNTUK RASPBERRY MENGGUNAKAN PRUNING DAN KUANTISASI

Imadudin Harjanto[1], Muhammad Amiruddin[2], Bambang Hadi Kunaryo[3]
[1]Program Studi Teknik Elektro Universitas PGRI Semarang
Jl. Sidodadi Timur No.24, Kota Semarang, Jawa Tengah 50232, e-mail: imaduddin@upgris.ac.id
[2]Program Studi Teknik Elektro Universitas PGRI Semarang
Jl. Sidodadi Timur No.24, Kota Semarang, Jawa Tengah 50232, e-mail: amiruddin@upgris.ac.id
[3] Program Studi Teknik Elektro Universitas PGRI Semarang
Jl. Sidodadi Timur No.24, Kota Semarang, Jawa Tengah 50232, e-mail: bhadikunaryo@upgris.ac.id

ARTICLE INFO

Article history:

Received : 30 – Juli - 2024
Received in revised form : 31 – Juli - 2024
Accepted : 22 – Agustus - 2024
Available online : 1 – September - 2024

ABSTRACT

Artificial intelligence with deep learning has been widely used at many fields especially in computer vision technology. It can be found in many purpose and variety of device for it's deplayment. Deep learning is a neural network that is arranged in very deep layers in order to extract information in more detail. However, with the high level of performance can be achieved, in other hand, another problem arises; the need for very large computing resources in the process. This study has objective to find method and strategies for optimizing edge device in order to continue implementing deep learning with good performance even with limited computing resources. This study uses pruning and quantization methods. After the conventional training process, additional treatment is carried out by reducing the layers and quantizing the weights to reduce the fractional numbers on each weight. At the final step, the conversion of the optimized model into a format suitable for Raspberry Pi 4 is carried out. The results of these experiment showed a significant increase in prediction latency of 2.8x faster and a decrease in file size to 13.74% smaller. This is very beneficial in the implementation of deep learning on Raspberry Pi which has minimal memory and computing capacity.

Keywords: Raspberry pi, pruning, quantization, deep learning.

1. PENDAHULUAN

Implementasi kecerdasan buatan saat ini sudah digunakan di berbagai bidang dengan berbagai jenis metode. Salah satu yang berkembang cukup pesat adalah pemanfaatan kecerdasan buatan pada perangkat portable seperti smartphone, mikrokontroller, kamera keamanan, sensor IoT, sistem robotik, drone dan lainya. Tantangan yang dihadapi dalam hal ini adalah bagaimana meningkatkan unjuk kerja sistem kecerdasan buatan, contohnya untuk keperluan pendeteksi gambar, agar diperoleh akurasi tinggi, kecepatan pemrosesan dan keamanan. Perangkat portable pada umumnya memiliki kapasitas dan spesifikasi minimal dengan memori dan CPU, serta sumber daya baterai yang terbatas. Salah satu solusi dengan menggunakan IoT, sistem komputasi awan dapat diterapkan. Namun disisi lain penerapan komputasi awan memiliki kendala dalam persepsi pengguna mengenai faktor keamanan, dan koneksi internet yang dirasakan pada saat sekarang masih belum bisa diandalkan [1]. Faktor teknis juga sangat berpengaruh dalam implementasinya antara lain: keandalan sistem, fleksibilitas, skalabilitas, virtualisasi, dan ketersediaan [2] dari penyedia layanan yang ada di Indonesia saat ini masih menjadi persoalan yang harus diperbaiki.

Received : 30 – Juli - 2024; Revised : 31 – Juli - 2024; Accepted : 22 – Agustus - 2024

p-ISSN : 2087-0868 e-ISSN : 2598-9707

Dengan kendala tersebut, diperlukan upaya agar implementasi kecerdasan buatan pada perangkat portable dapat mencapai tujuan yang diinginkan dengan segala keterbatasannya. Optimasi yang dapat dilakukan yaitu dengan membangun jaringan syaraf tiruan CNN untuk sistem pendeteksi gambar dengan arsitektur yang lebih efisien namun akurasi tetap terjaga.

Penggunaan jaringan neural konvolusi atau yang sering dikenal dengan CNN (*Convolutional Neural Network*) saat ini sudah masif digunakan untuk berbagai keperluan kecerdasan buatan mulai dari kalisifikasi data, pengenalan gambar, pengenalan pola. Dengan dukungan perkembangan sistem perangkat lunak dan perangkat keras yang memungkinkan model kecerdasan buatan dapat diterapkan pada berbagai jenis perangkat, baik dengan kapasitas dan spesifikasi maksimal maupun minimal. Unjuk kerja CNN juga semakin meningkat dengan salah satu teknik yang diterapkan dengan mengembangkan Deep Learning, yaitu menambah kedalaman lapisan jaringan syaraf tiruan sehingga ekstraksi data lebih tajam dan lebih detail. Hal ini tentu saja harus didukung dengan kemampuan komputasi yang tinggi karena *Deep Learning* memerlukan sumberdaya komputasi yang sangat besar, baik dalam proses pelatihan model maupun untuk implementasinya. Kemudahan pembuatan model kecerdasan buatan juga didukung dengan semakin banyaknya dataset yang mudah didapat, antaralain transfer learning yaitu *Deep Learning* yang berbasis *Pretrained model*, sudah dilatih dengan data gambar dalam jumlah masif, sehingga ketika kita memerlukan kecerdasan buatan dengan spesifikasi khusus, dapat dengan mudah mentransfer pembelajaran dari model yang sudah ada dengan cara mlakukan *fine tuning* untuk penyesuain dengan dataset yang kita miliki [3].

Setelah model deep learning melewati tahapan evaluasi untuk memastikan tingkat akurasi yang diinginkan, model tersebut perlu dioptimasi agar sesuai untuk diterapkan pada perangkat yang akan dikembangkatn. Pada implementasi untuk perangkat dengan kapasitas memori dan komputasi minimal seperti Raspberry Pi, sangat penting untuk menjaga akurasi namun dengan model yang lebih sederhana agar meringankan beban komputasi.Penting juga untuk mengurangi latency pada proses pendeteksian gambar agar sumber daya perangkat lebih tahan lama.

Sistem pendeteksi gambar dapat dilakukan menggunakan metode tradisional yaitu pengolahan citra dengan tujuan memperbaiki kualitas citra dan mentransformasi agar dapat diinterpretasi dengan lebih baik. Teknik ini meliputi pengolahan data awal (*preprocessing*) untuk menyesuaikan ukuran gambar dan skala intensitas pencahayaan gambar, transformasi dan filter menggunakan kernel mask untuk mengubah nilai piksesl untuk keperluan deteksi tepi, segmentasi. Pada umumnya teknik yang digunakan adalah dengan template matching. Dan deteksi obyek berbasis fitur seperti pada penggunaan Haar Cascade [4]. Salah satu metode mendeteksi gambar adalah dengan template matching yaitu mencocokkan input gambar yang diterima oleh kamera dengan sekumpulan data gambar dengan kategori yang sama. Namun metode ini memerlukan komputasi yang lama dan akurasi belum optimal [5].

Untuk memperoleh akurasi yang lebih tinggi, model deep learning dapat diandalkan, namun disisi lain, memerlukan sumberdaya yang cukup besar agar dapat berjalan secara optimal.Penelitian ini bertujuan untuk membangun model deep learning yang optimal untuk implementasi pada perangkat Raspberry 4 dengan sumberdaya memori, daya dan komputasi yang minimal. Dalam studi ini, dibangun model deep learning dengan metode transfer learning berbasis model Resnet50, VGG19, dan MobileNet yang sudah terbukti memiliki kinerja yang sangat baik, dengan kelebihan masing-masing pada faktor kecepatan proses prediksi dan tingkat akurasi. Selain dengan melakukan transfer learning, teknik optimasi juga dilakukan untuk penyesuaian dengan perangkat yang memiliki spesifikasi rendah. Strategi yang dilakukan adalah dengan menerapkan teknik pruning dan quantisasi. Diharapkan dengan metode ini, kinerja model tetap terjaga baik, namun kebutuhan sumberdaya komputasi dapat di minimalkan.

2. TINJAUAN PUSTAKA

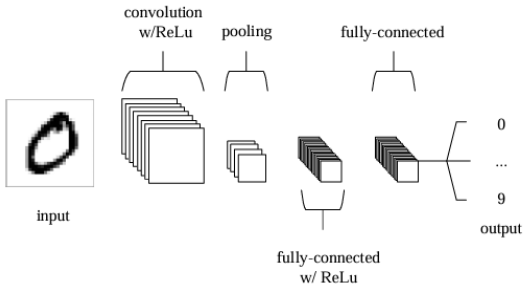
2.1. Pengolahan Citra dan Deep Learning

Pada teknik pengolahan citra, pendenalan pola dan pendeteksian gambar dilakukan menggunakan algorithma yang telah disesuaikan sehingga diperoleh output berdasarkan data input berupa gambar dan pengolahan informasi didalamnya menggunakan algoritma yang telah disusun sebelumnya. Machine Learning bekerja dengan cara berbeda yaitu tidak dengan diprogram diawal, melainkan mempelajari data input dan memperhatikan pola yang muncul menggunakan struktur statistik yang kemudian akan muncul pola yang dapat diterapkan untuk mengidentifikasi dan mengklasifikasi data sesuai tujuan. Sedangkan Deep Learning sebagai bagian dari Machine Learning mengekstrak informasi kedalam lapisan yang lebih dalam dengan merepresentasikan model pembelajaran menggunakan lapisan jaringan syaraf tiruan [3]. Spesifikasi dari lapisan tersebut di simpan dalam bentuk pengaturan bobot. Dalam konteks ini, penyesuaian bobot dilakukan melalui proses pembelajaran.

p-ISSN : 2087-0868 e-ISSN : 2598-9707

Deep learning adalah komputasi yang menirukan cara otak bekerja dalam mengenali dan mempelajari sebuah obyek. Pada jaringan syaraf tiruan dalam atau Deep Neural Network, pembelajaran neuron biologis diaplikasikan dengan menggunakan pembobotan dan fungsi aktifasi. Untuk memperoleh nilai keluaran dari sebuah neuron, sejumlah nilai dari masukan diberikan bobot dan dijumlahkan untuk di masukkan kedalamfungsi aktifasi. Keluaran neuron tersebut kemudian menjadi input bagi neuron pada lapisan berikutnya seprti diperlihatkan pada gambar 1. Jika terdapat n buah input yang masuk kedalam sebuah neuron, maka output dari neuron tersebut ditunjukkan dalam persamaan matematika pada persamaan (1) berikut ini [6] .

$$Y = \sum_{\square}^{n} W_i \cdot X_i \qquad (1)$$



Gambar 1 Blok diagram jaringan neuron konvolusional

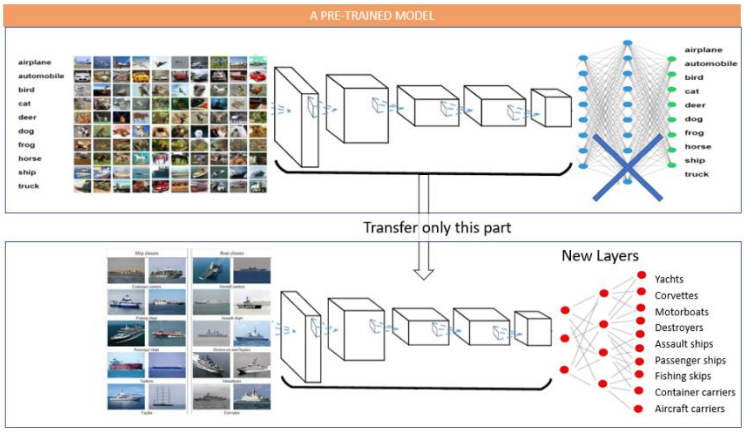
Terdapat jenis lapisan funsional sebagai berikut [7] : (a) **Lapisan input**, merupakan masukan data berupa nilai piksel dari gambar yang akan dianalisa. (b) **Lapisan Konvolusi**, menentukan nilai output yang terhubung dengan region lokal pada data input, melalui perhitungan perkalian skalar antara bobot dengan nilai input yang terhubung. Lapisan ini berfungsi untuk mengekstrak fitur dari sebuah gambar dengan menggunakan kernel dengan ukuran kecil yang bergeser ke seluruh area gambar (konvolusi). Lapisan kopnvolusi dapat terdiri dari sejumlah lapisan dengan jumlah lapisan yang semakin banyak, akan semakin akurat dalam mengekstrak data. (c) **Lapisan Pooling**, berfungsi untuk menyederhanakan dimensi dan merangkum informasi dari ekstraksi data input dalam region lokal, agar komputasi lebih cepat. Lapisan ini bekerja dengan mengambil nilai rata-rata (*average pooling*) atau nilai maksimal (*max pooling*). (d)**Lapisan Koneksi Penuh (*fully connected layer*),** yaitu lapisan yang bertanggung jawab dalam menentukan nilai akhir dengan merangkum data semua fitur yang tertangkap. Sebagai hasil akhir, lapisan ini terhubung dengan fungsi aktivasi yang menentukan skor dalam tugas klasifikasi atau regresi.

2.2 Transfer Learning

Pelatihan model deep learning memerlukan banyak sekali data input agar mampu mengenali lebih banyak fitur secara rinci. Semakin banyak fitur yang dpaat diekstrak dari banyak data sumber, akan semakin tinggi kemampuan untuk mengenali obyek pada sebuah gambar. Akan tetapi, selain memerlukan usaha yang ekstra besar, pengolahan data dengan jumlah yang masif juga memerlukan sumberdaya komputasi yang besar pula. Idealnya, pengolahan komputasi menggunakan prosesor GPU yang saat ini harganya masih mahal. Salah satu teknik alternatif untuk meningkatkan akurasi dalam proses training adalah dengan teknik *Transfer Learning*. Yaitu dengan memanfaatkan model yang telah dilatih dengan data lain yang hampir sejenis. Misalnya untuk proses pendeteksian gambar, secara umum gambar memiliki fitur yang hampir sama antara satu jenis dengan yang lain. Salah satu kumpulan data yang banyak digunakan untuk melatih model adalah imagenet yang merupakan dataset dengan 14 juta gambar lebih dan akan selalu bertambah.

Model yang sudah dilatih dengan data imagenet, memiliki probabilitas yang lebih tinggi untuk dapat mengenali fitur yang ada pada sebuah dataset baru yang akan digunakan sebagai pelatihan, dibandingkan dengan menggunaakn model yang disusun dari nol. Transfer learning diilakukan dengan mengkloning lapisan konvolusi yang sudah menyimpan bobot yang memiliki informasi fitur obyek yang beragam. Lapisan konvolusi tersebut tersimpan dalam pretrained model [8].

p-ISSN : 2087-0868 e-ISSN : 2598-9707

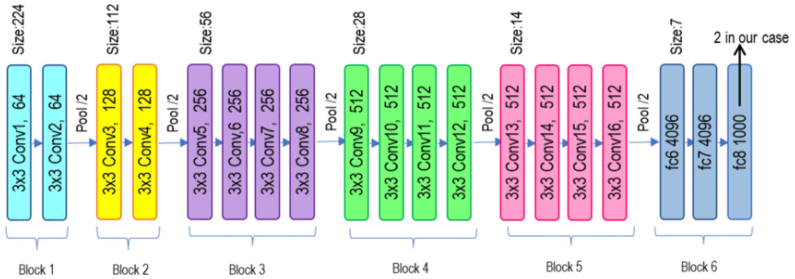


Gambar 2 Proses Transfer learning
Sumber : https://dataman-ai.medium.com/transfer-learning-for-image-classification-6-build-the-transfer-learning-model-67d87999af4a

Pada proses transfer learning ini, hanya lapisan pengektrak fitur yang diambil, dalam hal ini lapisan konvolusi, sedangkan lapisan pengklasifikasi, perlu di tambahkan lapisan baru menyesuaikan dengan tugas pengklasifikasi yang baru untuk kemudian di lakukan pelatihan ulang sebagai *fine tuning.* Pretrained model yang digunakan dalam penelitian in adalah VGG19 dan Resnet.

2.2.1 VGG 19

VGG19 merupakan deep convolutional neural network dengan jumlah layer tersembunyi sejumlah 19 sesuai dengan namanya [9]. Terdapat dua versi VGG16 dan VGG19, yang menunjukkan kedalaman lapisan konvolusi yang digunakan. Model ini menunjukkan tingkat akurasi untuk pengkalisifian deteksi gambar dengan training model dari gambar Imagenet, menunjukkan unjuk kerja yang sangat baik dari sisi kecepatan training maupun akurasi. [8]. VGG19 tersusun atas lapisan konvolusi yang diselingi dengan lapisan Maxpooling2D.

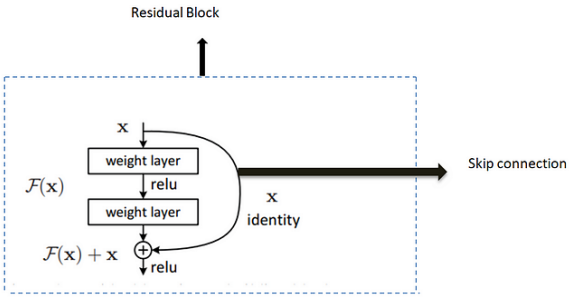


Gambar 3 Blok Lapisan Konvolusi pada VGG19 yang dipisahkan dengan lapisan Maxpool2D
Sumber gambar :

2.2.2 Resnet *(Residual Network)*

Pada saat jaringan neuron memiliki lapisan yang lebih dalam, konvergensi akan tercapai, namun disisi lain akan terjadi penurunan kinerja, yaitu dengan kedalaman lapisan yang meningkat, akan terjai kejenuhan pada akurasi. Masalah penurunan tersebut diatasi dengan kerangka pelatihan residual. Penyesuaian bobot tidak hanya mengandalkan pada lapisan dibawahnya namun pemetaan bobot dilakukan melalui penyesuaian dengan pemetaan residual. Hal ini dilakukan dengan memetakan hubungan langsung antar lapisan yang tidak berdekatan. Dengan demikian bobot pada lapisan awal dapat mempengaruhi secara langsung terhadap output pada dimensi yang sama [10].

p-ISSN : 2087-0868 e-ISSN : 2598-9707



Gambar 4 Jaringan residual
Sumber gambar : Deep Residual Learning for Image Recognition

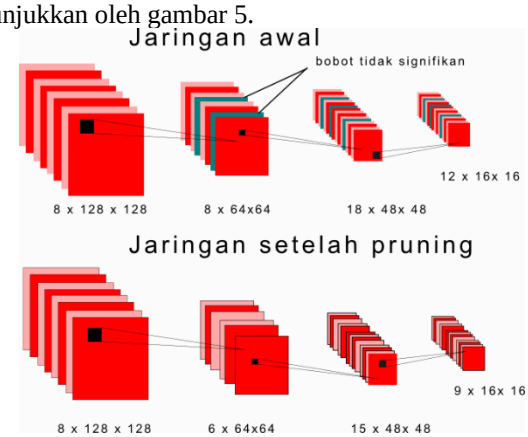
Keuntungan dengan mengimplementasikan jaringan residual ini adalah: (1) Pelatihan lapisan tersembunyi menjadi lebih mudah, karena dengan lapisan yang sangat dalam, terdapat alternatif jalan pintas untuk optimasi gradien selama pelatihan. (2). Dengan adanya jalan pintas, konvergensi menjadi lebih cepat terjadi. (3) akurasi menjadi lebih meningkat karena dengan semakin dalam lapisan, ekstraksi fitur menjadi lebih detail.

2.3 Teknik Optimasi Model Deep Learning

Optimasi model deep learning untuk implementasi pada perangkat portable bertujuan agar model tidak hanya akurat tetapi juga efisien dalam penggunaan penyimpanan memori, pemakaian sumberdaya unit pemrosesan, penggunaan baterai dan kecepatan inferensi, karena memang perangkat jenis ini memiliki sumberdaya komputasi , daya dan memori yang terbatas. Pendekatan optimasi yang dapat dilakukan adalah dengan meminimalkan struktur model deep learning yang akan digunakan, dengan tidak mengurangi fungsionalitas masing-masing elemen yang ada dalam model deep learning. Teknik optimalisasi dapt dilakukan dengan beberapa pendekatan berikut ini :

2.3.1 Pruning

Pruning dilakukan dengan cara mengeliminasi lapisan atau kernel yang memiliki signifikasi rendah, yaitu lapisan dengan nilai bobot mendekati nol atau dibawah nilai tertentu. Secara teori pruning dapat dilakukan pada bobot individual ataupun secara terstruktur untuk keseluruhan lapisan [11]. Sedangkan dlaam meilih lapisan mana yang akan di eliminasi dapat dilakukan dengan pendekatan statistik, misalnya dengan memilih nilai rata2 aktivasi, distribusi normal, persentase. Secara arsitektur, model deep learning tidak berubah. Proses pruning ditunjukkan oleh gambar 5.



Gambar 5 jaringan neural network dengan proses pruning

2.3.2 Quantizing

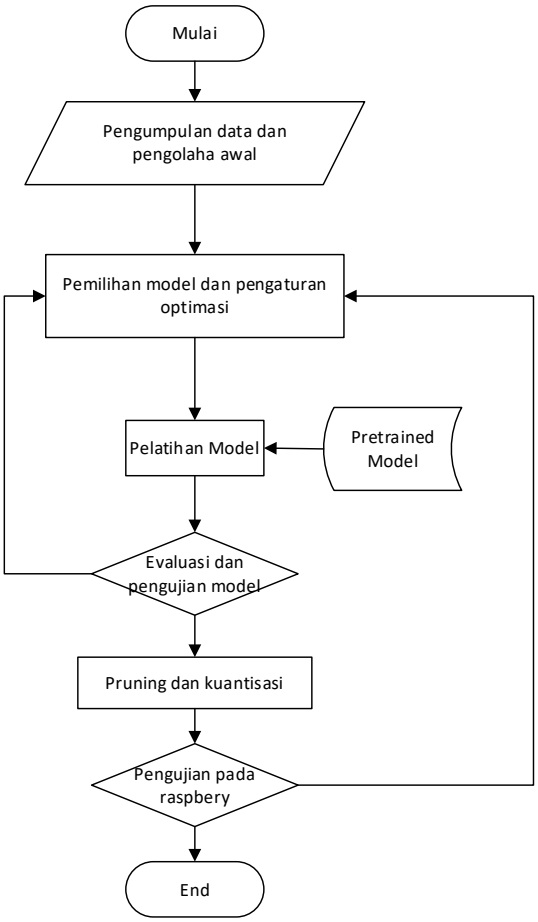
Proses quantisasi adalah mengurangi presisi pada nilai bobot, pada umumnya dari data tipe float 16 bit menjadi 8 bit. Quantisasi juga dapat dilakukan dengan mengubah nilai tipe data float menjadi tipe integer. Dengan demikian, kecepatan komputasi dan penggunaan memori akan lebih efisien [12].

3. METODOLOGI PENELITIAN

p-ISSN : 2087-0868 e-ISSN : 2598-9707

Tahapan penelitian yang dilakukan ditunjukkan oleh flowchart pada gambar 6. Tahapan tersebut meliputi :

1. Pengumpulan dan preprocessing data. Proses ini bertujuan untuk menyusun dataset, yaitu sekumpulan data gambar yang telah dilabeli dengan kelas kategori sesuai gambar. Dalam hal ini digunakan gambar wajah dengan masker dan tidak memakai masker, atau sejumlah 2 kelas. Setelah data terkumpul, data dibagi menjadi 3 bagian, yaitu untuk ekperluan training, validasi dan testing. Agar hasil pendeteksian gambar lebih akurat, gambar dipilih secara seimbang, dengan jumlah masih-masing kelas sama.



Gambar 6 Flow Chart proses optimasi model *deep learning*

2. Pembuatan model dengan base model pretrained VGG19 dan Resnet50
   VGG19 dan Resnet merupakan arsitektur yang sudah memiliki komponen lapisan konvolusi, max pooling, dan lapisan pengklasifikasi yang sudah baku. Dari struktur model tersebut, kemudian dilatih dengan dataset dari imagenet, yang merupakan koleksi gambar yang telah dilabeli, sejumlah 14 juta gambar dengan 1000 kelas kategori. Dengan demikian lapisan konvolusi yang sudah terlatih sudah mampu mengenali fitur gambar secara rinci. Agar dapat mengenali data sesuai klasfikasi yang sudah ditentukan pada langkah 1 diatas, maka dilakukan fine tuning dengan melatih kembali model tersebut dengan dataset spesifik yang telah dipersiapkan. Dalam hal ini lapisan-lapisan yang diambil adalah lapisan konvolusi saja, tidak termasuk lapisan pengklasifikasi. Sehingga perlu ditambahkan kembali lapisan pengklasifikasi dengan susuanan sebagai berikut :
   - Lapisan Flatten, merupakan lapisan untuk mengkondisikan peta fitur dari model pretrained agar bisa masuk kek lapisan berikutnya dengan mengubah lapisan multidimensi kedalam bentuk vektor pada lapisan *fully connected* setelahnya.
   - Lapisan Dense, merupakan lapisan yang merangkum semua
   - Lapisan Dropout, yaitu lapisan yang bekerja untuk mengeliminasi fitur secara random agar tidak terjadi overfitting.
   - Lapisan Dense, merupakan lapisan akhir yang menjadi pengklasifikasi gambar, dengan menggunakan fungsi aktivasi sigmoid.

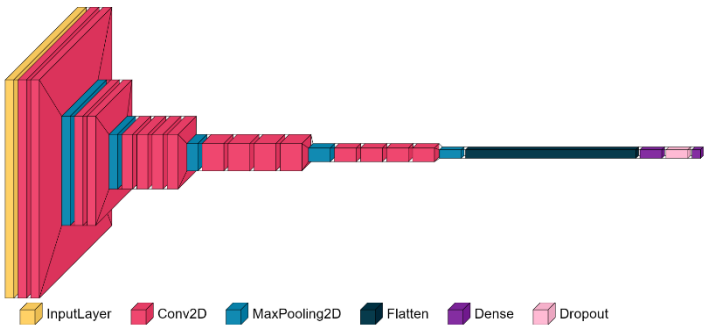
p-ISSN : 2087-0868 e-ISSN : 2598-9707

Untuk menghasilkan pelatihan yang efisien, optimasi pelatihan dilakukan dengan menambahkan optimizer Adam (*Adaptive Moment Estimation*), yaitu algoritma optimasi iteratif untuk meminmalisasi fungsi loss.

3. Pelatihan dan validasi
   Pada proses pelatihan, dataset di bagi menjadi 3 kategori, Masing-masing digunakan untuk keperluan berbeda, yaitu untuk keperluan pelatihan, validasi dan test. Pelatihan, dilakukan secara bertahap, menyesuaikan kemampuan komputasi komputer yang digunakan untuk pelatihan dengan batchsize 32, jumlah epoch 8. Dalam pelatihan karena data gambar yang digunakan cukup banyak, batchsize diperlukan untuk membagi data ke dalam beberapa batch sehingga tidak terlalu membebani komputasi pada saat training. Pelatihan dilakukan secara iteratif sejumlah epoch, namun dengan tetap memonitor akurasi dan loss. Jika akurasi dan loss sudah mengalami kejenuhan, pelatihan segera dihentikan untuk mencegah overfitting. Pada setiap epoch, metrik kinerja pelatihan juga dicatat menggunakan fungsi callback tensorboard untuk keperluan analisa.
4. Proses optimasi dengan pruning dan quantisasi
   - proses pruning : model di analisa tingkat sparsity yaitu berapa persen bobot memiliki sgnifikansi. Bobot dengan nilai mendekati 0 akan dieliminasi. Setelah itu dilakukan finetuning kembali dengan dataset test.
   - Proses quantisasi : mengubah tipedata floating 32 bit menjadi tipe data 8 bit.
5. Pengujian pada raspberry
   Untuk dapat diuji pada perangkat reaspberry pi, maka model yang sudah terbangun, dirubah dari format h5 ke dalam format tensorflow lite yang (*.tflite) yang dapat dijalankan pada sistem raspberrypi4. Proses ini dilakukan menggunakan konverte pada tensorflowlite.
6. Membandingkan hasil setelah optimasi dan sebelum optimasi

4. HASIL DAN PEMBAHASAN

Model deep learning dibangun dari pretrained model dengan mengganti top layer, dari lapisan model asal, diganti dengan lapisan Flatten, Dense dan Dropout menyesuaikan dengan kelas dataset yang ada. Setelah model terbentuk, berikut visualisasi model digambarkan pada gambar 7 untuk model dengan basis VGG19 dan gambar 8 untuk model dengan basis Resnet50



Gambar 7 Visualisasi model hasil transfer learning menggunakan VGG19

Gambar 8 Visualisasi model hasil transfer learning menggunakan bas model Resnet50

Pada Gambar 7 dan 8 diatas, ditunjukkan hasil visualisasi model transfer learning VGG19 dan Resnet50. VGG19 memiliki 19 lapisan convolusional dan diantaranya terdapat lapisan maxpooling2D. Resnet50, memiliki jumlah lapisan 50, lapisan konvolusional yang berada dalam blok residual, dan 1 lapisan fully connected layer.

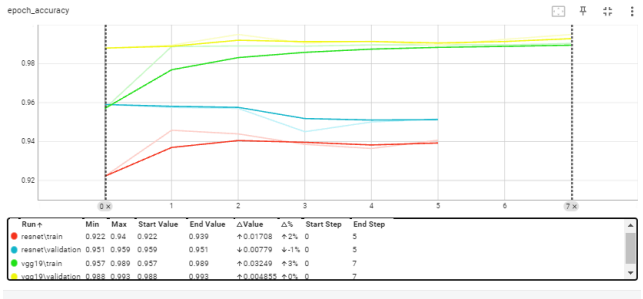
Langkah selanjutnya setelah terbentuk model dengan penyesuaian lapisan akhir untuk klasifikasi gambar pendeteksi wajah menggunakan masker, adalah proses pelatihan. Dalam proses ini diberikan dataset berupa gambar data dengan dua kelas. Untuk 2 kelas kategori data dapat digunakan pengklasifikasi Kategori atau Biner. Dataset yang adaa, terdiri dari 10000 gambar, dibagi kedalam dua subset, satu set untuk proses training dan sisanya untuk proses validasi, dengan perbandingan 80%:20%. Untuk meningkatkan akurasi,

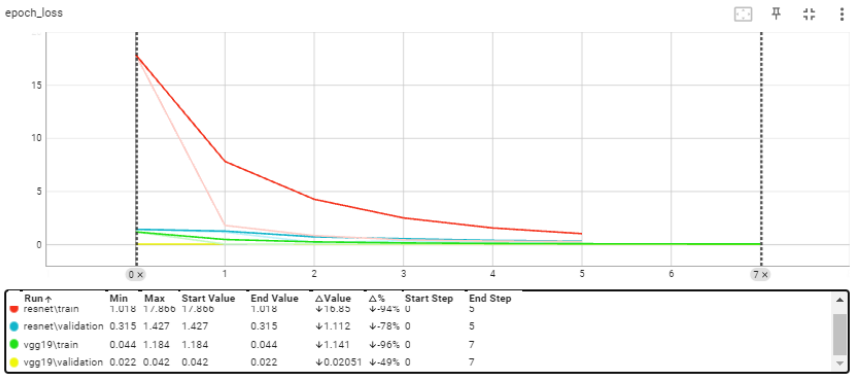
p-ISSN : 2087-0868 e-ISSN : 2598-9707

telah dilakukan augmentasi berupa operasi transformasi citra untuk menambah variasi gambar dengan tujuan meningkatkan akurasi [13]. Perlakuan tambahan sebelum masuk ketahap pelatihan adalah melakukan augmentasi data pada dataset dengan perlakuan rotasi, pergeseran, zoom, dan horizontal flip secara acak. Gambar 9 berikut ini menunjukkan hasil proses augmentasi.

Gambar 9 Proses Augmentasi Gambar

Gambar 10. Menunjukkan hasil akurasi dan loss yang dilakukan pada saat pelatihan data dasar. Pada inisiasi awal, epoch diseting pada nilai 8, namun pada model Resnet, telah terjadi konvergensi pada epoch ke 5 sehingga pelatihan dihentikan karena adanya pengaturan *early stop* , yaitu penghentianproses training jika terjadi konvergensi atau tidak adanya kenaikan akurasi dan penurunan loss secara signifikan.





p-ISSN : 2087-0868 e-ISSN : 2598-9707

Gambar 10 Akurasi dan loss hasil pelatihan dengan model hasil *finetuning*

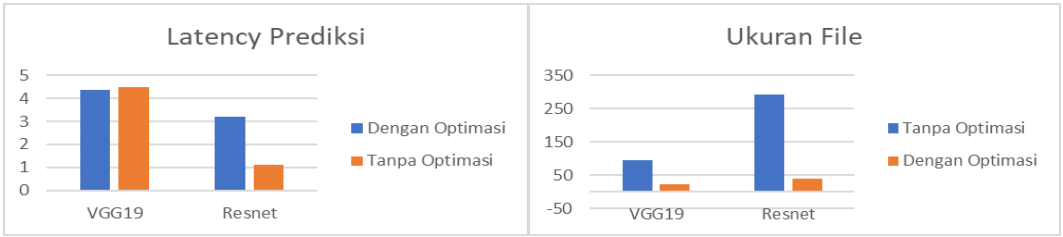
Gambar 11 Hasil pengujian dari beberapa sample gambar

Setelah terbentuk model dengan akurasi yang sudah baik, dilakukan pengujian prediksi dengna memasukkan gambar secara acak. Tampak pada gambar 11, terdapat 1 gambar yang mengalami kesalahan pendeteksian. Hal ini wajar karena memang akurasi tidak 100%, sehingga masih ada kemungkinan kecil terjadi kesalahan pendeteksian.

Sebagai pengujian akhir, dilakukan uji prediksi dengan 150 buah gamabr input dilakukan pada perangkat Raspberry pi 4. Hasilnya ditunjukkan pada tabel berikut ini .

Tabel 1. Hasil Uji coba prediksi model pada Raspberry pi 4

Keterangan	Model	Accuracy	loss	Latency (detik)	Ukuran File (KByte)
Tanpa Optimasi	VGG19	1	0	4,3748	94,622
	Resnet	1	0	3,2024	292,599
Dengan Optimasi	VGG19	1	0	4,4706	23,81
	Resnet	1	0	1,11791	40,21



Gambar 12 Grafik hasil uji coba pada perangkat raspberry pi4

Grafik diatas menunjukkan efek perubahan pada pretrained model yang telah dioptimasi. Berdasarkan latency prediksi, yaitu terjadinya penurunan yang sangat signifikan pada latensi prediksi yang merupakan lama waktu pemrosesan data dari saat diterima data sampai dengan keluar hasil prediksi. Hasil yang baik

didapat pada model dengan pretrained Resnet50 yang telah diptimasi. Sedangkan efek pada pengurangan ukuran file juga terjadi baik pada VGG19 maupun Resnet50.
Parameter latensi prediksi dan ukuran file bagi perangkat portabel seperti Raspberry pi4. Dengan kecepatan pemrosesan data, akan memberikan dampak bagus bagi pengguna dan merupakan indikator bahwa komputasi lebih ringan dibandingkan sebelum optimasi. Ukuran file yang lebih kecil juga akan sangat bermanfaat dalam menghemat pemakaian memori pada perangkat.

5. KESIMPULAN DAN SARAN

Optimasi model deep learning sangat bermanfaat dalam mengoptimalkan kinerja perangkat portable seperti raspberry, drone, kamera keamanan dan alat sejenis lain. Dengan hasil yang diperoleh diatas dapat disimpulkan :

1. Pemilihan pretrained model yang tepat sangat penting dalam meningkatkan akurasi dan kecepatan pemrosesan deteksi gambar
2. Optimasi model dapat dilakukan dengan cara mengeliminasi lapisan yang memiliki bobot kecil dengan signifikansi rendah dan quantisasi, dengan tetap mempertahankan tingkat akurasi.
3. Implementasi deep learning pada perangkat portable sangat mungkin dilakukan dengan mengkonversi model hasil pelatihan dan fine tuning dari pretrained model, ke dalam format yang sesuai dengan perangkat yang dibutuhkan.

Dengan keberhasilan implementasi penerapan deep learningpada perangkat Raspberry pi 4 ini, diharapkan kedepannya dapat dilakukan penelitian lebih lanjut dengan mengimplementasikan deep learning untuk tujuan yang lebih luas, dan juga dapat dilakukan penelitian untuk mengoptimasi model deep learning pada perangkat lain seperti arduino, mikrokontroller, FPGA dan perangkat portabel lain.

6. DAFTAR PUSTAKA

[1] R. I. Cahyani, "ANALISIS FAKTOR-FAKTOR YANG MEMPENGARUHI PENGGUNA DALAM MENGADOPSI TEKNOLOGI KOMPUTASI AWAN (CLOUD STORAGE) DI KOTA SEMARANG," Universitas Katolik Sugiyopranoto, Semarang, 2021.

[2] B. W. Rizdawaty and H. Mustafidah, "Faktor-Faktor yang Mempengaruhi Adopsi Cloud oleh Instansi Pemerintah: Tinjauan Pustaka Sistematis," *Sainteks,* vol. 18, no. 2, 2021.

[3] F. Chollet, Deep Learning with Python, New York: Manning Publication Co., 2018.

[4] F. T. Nugroho, "Deteksi Citra Wajah Menggunakan Algoritma Haar Cascade Classifier," *Jurnal Media Teknologi dan Informasi,* vol. 1, no. 1, Januari 2024.

[5] Moch.Zamroni, H. Fitriyah and R. Maulana, "Sistem Pendeteksi Penyakit Daun Bawang Merah Probolinggo Menggunakan Metode Template Matching Berbasis Raspberry Pi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* pp. 6026-6031, Desember 2018.

[6] L. Z. J. H. A. Alzubaidi, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data,* p. 53, 2021.

[7] K. O'Shea and R. Nash, "AnIntroduction to Convolutional Neural Networks," *arXiv,* vol. 1511, no. 08458v2 , 2015.

[8] C. Raju, "HybridTransferNet: Advancing Soil Image Classification through Comprehensive Evaluation of Hybrid Transfer Learning".

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv,* vol. 1409, no. 1556, 2015.

[10] K. He and X. Zhang, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2016.

[11] D. A. T. B.-N. N. D. A. P. Torsten Hoefler, "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks," *Arxiv,* vol. 2102, no. 00554, 2021.

[12] B. Hawks and J. Duarte, "Ps and Qs: Quantization-Aware Pruning for Efficient Low Latency Neural Network Inference," *Frontiers in Artificial Intelligence,* vol. 4, p. 676564, 2021.

[13] I. Harjanto, M. Amiruddin and Margono, "PENGARUH AUGMENTASI DATA TERHADAP UNJUK KERJA MODEL PENDETEKSI WAJAH MENGGUNAKAN CNN," in *PROSIDING*

p-ISSN : 2087-0868 e-ISSN : 2598-9707

SENANTIAS: Seminar Nasional Hasil Penelitian dan Pengabdian kepada Masyarakat, Pamulang, Indonesia, 2023.

[14] Keras, "Keras," 2 10 2020. [Online]. Available: https://keras.io/getting_started/intro_to_keras_for_researchers/#keras-layers.

[15] E. Kristiani, C.-T. Yang and K. L. P. Nguyen, "Optimization of Deep Learning Inference on Edge Devices," in *2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*, 2020.

[16] B. A. I. Research, "Caffe Deep Learning Framework," 2023. [Online]. Available: https://caffe.berkeleyvision.org/tutorial/.

[17] B. K. Mukti, "A Comprehensive Analysis of Mask Detection Using Convolutional Neural Networks (CNN) and Single Shot Multibox Detector (SSD) Approach," in *2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*, Ghaziabad, India, 2023.

[18] E. Assunção and P. D. Gaspar, "Real-Time Image Detection for Edge Devices: A Peach Fruit Detection Application," *Future Internet*, vol. 14, no. 11, 8 November 2022.