

Deteksi Citra Digital Penyakit Cacar Monyet menggunakan Algoritma *Convolutional Neural Network* dengan Arsitektur *MobileNetV2*

Digital Image Detection of Monkeypox Disease using Convolutional Neural Network Algorithm with MobilenetV2 Architecture

PUTRI SARAH FRANSISCA¹, NURHAFIFAH MATONDANG^{1*}

Abstrak

Pada bulan Juli tahun 2022, penyakit cacar monyet ditetapkan sebagai darurat kesehatan global. Hal ini dikarenakan penyakit cacar monyet sudah terjadi di lebih dari 70 negara. Kasus cacar monyet di Indonesia ditemukan pertama kali pada bulan Agustus 2022 di Kota Jakarta. Kesamaan gejala yang dimiliki oleh penyakit cacar monyet, cacar air, dan campak menjadi tantangan untuk para tenaga kerja kesehatan dalam membedakan penyakit tersebut. Penelitian ini mengembangkan sebuah model algoritma deteksi otomatis untuk mendeteksi citra digital penyakit cacar monyet. Algoritma tersebut adalah *convolutional neural network* dengan arsitektur *MobileNetV2* serta penerapan *transfer learning*. Pelatihan model dilakukan dengan jumlah 5 *epoch* dan mengimplementasikan dua jenis *optimizer*, yaitu Adam dan RMSprop. Penerapan Adam *optimizer* dengan *learning rate* 10^{-4} menghasilkan akurasi pada data uji sebesar 94%, akurasi pada data latih sebesar 92%, dengan nilai *loss function* 27%. Hasil yang berbeda dihasilkan oleh penerapan RMSprop *optimizer* dengan *learning rate* 45×10^{-3} menghasilkan akurasi pada data uji sebesar 97%, akurasi pada data latih mencapai 97% namun nilai *loss function* cukup tinggi yaitu mencapai 52%. Hasil ini menunjukkan bahwa Adam *optimizer* lebih efektif dalam menyempurnakan parameter model untuk mengoptimalkan deteksi gambar cacar monyet selama pelatihan.

Kata Kunci: cacar monyet, *convolutional neural network*, *MobileNetV2*, pengenalan gambar, penyakit lesi kulit.

Abstract

In July 2022, monkeypox was declared a global health emergency due to its occurrence in over 70 countries. The first case of monkeypox in Indonesia emerged in Jakarta in August 2022. The challenge faced by healthcare workers in distinguishing between monkeypox, chickenpox, and measles—ailments sharing similar symptoms—prompted the initiation of a research study. This study aimed to develop an automated algorithm for detecting digital images of monkeypoxes. The algorithm used was a convolutional neural network with MobileNetV2 architecture, implementing transfer learning. The model was trained for a total of 5 epochs and utilized two types of optimizers: Adam and RMSprop. Applying the Adam optimizer with a learning rate of resulted in a test accuracy of 94%, training accuracy of 92%, and a loss function value of 27%. Conversely, the implementation of the RMSprop optimizer, with a learning rate of , resulted in a test accuracy of 97%, and a training accuracy of 97%, albeit with a slightly higher loss function value of 52%. The results indicate that the Adam optimizer may be more effective in fine-tuning the model parameters to optimize the detection of monkeypox images during training.

Keywords: convolutional neural network, image recognition, MobileNetV2, monkeypox, skin lesion disease.

¹ Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jakarta, Depok 12450;

* Penulis Korespondensi: Tel: 085718827290; Surel: nurhafifahmatondang@upnvj.ac.id

PENDAHULUAN

Wabah cacar monyet mulai menyebar di beberapa negara dan menyebabkan kekhawatiran kepada masyarakat di dunia. Cacar monyet adalah penyakit yang disebabkan oleh *virus human monkeypox* (MPXV) *orthopoxvirus* dari famili *poxviridae* yang bersifat *highlipatogenik* atau *zoonosis* (Unit Pelayanan Kesehatan Kementerian Kesehatan 2022). Organisasi Kesehatan Dunia atau *World Health Organization* (WHO) menetapkan penyakit cacar monyet berstatus darurat kesehatan global sejak bulan Juli 2022. Hal ini dikarenakan penyakit cacar monyet telah menyebar ke lebih dari 70 negara dan sudah termasuk ke dalam kriteria darurat kesehatan global. Kasus pertama cacar monyet di Indonesia sendiri telah ditemukan di Jakarta. Wabah ini awalnya teridentifikasi di benua Afrika lalu pada Agustus 2022, dikabarkan telah menginfeksi seorang pria asal Jakarta (CNBC Indonesia 2022).

Berdasarkan data dari *Center of Disease Control and Prevention* (CDC) atau Pusat Pengendalian dan Pencegahan Penyakit, gejala yang akan timbul jika seseorang terinfeksi penyakit cacar monyet adalah timbulnya ruam pada kulit bagian tangan, kaki, dada, wajah serta anggota tubuh lainnya (CDC Government 2022). Gejala-gejala tersebut memiliki kesamaan dengan penyakit cacar pada umumnya karena penyakit ini berasal dari famili yang sama. Kemiripan gejala tersebut menjadi tantangan untuk para tenaga kerja kesehatan dalam membedakan penyakit cacar monyet dengan penyakit cacar biasa, cacar air, dan campak.

Berbagai penerapan *Artificial Intelligence* (AI), khususnya pendekatan dengan *deep learning*, telah banyak digunakan dalam melakukan tugas analisis citra medis. Penelitian terkait yang membahas mengenai deteksi citra penyakit kulit dengan mengimplementasikan *deep learning* adalah penelitian yang dilakukan oleh Ahsan *et al.* (2022). Pada penelitian tersebut dihasilkan sebuah *monkeypox dataset* dengan menerapkan model algoritma *deep learning* untuk mendeteksi citra digital cacar monyet. Penelitian tersebut menggunakan algoritma VGG16 yang merupakan modifikasi dari arsitektur *Convolutional Neural Network* (CNN). Penelitian ini melakukan dua percobaan dalam melakukan pengujian terhadap model algoritma yang dibuat, yaitu percobaan pertama adalah dengan menggunakan 90 data dan menghasilkan akurasi sebesar 90%. Percobaan kedua menggunakan 1754 data dan menghasilkan akurasi sebesar 88%. Penelitian yang dilakukan oleh Ali *et al.* (2022) juga membuat sebuah model *deep learning* untuk mendeteksi penyakit cacar monyet dengan menerapkan algoritma VGG16 serta dikomparasi dengan algoritma ResNet50 dan InceptionV3. Dari hasil komparasi akurasi model algoritma yang telah dicoba, didapatkan akurasi terbesar adalah algoritma ResNet50 dengan akurasi sebesar 82.96%.

Penelitian terkait arsitektur dalam penerapan algoritma CNN dilakukan oleh Huu *et al.* (2022) membuat sebuah model deteksi otomatis jika seseorang menggunakan masker atau tidak menggunakan masker. Penelitian ini menerapkan arsitektur MobileNetV2 sebagai pengklasifikasi gambar dan beberapa arsitektur CNN lainnya sebagai komparasi hasil akurasi. Hasil akurasi yang didapatkan sebesar 99.37%. Penelitian terkait juga dilakukan oleh Gulzar (2023) dalam pengklasifikasian citra digital buah yang diperoleh dari dataset public, yaitu kaggle.com menggunakan algoritma CNN, dengan arsitektur MobileNetV2. Berbeda dari penelitian lainnya, pada penelitian ini dilakukan *transfer learning* yang membuat hasil akurasi naik 9% dari akurasi sebelum diterapkan *transfer learning* pada arsitektur MobileNetV2, yaitu dari 90% menjadi 99%.

Dengan melihat kelebihan dari arsitektur MobileNetV2 yang memiliki penggunaan daya komputasi yang rendah untuk dijalankan sehingga membuatnya sangat cocok untuk diterapkan pada perangkat seluler dan komputer untuk berjalan tanpa GPU serta urgensi dari kondisi penyebaran wabah penyakit cacar monyet di Indonesia maupun dunia, maka melalui penelitian ini akan dikembangkan sebuah model algoritma deteksi otomatis untuk mendeteksi citra digital penyakit cacar monyet. Model algoritma yang digunakan adalah CNN dengan mengimplementasikan arsitektur *MobileNetV2*.

METODE

Identifikasi Masalah

Tahapan ini merupakan proses identifikasi poin permasalahan yang diangkat yaitu tentang bagaimana implementasi algoritma *Convolutional Neural Network* (CNN) dalam mendeteksi citra digital penyakit cacar monyet menggunakan data publik yang diperoleh dari *github.com* dengan menggunakan arsitektur *MobileNetV2*.

Studi Literatur

Pada tahap ini dilakukan pencarian beberapa literatur untuk mendapatkan referensi dan perbandingan yang sesuai dengan penelitian. Literatur yang dicari memiliki kata kunci sebagai berikut: *deep learning*, *cacar monyet*, *image recognition*, *convolutional neural network*, dan *MobileNetV2*.

Akuisisi Data

Pada tahap ini dilakukan akuisisi data yang diperoleh dari dataset yang bersifat *open source*. Data yang digunakan berjumlah 135 sampel data *original* yang terdiri dari kelas cacar monyet (*monkeypox*), cacar air (*chickenpox*), campak (*measles*), dan normal. Dataset dapat diakses melalui link <https://github.com/mahsan2/Monkeypox-dataset-2022>.

Praproses Data

Pada tahap ini dilakukan praproses data yaitu mengubah format gambar, melakukan *resize* gambar, mengaugmentasikan data gambar, dan melakukan inisialisasi data dan *list* label. Teknik augmentasi data ini akan meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru.

Pembuatan Model Arsitektur MobileNetV2

Pada tahap ini dilakukan pembuatan model arsitektur MobileNetV2 dengan menggunakan API Keras Tensorflow serta menerapkan teknik *Transfer Learning* (TL). *Transfer learning* adalah penggunaan pengetahuan yang telah diperoleh dari sebuah tugas dan dataset lain (bahkan yang tidak terkait erat dengan tugas sumber atau dataset yang akan digunakan) untuk mengurangi *learning costs* (Iman *et al.* 2022). Teknik *transfer learning* digunakan untuk melakukan klasifikasi kategorikal menggunakan MobileNetV2 dengan memanfaatkan *pre-trained* model. *Pre-trained* model sendiri terbagi menjadi dua bagian besar, yaitu *base model* dan *head network*. Pada penelitian ini digunakan *base model* dari MobileNetV2, lalu mengubah bagian *head network* agar sesuai dengan kebutuhan klasifikasi citra cacar monyet dan kelas lain yang ada pada dataset.

Berdasarkan penelitian Sandler *et al.* (2018) yang mengembangkan arsitektur MobileNetV2, arsitektur ini memiliki susunan seperti pada Gambar 1. Simbol t adalah faktor ekspansi, c adalah jumlah *output channel*, n adalah angka berulang, s adalah *stride*. Selanjutnya untuk konvolusi spasial digunakan kernel 3×3 .

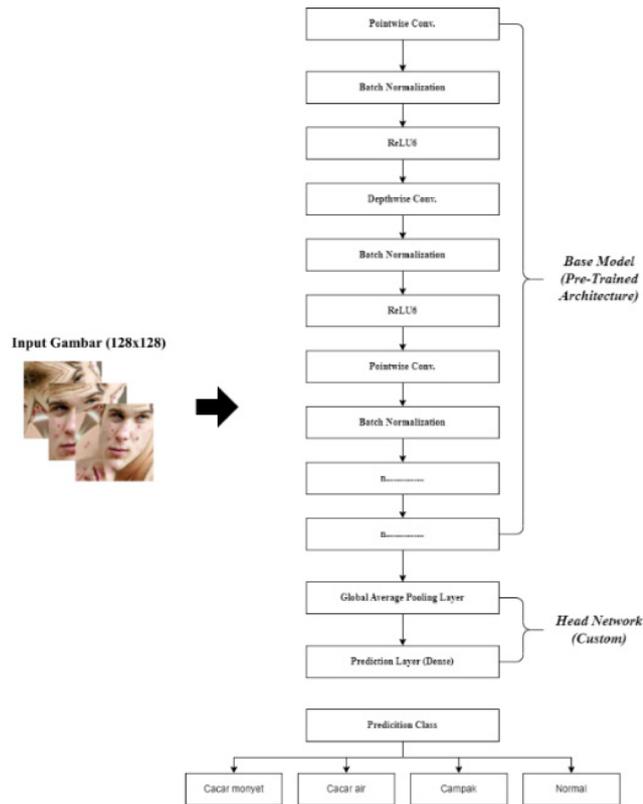
Input	Operator	t	c
$224^2 \times 3$	Conv2d	-	32
$112^2 \times 32$	Bottleneck	1	16
$112^2 \times 16$	Bottleneck	6	24
$56^2 \times 24$	Bottleneck	6	32
$28^2 \times 32$	Bottleneck	6	64
$14^2 \times 64$	Bottleneck	6	96
$14^2 \times 96$	Bottleneck	6	160
$7^2 \times 160$	Bottleneck	6	320

Gambar 1 Arsitektur MobileNetV2 (Sandler *et al.* 2018)

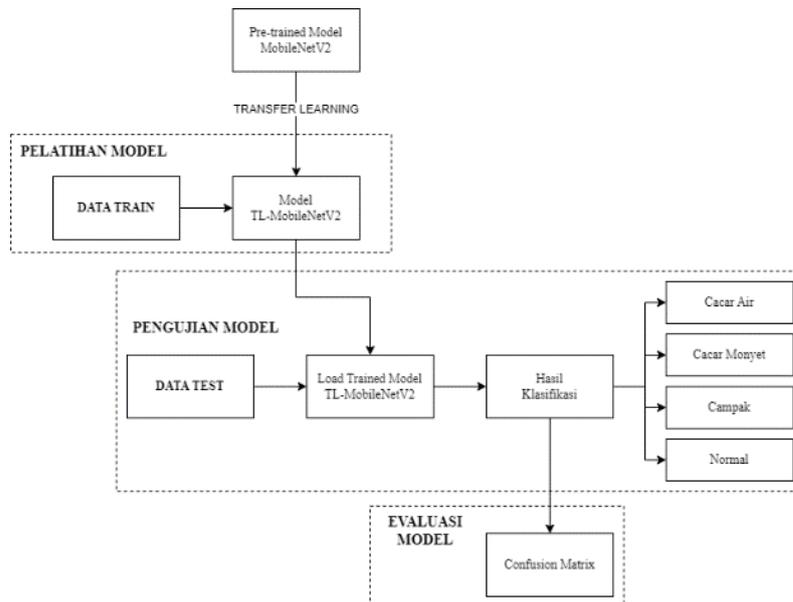
Dengan diterapkannya *transfer learning* pada model arsitektur ini, bagian *head network* akan diubah dengan menyesuaikan dataset citra cacar monyet yang digunakan pada penelitian ini. Gambar 2 menunjukkan arsitektur MobileNetV2 yang akan dibuat pada penelitian ini dengan menerapkan *transfer learning*.

Pelatihan Model

Pada tahap ini dilakukan proses pelatihan pada model untuk mendapatkan hasil performa dari model yang akan dibuat. Pembagian data latih dan data uji sebanyak 80% dari total dataset untuk data latih dan 20% untuk data uji. Tahap pelatihan model yang diusulkan ditunjukkan pada Gambar 3.



Gambar 2 Arsitektur MobileNetV2 dengan *transfer learning*



Gambar 3 Tahap pelatihan, pengujian dan evaluasi model

Pengujian Model dan Evaluasi Model

Pengujian model dilakukan dengan menggunakan data uji. Selanjutnya, hasil pengujian akan di evaluasi. Evaluasi performa model secara keseluruhan akan diukur dan disajikan menggunakan pendekatan statistik yaitu *accuracy*, *precision*, *recall*, *F1 score*, *sensitivity*, dan *specificity*.

Pengambilan Kesimpulan

Tahapan ini membahas mengenai kesimpulan berdasarkan hasil dari penelitian yang telah dilakukan. Pembahasan tersebut akan menjelaskan mengenai hasil pembuatan model CNN dengan arsitektur MobileNetV2 dan bagaimana performanya. Evaluasi performa model dilakukan dengan melihat hasil akurasi, *precision*, dan *recall*.

HASIL DAN PEMBAHASAN

Akuisisi Data

Dataset berisi citra digital bagian tubuh manusia yang terinfeksi penyakit kulit dengan jenis virus *orthopoxvirus* (cacar air, campak, cacar moyet) dan bagian tubuh manusia normal. Dataset terbagi menjadi 4 kelas yaitu citra kulit normal, citra kulit yang terinfeksi cacar monyet, cacar air, dan campak dengan total keseluruhan data *original* adalah 135 citra digital (Ahsan *et al.* 2022). Kelas cacar air memiliki data yang sangat sedikit dibandingkan dengan jumlah data dari kelas lainnya sehingga dilakukan penambahan citra pada kelas cacar air dengan jumlah 15 data citra baru yang diperoleh dari dataset yang berbeda. Jumlah citra dari masing-masing kelas yang ada pada dataset ini dapat ditunjukkan pada Tabel 1. Gambar 4 menunjukkan tampilan citra pada setiap kelasnya.

Tabel 1 Jumlah citra pada tiap kelas

Kelas	Jumlah dataset <i>original</i>	Jumlah dataset setelah penambahan
Cacar Monyet	43	43
Cacar Air	11	26
Campak	27	27
Normal	54	54
Total	135	150



Gambar 4 Tampilan citra pada tiap kelasnya

Praproses Data

Dengan jumlah dataset yang hanya memiliki 150 sampel data tentunya sangat sedikit untuk melatih sebuah jaringan *neural network* seperti MobileNetV2 yang termasuk ke dalam *deep learning*. Oleh karena itu, dilakukan proses augmentasi data yang akan memperbanyak data yang sudah ada dengan cara memodifikasi atau memanipulasi data citra tanpa benar-benar mengubah data citra aslinya. Beberapa proses yang dilakukan pada tahap ini adalah sebagai berikut:

- Konversi format gambar. Pada tahap ini dilakukan perubahan data citra yang memiliki format memiliki format JFIF (*JPEG File Interchange Format*) menjadi format JPG (*Joint Photographic Group*).
- Resize gambar. Pada tahap ini dilakukan penyamaan seluruh ukuran gambar menjadi satu ukuran yaitu 128×128 .
- Augmentasi data. Mengaugmentasikan data yang ada agar lebih banyak dan bervariasi dengan menggunakan API dari Keras Tensorflow yaitu *ImageDataGenerator*. Tabel 2 menunjukkan nilai parameter augmentasi data yang digunakan.
- Under sampling data* dan list label. Melakukan proses *under sampling* pada kelas yang memiliki jumlah data yang terlalu besar dibandingkan dengan kelas lainnya, yaitu kelas normal dan kelas cacar monyet. *Under sampling* ini dilakukan untuk menangani kasus *imbalanced data* atau ketidakseimbangan data pada dataset, dengan tujuan agar model yang akan dilatih tidak menjadi *bias* pada data yang memiliki jumlah paling banyak. *Under sampling* data dilakukan dengan cara mengurangi jumlah data pada kelas cacar monyet dan kelas normal sehingga jumlahnya tidak jauh berbeda dengan kelas cacar air dan campak. Tabel 3 menunjukkan jumlah data sebelum diaugmentasi, setelah diaugmentasi, serta setelah augmentasi dan *under sampling*.

Hasil akhir dari tahap praproses data ini adalah dataset yang berjumlah 2260 data citra yang terdiri dari 4 kelas, dengan format data citra adalah PNG serta ukuran semua data citra adalah 128×128 dengan moda warna RGB. Data dibagi ke dalam 3 kategori data yaitu data latih, data validasi dan data *test* dengan rasio 80:10:10 dimana 80% untuk data latih, 10% untuk data validasi dan 10% untuk data *test*. Jumlah dari masing-masing data tersebut ditunjukkan pada Tabel 4.

Tabel 2 Parameter augmentasi data

<i>Type Generator</i>	Nilai	Keterangan
<i>Rotation_range</i>	45	Citra akan dirotasi secara random dengan rentang nilai antara 0° - 45°
<i>Width_shift_range</i>	0.2	Menggeser citra sepanjang sumbu X sebesar 20%.
<i>Height_shift_range</i>	0.2	Menggeser citra sepanjang sumbu Y sebesar 20%.
<i>Shear_range</i>	0.2	Memiringkan citra sepanjang sumbu X dan Y sebesar 20%.
<i>Zoom_range</i>	0.2	Memperbesar gambar secara random sebesar 20%.
<i>Horizontal_flip</i>	<i>True</i>	Membalik citra secara horizontal.
<i>Fill_mode</i>	<i>Reflect</i>	Membuat refleksi dari citra untuk mengisi nilai yang kosong.

Tabel 3 Perbandingan jumlah citra setelah augmentasi dan *under sampling*

Kelas	Jumlah Sebelum Augmentasi	Jumlah Setelah Augmentasi	Jumlah Setelah Augmentasi dan <i>Undersampling</i>
Cacar Monyet	43 citra	860 citra	600 citra
Cacar Air	26 citra	520 citra	520 citra
Campak	27 citra	540 citra	540 citra
Normal	54 citra	1080 citra	600 citra
Total	150 citra	3000 citra	2260 citra

Tabel 4 Jumlah data latih, data validasi, dan data *test*

Kelas	Data <i>train</i>	Data <i>valid</i>	Data <i>test</i>
Cacar Monyet	480	60	60
Cacar Air	416	52	52

Kelas	Data <i>train</i>	Data <i>valid</i>	Data <i>test</i>
Campak	432	54	54
Normal	480	60	60
Total	1808	226	226

Pembuatan Model Arsitektur MobileNetV2

Pembuatan model arsitektur MobileNetV2 untuk mendeteksi citra digital penyakit cacar monyet dilakukan dengan menggunakan teknik *transfer learning*. *Pre-trained* model (model yang sudah dilatih) dari *base model* MobileNetV2 akan dibekukan (*freeze*) dan *head network* akan diperbaharui menggunakan lapisan baru yang sesuai dengan kebutuhan klasifikasi data citra cacar monyet. *Base model* dibekukan agar nilai *weight* yang sudah ada tidak berubah pada saat model sedang dilatih, selain itu dengan membekukan *base model* artinya bagian tersebut tidak perlu dilakukan pelatihan ulang sehingga akan menghemat *learning costs*. Pada Gambar 5 menunjukkan potongan kode untuk mengunduh *pre-trained model* dari MobileNetV2 sebagai *base model* dengan menggunakan API Keras Tensorflow.

Selanjutnya, ditambahkan lapisan *head network* baru pada *base model* MobileNetV2. Nilai parameter *include top = False*, karena akan ditambahkan lapisan baru pada *head network* yang akan menggantikan *fully-connected layer* pada *base model* MobileNetV2. Hal ini bertujuan untuk menyesuaikan kebutuhan klasifikasi citra digital penyakit cacar monyet pada penelitian ini. Lapisan baru yang akan ditambahkan adalah lapisan *global average pooling* dan lapisan prediksi (*dense layer*). Lapisan *global average pooling* adalah pengganti *fully-connected layer* pada CNN yang akan menghasilkan satu *feature map* untuk setiap kategori tugas klasifikasi yang ada pada lapisan terakhir dengan cara mengambil nilai rata-rata dari setiap *feature map* lalu hasil vektornya akan diberikan ke lapisan *softmax* (Lin et al. 2013). Lapisan prediksi atau *dense layer* adalah lapisan dalam model yang menerima *output* dari setiap neuron pada lapisan sebelumnya, di mana neuron pada *dense layer* melakukan perkalian matriks-vektor (Verma 2021).

Lapisan terakhir *base model* memiliki ukuran *output* $4 \times 4 \times 1280$ yaitu pada lapisan *output*. Ukuran *output* pada lapisan ini belum sesuai dengan *head network* yang akan dibuat, sehingga untuk menyamakan ukuran *output* dibuat *global average pooling layer* yang selanjutnya dikoneksikan ke *dense layer*. Lapisan prediksi atau *dense layer* bertugas untuk mengklasifikasi citra. Dataset pada penelitian ini memiliki 4 kelas sehingga ukuran *output* yang diharapkan adalah 4 dan fungsi aktivasi yang digunakan adalah *Softmax*.

Dengan telah dibuatnya *prediction layer* atau *dense layer*, artinya *head network* sudah selesai ditambahkan, namun belum digabungkan dengan *base model*. Penggabungan *base model* dengan *head network* dilakukan dengan *input* untuk model adalah *base model* yang sebelumnya sudah dibekukan dan *output* model adalah *prediction layer*. Ketika *base model* sudah digabungkan dengan *custom head network*, model menjadi mempunyai *trainable params* sebanyak 5124 yang artinya terdapat 5124 parameter yang dapat dilatih dari model arsitektur MobileNetV2 yang sudah diterapkan *transfer learning*.

Secara keseluruhan, arsitektur MobileNetV2 yang telah dibuat terdiri dari lapisan-lapisan serta operator untuk tiap lapisannya yang dapat dilihat pada Tabel 5. Model arsitektur MobilNetV2 ini akan menerima *input* gambar dengan ukuran $128 \times 128 \times 3$. Kemudian, *input* gambar tersebut akan dikonvolusikan dengan lapisan konvolusi standar berukuran 3×3 dengan *2 stride*. Lapisan konvolusi ini akan menghasilkan *output channel* sebanyak 32 dengan ukuran citra berubah menjadi setengah dari *input* awal, yaitu 64×64 . *Output* dari lapisan konvolusi standar berukuran 3×3 selanjutnya akan diproses pada lapisan *linear bottleneck*.

Pelatihan Model

Pelatihan model ini dilakukan dengan dua jenis *optimizer* dan *learning rate* yang berbeda. Hal ini dilakukan untuk melihat perbandingan dari hasil akurasi masing-masing *optimizer* dan parameter *learning rate*. Pelatihan pertama menggunakan *Adaptive Moment Estimation* (Adam) *optimizer* dengan *batch size* 5 dan *learning rate* 10^{-4} . Angka *learning rate* dan *batch size* didapatkan dari penelitian yang dilakukan oleh Rochmawati *et al.* (2021) dimana pada penelitian tersebut, nilai *batch size* 5 dan *learning rate* 10^{-4} dengan Adam *optimizer* menghasilkan nilai *loss* terendah dan akurasi yang tinggi. *Categorical crossentropy* digunakan sebagai *loss function* karena *output* pada penelitian ini bersifat *multiclass* yaitu terdiri dari 4 label kelas. Pelatihan kedua menggunakan *Root Mean Squared Propagation* (RMSprop) *optimizer* dengan *batch size* 5 dan *learning rate* 45×10^{-3} . Pemilihan *Optimizer* dan angka *learning rate* didapatkan dari penelitian yang dilakukan oleh Sandler *et al.* (2018) yang merupakan pengembang dari arsitektur MobileNetV2. *Categorical crossentropy* juga digunakan sebagai *loss function*.

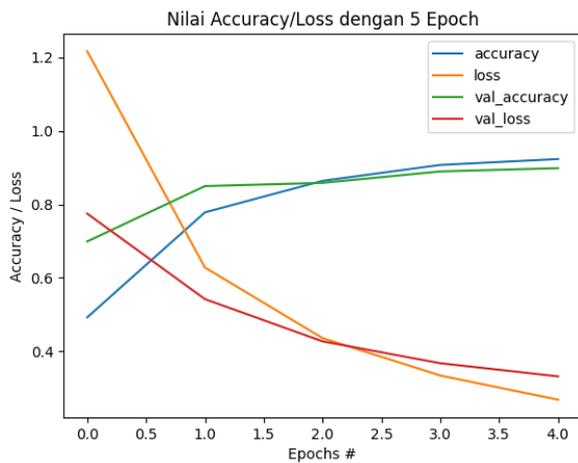
Pelatihan model pertama dilakukan satu kali dengan jumlah 5 *epoch* atau 5 kali iterasi, nilai *epoch* yang digunakan tidak besar karena untuk menghindari terjadinya *overfitting*. Gambar 5 menunjukkan grafik pelatihan model dimana pelatihan model yang pertama dengan jumlah *epoch* = 5 menunjukkan perubahan nilai *loss* dan *val_loss* semakin rendah pada tiap *epoch*-nya dan nilai *accuracy* dan *val_accuracy* semakin meningkat pada tiap *epoch*-nya. Nilai akurasi tertinggi didapatkan pada *epoch* ke 5 dengan nilai akurasi 0.92 dan nilai *loss* 0.27.

Pelatihan model kedua dilakukan satu kali dengan jumlah 5 *epoch* atau 5 kali iterasi. Gambar 6 menunjukkan grafik pelatihan model. Dari grafik tersebut menunjukkan *loss* dan *val_loss* yang tidak stabil namun cenderung menurun pada setiap *epoch*-nya. Nilai akurasi tertinggi didapatkan pada *epoch* ke 5 yaitu sebesar 0.97, namun nilai *loss* yang dihasilkan cukup tinggi yaitu 0.52 pada *epoch* ke 5.

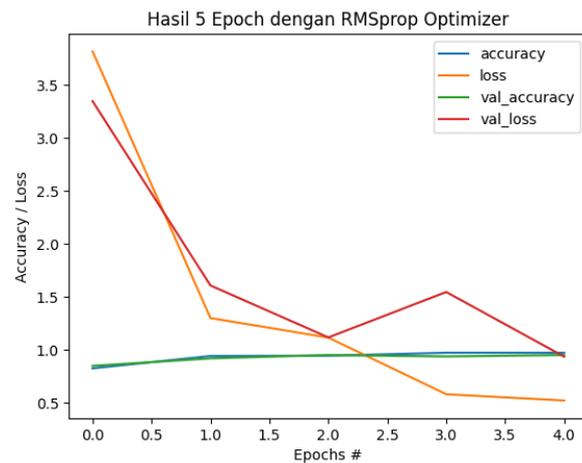
Hasil *accuracy*, *loss*, *validation accuracy* dan *validation loss* pada setiap *epoch* dengan masing-masing *optimizer* ditunjukkan pada Tabel 6. Pelatihan dengan kedua *optimizer* menghasilkan nilai *loss* dan *val_loss* yang sangat jauh berbeda. Adam *optimizer* memberikan hasil *loss* dan *val_loss* yang rendah, sedangkan RMSprop *optimizer* memberikan hasil *loss* dan *val_loss* yang cukup tinggi.

Tabel 5 Arsitektur MobileNetV2 pada penelitian ini.

Input	Operator	t	c	n	s
$128^2 \times 3$	Conv2d	-	32	1	2
$64^2 \times 32$	Bottleneck	1	16	1	1
$64^2 \times 16$	Bottleneck	6	24	2	2
$32^2 \times 24$	Bottleneck	6	32	3	2
$16^2 \times 32$	Bottleneck	6	64	4	2
$8^2 \times 64$	Bottleneck	6	96	3	1
$8^2 \times 96$	Bottleneck	6	160	3	2
$4^2 \times 160$	Bottleneck	6	320	1	1
$4^2 \times 320$	Conv2d 1x1	-	1280	1	1
$4^2 \times 1280$	Avgpool 4x4	-	-	1	-
$1 \times 1 \times 1280$	Conv2d 1x1	-	k	-	-
$1 \times 1 \times 4$	Softmax	-	-	-	-



Gambar 5 Grafik hasil pelatihan model dengan Adam optimizer



Gambar 6 Grafik hasil pelatihan model dengan RMSprop optimizer

Tabel 6 Hasil training data

No	Optimizer	Epoch ke-	Accuracy	Loss	Val_accuracy	Val_loss
1	<i>Adaptive Moment Estimation (Adam)</i>	1	0.4928	1.2159	0.6991	0.7747
		2	0.7782	0.6283	0.8496	0.5423
		3	0.8639	0.4362	0.8584	0.4275
		4	0.9071	0.3346	0.8894	0.3680
		5	0.9231	0.2690	0.8982	0.3323
2	<i>Root Mean Squared Propagation (RMSprop)</i>	1	0.8263	3.8149	0.8496	3.3459
		2	0.9425	1.3003	0.9204	1.6079
		3	0.9452	1.1161	0.9513	1.1177
		4	0.9729	0.5823	0.9381	1.5460
		5	0.9729	0.5228	0.9513	0.9367

Pengujian Model dan Evaluasi Model

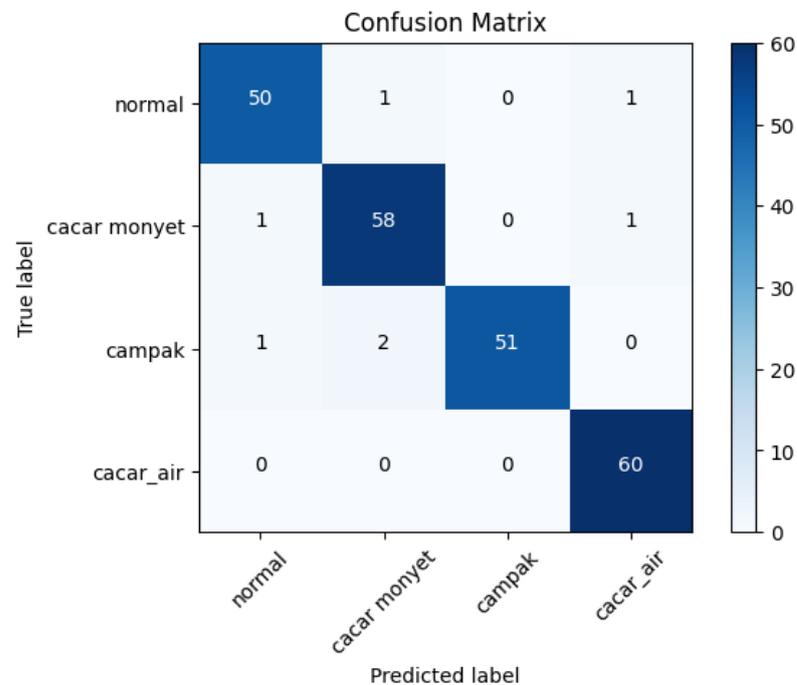
Model diuji dengan menggunakan dataset *data test*. Pengujian dilakukan setelah pelatihan disetiap *epoch* selesai. Pengujian model memanfaatkan fungsi dari API Keras Tensorflow yaitu `model.predict()`. Model yang telah diuji selanjutnya dievaluasi dengan menggunakan *confusion matrix*. Gambar 7 menunjukkan hasil evaluasi model yang dilatih menggunakan RMSprop optimizer.

Dilihat dari hasil *classification report* yang ditunjukkan pada Gambar 8, model yang telah dilatih memiliki akurasi sebesar 97%. Khususnya pada citra penyakit cacar monyet, model ini menghasilkan nilai *precision*, *recall* dan *f1-score* berturut-turut yaitu 95%, 97% dan 96%. Hasil *precision* menunjukkan bahwa model TL-MobileNetV2 mampu meminimalkan nilai *false positive* sampai 95% dengan RMSprop optimizer. Nilai *recall* sebesar 97% menunjukkan bahwa model mampu memprediksi sebagian besar nilai *true positive* dengan benar. Namun angka ini belum dapat dikatakan memiliki kualitas yang baik dalam mendeteksi citra penyakit cacar monyet karena pada hasil pelatihan sebelumnya, RMSprop optimizer menghasilkan nilai *loss function* yang cukup tinggi, yaitu mencapai 50%.

Gambar 9 menunjukkan hasil evaluasi model yang dilatih menggunakan Adam optimizer. Hasil evaluasi model pada Gambar 9 menunjukkan bahwa model berhasil mengklasifikasi dengan sempurna pada pelatihan ketiga dengan 15 *epoch*. Data uji yang berhasil terklasifikasi dengan benar untuk citra cacar monyet sebanyak 50 data citra dari 50 data uji. Hal yang sama juga untuk kelas citra cacar air dan normal. Namun untuk kelas campak terdapat 1 data citra yang diklasifikasikan sebagai citra cacar monyet, sehingga hanya 49 citra yang terklasifikasi dengan benar.

Secara ringkas, evaluasi model dapat dilihat dengan menggunakan fungsi *classification report* dari *library scikit-learn*. Gambar 10 menunjukkan hasil *classification report* dari model yang telah dilatih. Hasil *classification report* menunjukkan nilai dari *precision*, *recall*, *f1 score*, *accuracy*, *macro average*, dan *weighted average*. *Macro-averaged* adalah semua kelas memberikan kontribusi yang sama terhadap metrik rata-rata akhir. *Weighted-averaged* adalah kontribusi setiap kelas terhadap rata-rata diberi bobot berdasarkan ukurannya.

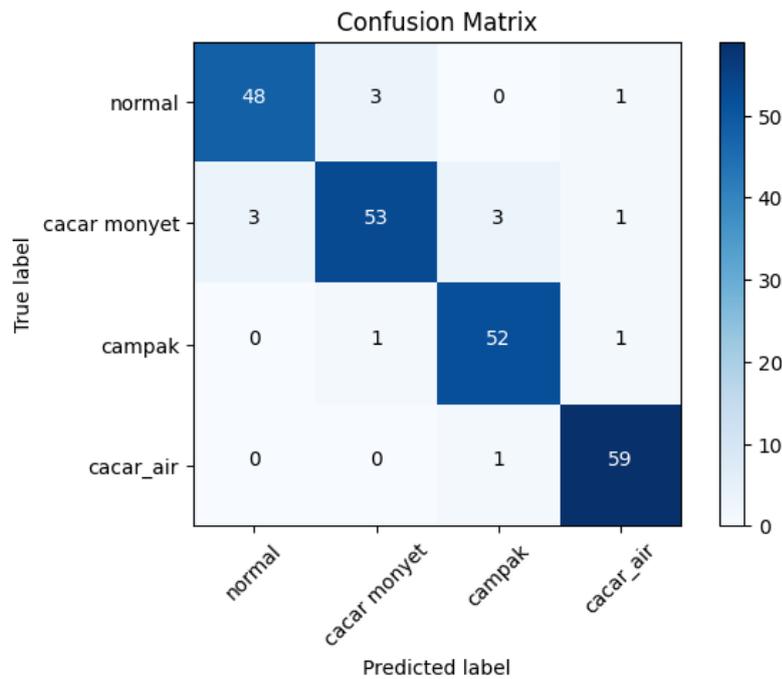
Dilihat dari hasil *classification report*, model yang telah dilatih memiliki akurasi sebesar 94%. Khususnya pada citra penyakit cacar monyet, model ini menghasilkan nilai *precision*, *recall*, dan *f1-score* berturut-turut yaitu 93%, 88%, dan 91%. Model ini menunjukkan nilai *precision* yang baik yaitu mencapai 94%. Hal ini menunjukkan kemampuan model TL-MobileNetV2 dengan menggunakan Adam *optimizer* dalam mengidentifikasi nilai positif dengan benar. Nilai *precision* yang tinggi ini sangat menguntungkan karena *precision* menunjukkan bagaimana model dapat meminimalkan nilai *false positive*. Meskipun nilai *precision* tinggi, model tetap mempertahankan *recall* yang kuat sebesar 92%. Nilai ini menunjukkan kemampuan model untuk menangkap proporsi substansial dari nilai positif aktual. *F1-score* sebesar 93% menegaskan efektivitas model dalam mencapai tujuan keseimbangan antara *precision* dan *recall*. Nilai akurasi keseluruhan mencapai 94% mengkonfirmasi akurasi model dalam semua prediksi.



Gambar 7 Hasil *confusion matrix* pada RMSprop *optimizer*

	precision	recall	f1-score	support
normal	0.96	0.96	0.96	52
cacar monyet	0.95	0.97	0.96	60
campak	1.00	0.94	0.97	54
cacar_air	0.97	1.00	0.98	60
accuracy			0.97	226
macro avg	0.97	0.97	0.97	226
weighted avg	0.97	0.97	0.97	226

Gambar 8 *Classification report* (RMSprop *Optimizer*)



Gambar 9 Confusion matrix (Adam Optimizer)

	precision	recall	f1-score	support
normal	0.94	0.92	0.93	52
cacar monyet	0.93	0.88	0.91	60
campak	0.93	0.96	0.95	54
cacar_air	0.95	0.98	0.97	60
accuracy			0.94	226
macro avg	0.94	0.94	0.94	226
weighted avg	0.94	0.94	0.94	226

Gambar 10 Classification report (Adam optimizer)

Analisis Hasil

Berdasarkan hasil pelatihan dataset dengan model yang telah dibuat dapat diambil kesimpulan bahwa model TL-MobileNetV2 yang dilatih dengan menggunakan Adam *optimizer* memiliki hasil akurasi yang lebih rendah dibandingkan dengan RMSprop *optimizer* dalam mengklasifikasi citra penyakit cacar monyet. Hasil akurasi pada kedua *optimizer* memiliki selisih 5%, namun pada nilai *loss* memiliki selisih yang jauh. Hasil *loss function* dari RMSprop *optimizer* lebih besar daripada hasil *loss function* pada penggunaan Adam *optimizer*, yaitu lebih dari 50%. Sedangkan, pada Adam *optimizer* nilai *loss* kurang dari 30%. *Loss function* mewakili ketidakakuratan prediksi dalam klasifikasi sehingga semakin kecil nilai *loss function* maka semakin baik modelnya (Zhang *et al.* 2018).

Berdasarkan hasil pengujian dan evaluasi performa model, dapat diambil kesimpulan bahwa model TL-MobileNetV2 memiliki performa yang baik dengan tingkat akurasi pada data uji mencapai 94%. Hasil akurasi ini dipengaruhi oleh keberagaman dataset, *hyperparameter*, dan *optimizer* yang digunakan.

SIMPULAN

Penelitian ini telah berhasil mengimplementasikan model arsitektur MobileNetV2 dalam mendeteksi citra penyakit cacar monyet sekaligus mengklasifikasi data citra penyakit kulit dengan *famili orthopoxvirus* (cacar monyet, cacar air, campak) yang diperoleh dari dataset

publik. Pembangunan model MobileNetV2 dengan menerapkan teknik *transfer learning* dan menggunakan *pre-trained* model dari MobileNetV2 pada citra digital penyakit cacar monyet dengan menggunakan algoritma CNN akan menghemat *learning cost* karena *base model* tidak akan melakukan pembaruan bobot. Kemudian pada *head network* akan diubah dan disesuaikan dengan kebutuhan klasifikasi. Hasil analisis menunjukkan bahwa model TL-MobileNetV2 yang dilatih dengan menggunakan Adam *optimizer* dengan *hyperparameter learning rate* 10^{-4} dan *epoch* 5 menghasilkan akurasi pada data uji sebesar 94%, dan akurasi pada data latih sebesar 92% dengan nilai *loss* 27%. Disisi lain penerapan RMSprop *optimizer* dengan *hyperparameter learning rate* 45×10^{-3} dan *epoch* 5 menghasilkan akurasi pada data uji sebesar 97%, dan akurasi pada data latih sebesar 97% dengan nilai *loss* mencapai 52%. Kedua *optimizer* ini menghasilkan akurasi yang tinggi, namun dilihat dari nilai *loss* yang dihasilkan, Adam *optimizer* lebih baik dibandingkan dengan RMSprop *optimizer*.

DAFTAR PUSTAKA

- Ahsan, M.M. *et al.* 2022. Image Data collection and implementation of deep learning-based model in detecting Monkeypox disease using modified VGG16. Available at: <http://arxiv.org/abs/2206.01862>.
- Ahsan, M.M., Uddin, M.R. and Luna, S.A. 2022. Monkeypox Image Data collection. Available at: <http://arxiv.org/abs/2206.01774>.
- Ali, S.N. *et al.* 2022. Monkeypox Skin Lesion Detection Using Deep Learning Models: A Feasibility Study. Available at: <https://doi.org/10.48550/arXiv.2207.03342>.
- CDC Government. 2022. *Monkeypox Signs and Symptoms*.
- CNBC Indonesia. 2022. Kasus Pertama Cacar Monyet di Jakarta! Tertular Dari Mana? Available at: <https://www.cnbcindonesia.com/news/20220822075423-4-365452/kasus-pertama-cacar-monyet-di-jakarta-tertular-dari-mana> (Accessed: 24 August 2022).
- Gulzar, Y. 2023. Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability* 15(3)-1906. Available at: <https://doi.org/10.3390/su15031906>.
- Huu, P.N. *et al.* 2022. Proposed Detection Face Model by MobileNetV2 Using Asian Data Set. *Journal of Electrical and Computer Engineering*. Available at: <https://doi.org/10.1155/2022/9984275>.
- Iman, M., Rasheed, K. and Arabnia, H.R. 2022. A Review of Deep Transfer Learning and Recent Advancements. Available at: <https://doi.org/10.3390/technologies11020040>.
- Lin, M., Chen, Q. and Yan, S. 2013. Network In Network. Available at: <http://arxiv.org/abs/1312.4400>.
- Rochmawati, N. *et al.* 2021. Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam.
- Sandler, M. *et al.* 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Available at: <http://arxiv.org/abs/1801.04381>.
- Unit Pelayanan Kesehatan Kementerian Kesehatan. 2022. Kemenkes Bersiap Menghadapi Cacar Monyet. Available at: <https://upk.kemkes.go.id/new/kemenkes-bersiap-menghadapi-cacar-monyet-> (Accessed: 24 August 2022).
- Verma, Y. 2021. A Complete Understanding of Dense Layers in Neural Networks. Available at: <https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/> (Accessed: 21 April 2023).
- Zhang, Y. *et al.* 2018. *Tire Defects Classification Using Convolution Architecture for Fast Feature Embedding*.