JTERA (Jurnal Teknologi Rekayasa), Vol. 7, No. 2, Desember 2022, Hal. 209-216
Terakreditasi "**Peringkat 3**" oleh Kemenristek/BRIN, Nomor SK: 85/M/KPT/2020
DOI: 10.31544/jtera.v7.i2.2022.209-216
p-ISSN 2548-737X
e-ISSN 2548-8678

Level Optimum Hyperparameter Tuning Epoch dalam Klasifikasi Citra Bencana Kebakaran

Rian Rahmanda Putra[1], Indra Griha Tofik Isa[2#], Ahmad Bahri Joni Malyan[3], Ema Laila[4], Agum Try Wardhana[5]
[1,3,4]Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya
[2]Jurusan Manajemen Informatika, Politeknik Negeri Sriwijaya
[5]Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya
Jalan Srijaya Negara Palembang, Indonesia
[#]indra_isa_mi@polsri.ac.id

Abstrak

Hyperparameter tuning menjadi bagian yang tidak terpisahkan dalam sebuah pemodelan untuk mencapai performa akurasi yang terbaik. Salah satu parameter yang digunakan dalam *hyperparameter tuning* adalah *epoch*, dimana merupakan bagian dari pelatihan dalam sebuah model untuk melihat akurasi terbaiknya. Dalam penelitian ini dilakukan *hyperparameter tuning* terhadap *epoch* untuk melihat titik optimum yang dihasilkan dengan 11 skenario *epoch* yakni *epoch* 50, *epoch* 55, *epoch* 60, *epoch* 65, *epoch* 70, *epoch* 75, *epoch* 80, *epoch* 85, *epoch* 90, *epoch* 95 dan *epoch* 100. Adapun batasan dalam penelitian ini adalah: (1) data yang dicermati merupakan citra kebakaran sebanyak 4650 citra yang terdiri dari kondisi kebakaran dan tidak kebakaran dan (2) pemodelan dengan *deep learning* CNN. Tahapan yang dilakukan meliputi *preparation data*, *image augmentation*, *pre-processing data*, pemodelan menggunakan CNN, *epoch hyperparameter tuning*, dan *data interpretation*. Di dalam penelitian ini, acuan hasil yang optimal berpatokan pada nilai rerata *accuracy* dan *val accuracy*. Hasil penelitian menunjukkan bahwa secara keseluruhan skenario *epoch* memiliki rerata *accuracy* dan *val accuracy* di atas 89%, dengan terdapat 6 skenario *epoch* dengan persentasi rerata *accuracy* dan *val accuracy* di atas 90%, yakni *epoch* 85, *epoch* 90, *epoch* 80, *epoch* 95, *epoch* 75, dan *epoch* 100. Sedangkan untuk skenario *epoch* dengan nilai rerata *accuracy* dan *val accuracy* tertinggi terdapat pada *epoch* 85 dimana disimpulkan bahwa *epoch* 85 merupakan titik optimal dalam penelitian ini.

Kata kunci: *epoch*, *hyperparameter tuning*, *image processing*, *deep learning*, CNN

Abstract

Hyperparameter tuning is an integral part of modeling to achieve the best accuracy performance. One of the parameters used in hyperparameter tuning is epoch, which is part of training in a model to see the best accuracy. In this study, hyperparameter tuning of epochs was carried out to see the optimum point generated with 11 epoch scenarios namely epoch 50, epoch 55, epoch 60, epoch 65, epoch 70, epoch 75, epoch 80, epoch 85, epoch 90, epoch 95 and epoch 100. The limitations of this study are: (1) the data examined are fire images of 4650 images consisting of fire and non-fire conditions and (2) modeling with CNN deep learning. The steps taken include data preparation, image augmentation, data pre-processing, modelling using CNN, epoch hyperparameter tuning, and data interpretation. In this study, the reference for optimal results is based on the average value of accuracy and val accuracy. The results showed that overall the epoch scenarios had an average accuracy and val accuracy above 89%, with 6 epoch scenarios with an average percentage accuracy and val accuracy above 90%, namely epoch 85, epoch 90, epoch 80, epoch 95, epoch 75, and epoch 100. As for the epoch scenario with the highest average value of accuracy and val accuracy, it is found at epoch 85 where it is concluded that epoch 85 is the optimal point in this study.

Keywords: *epoch, hyperparameter tuning, image processing, deep learning, CNN*

I. PENDAHULUAN

Salah satu faktor keberhasilan dalam pemodelan adalah *hyperparameter tuning* pada data yang dicermati. Khususnya dalam *image processing*, *hyperparameter tuning* diimplementasikan setelah tahapan *image augmentation* dan pemodelan. *Hyperparameter tuning* dalam *image processing*

Diterima: 16 Oktober 2022; Direvisi: 19 Desember 2022; Disetujui: 21 Desember 2022
JTERA, Vol. 7, No. 2, Desember 2022

© 2022 JTERA (Jurnal Teknologi Rekayasa), Politeknik Sukabumi

dikaitkan dengan bagaimana melakukan penyesuaian terhadap nilai "*epoch*", dimana penentuan berapa kali algoritma *deep learning* bekerja melewati seluruh dataset baik secara alur maju maupun alur mundur [1], dengan kata lain *epoch* bekerja ketika seluruh *dataset* sudah melalui proses latih pada jaringan syaraf sampai dikembalikan ke awal untuk sekali putaran [2]. Karena satu *epoch* terlalu besar untuk dimasukkan ke dalam komputer, maka dari itu perlu dipartisi menjadi bagian kecil, yang disebut dengan "batch".

Hyperparameter tuning dalam *image processing* tidak hanya dapat diimplementasikan pada nilai *epoch*, namun dapat dilakukan pada nilai "optimizer", "dropout", ataupun "dense" [3]. Penelitian yang dilakukan oleh Singh and Gupta [4] mengimplementasikan *hyperparameter tuning* pada parameter nilai "hidden layer" dan "neurons" yang dikenal dengan istilah metode *Particle Swarm Optimization* (PSO). Hasil PSO menunjukkan optimasi pencarian hingga 77% - 85% lebih cepat dibandingkan dengan tanpa *hyperparameter tuning*.

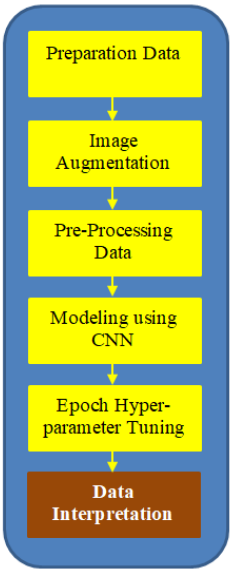
Epoch hyperparameter tuning telah banyak dilakukan oleh peneliti sebelumnya untuk menghasilkan akurasi pemodelan yang optimum. Alasan yang mendasarinya adalah bahwa *epoch* menunjukkan bagaimana kualitas algoritma yang telah dilatih untuk mengenali *dataset* dalam satu siklus. *Epoch* ditunjukkan oleh bilangan integer, misalnya *epoch* 30, artinya adalah *learning* algoritma yang bekerja dalam keseluruhan *dataset* adalah sebanyak 30 *cycles*. Badriyah [5] mengimplementasikan *hyperparameter tuning* untuk meningkatkan performa akurasi dengan pendekatan *random search* dan Bayesian *optimization* terhadap 7 parameter, yakni "epoch", "learning rate", "activation function", "batch size", "hidden node", "dropout rate", dan "hidden layer". Dari pengamatan yang dilakukan, terdapat 2 aspek yang menjadi penilaian dari kedua pendekatan tersebut. Dalam aspek *accuracy*, pendekatan *random search* memiliki nilai tertinggi dibandingkan dengan Bayesian *optimization*. Sementara dari segi optimasi waktu, Bayesian *optimization* memiliki nilai terbaik dibandingkan *random search*. *Hyperparameter tuning* pada pemodelan klasifikasi deteksi penyakit COVID-19 [3] dilakukan dengan menggunakan parameter "optimizer", "dense layer", "dropout setelah pooling layer", dan "dropout pada fully connecter layer". Hasil akurasi terbaik menunjukkan persentasi 84,2% dengan *hyperparameter tuning*, sementara tanpa *hyperparameter tuning* menunjukkan tingkat akurasi sebesar 74,1%

Di dalam penelitian ini, citra yang diamati adalah beragam citra kebakaran dengan kondisi kebakaran dengan adanya api, asap, maupun kondisi normal. Hal ini dilatarbelakangi dengan isu bencana kebakaran yang kerap terjadi, sehingga bagaimana melalui *machine learning* membangun pemodelan citra kebakaran yang akurat. Kumar [6], memprediksikan kebakaran hutan menggunakan tensorflow dengan pemodelan U-Net, menghasilkan nilai akurasi 92%. Sementara Boadi [7] menggunakan citra kebakaran untuk menentukan resiko kebakaran di Ghana. Di dalam penelitian ini pemodelan menggunakan *deep learning* CNN untuk mengklasifikasi citra kebakaran dan tidak kebakaran. Namun fokus utamanya adalah nilai akurasi pemodelan dengan melakukan *hyperparameter tuning* dalam *epoch* pada model yang dibentuk.

Hal yang menjadi keterbaruan dalam penelitian ini berfokus pada *epoch hyperparameter tuning* terhadap pemodelan yang diimplementasikan dalam citra kondisi kebakaran dan tidak kebakaran. *Hyperparameter tuning* pada *epoch* dilakukan untuk mencari nilai akurasi terbaik dari model yang sudah dilatih. Jumlah *dataset* citra yang diamati sebanyak 4650 citra yang terbagi ke dalam 2 label klasifikasi, yakni Fire_Condition dan NonFire_Condition. Batasan nilai *epoch* yang diimplementasikan dalam *hyperparameter tuning* pada range 50-100, dengan penambahan 5 *epoch* pada setiap *epoch* value, yakni *epoch* 50, *epoch* 55, *epoch* 60, *epoch* 65, *epoch* 70, *epoch* 75, *epoch* 80, *epoch* 85, *epoch* 90, *epoch* 95, dan *epoch* 100.

II. METODE PENELITIAN

Tahapan penelitian mengikuti metode penelitian berikut ini pada Gambar 1.



Gambar. 1 Tahapan penelitian

Pada penelitian ini, data citra yang diamati sejumlah 4650 dataset, yang terbagi ke dalam 2 kelas, yakni kelas Fire_Condition dengan jumlah data 2865 dan kelas NonFire_Condition dengan jumlah data 1785. Adapun persentasi data uji sejumlah 25% dari total data, yakni sebanyak 1162 data. *Tools* yang digunakan dalam penelitian ini adalah Google Colab dengan *dataset* yang diunduh secara publik dari Google Image dan Kaggle.

A. Preparation Data

Tahapan *preparation data* merupakan penyiapan data yang berasal dari objek riil dan *public repository* dari Google Image. Data latih merupakan sekumpulan dataset yang digunakan untuk melatih algoritma dan pemodelan yang digunakan [8], sedangkan data uji digunakan untuk menguji bagaimana pemodelan yang dihasilkan atau pemodelan yang sudah dilatih tersebut.

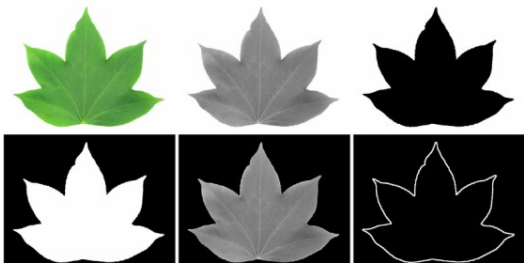
B. Image Augmentation

Image augmentation bertujuan untuk memproses *dataset* dimana *dataset* tersebut memiliki *imbalance* atau data yang tidak berimbang [9]. Penyeimbangan data dilakukan baik kepada data latih maupun data uji. Secara teknis, penyeimbangan data dilakukan dengan menghapus data yang berlebih, atau menyeimbangkan data dengan menambah data kepada data yang kurang, sehingga terjadi keseimbangan data.

C. Pre-Processing Data

Untuk memastikan data valid dan kredibel dilakukan *image processing* yang dilakukan pada tahapan *pre-processing data*. Hal ini dikarenakan dataset citra yang dicermati memiliki ragam kondisi, seperti pencahayaan dalam kondisi tertentu, kemiringan gambar, kondisi *zoom* gambar apakah dalam kondisi *zoom in* atau *zoom out*, dan *imbalance* warna antar gambar. Sehingga perlu dilakukan *pre-processing data* pada dataset citra yang terdiri dari: (1) *rotation range*; (2) *rescale*; (3) *zoom range*; (4) *shear range*; dan (5) *fill mode*.

Aspek citra yang akan diproses adalah *scaling* ukuran citra dengan menstandarisasi ukuran *pixel*, penambahan *blur* atau menghilangkan *noise* citra (yang disesuaikan dengan kondisi citra, *gray scaling* dimana dilakukan konversi warna citra menjadi *grayscale*, seperti ditunjukkan pada contoh Gambar 2.



Gambar 2. Contoh konversi citra ke dalam *grayscale*

Tabel 1. Skenario *epoch*

Skenario	Nilai *epoch*
Epoch 1	50
Epoch 2	55
Epoch 3	60
Epoch 4	65
Epoch 5	70
Epoch 6	75
Epoch 7	80
Epoch 8	85
Epoch 9	90
Epoch 10	95
Epoch 11	100

D. Pemodelan Menggunakan CNN

Pemodelan dilakukan menggunakan *Convolution Neural Network* (CNN) dimana terjadi proses konvolusi pada citra yang dicermati. CNN bertujuan untuk secara otomatis dan adaptif mempelajari struktur hierarki spasial elemen dengan menggunakan *backpropagation* dari beberapa blok penyusun [10]. CNN merupakan konstruksi matematis yang pada umumnya terdiri dari tiga jenis lapisan, yakni lapisan konvolusional, lapisan sambungan, dan lapisan yang sepenuhnya terhubung. Ada 2 tahapan utama di dalam pemodelan CNN, tahap pertama adalah klasifikasi citra menggunakan *feedforward*. Tahap kedua merupakan tahap pembelajaran dengan metode *backpropagation* [11]. Secara teknis pemodelan CNN mengimplementasikan beberapa elemen yakni konvolusi itu sendiri, proses aktivasi, dan *pooling*.

E. Epoch Hyperparameter Tuning

Setelah dilakukan pemodelan terhadap data latih, tahapan berikutnya adalah melakukan *hyperparameter tuning* terhadap nilai *epoch*. Pengaturan *epoch* value dilakukan dengan batasan range 50-100 dengan penambahan 5 nilai untuk setiap *epoch*, sehingga terdapat 11 skenario *epoch* seperti ditunjukkan pada Tabel 1.

F. Data Interpretation

Hasil *hyperparameter tuning* terhadap *epoch* memberikan variasi akurasi dalam pemodelan. Interpretasi data yang digunakan adalah titik optimal

antara hasil akurasi dengan mempertimbangkan banyaknya waktu yang digunakan dalam setiap skenario *epoch*. Sehingga dapat disimpulkan berapa nilai *epoch* yang optimal dalam menghasilkan titik akurasi, yang dikomparasi dengan efektifitas banyaknya waktu yang dilakukan selama proses *epoch*.

III. HASIL DAN PEMBAHASAN

A. Preparation Data

Dalam *preparation data* menggunakan *library* matplotlib.pyplot dan matplotlib.images untuk memvisualisasikan citra seperti Gambar 3.

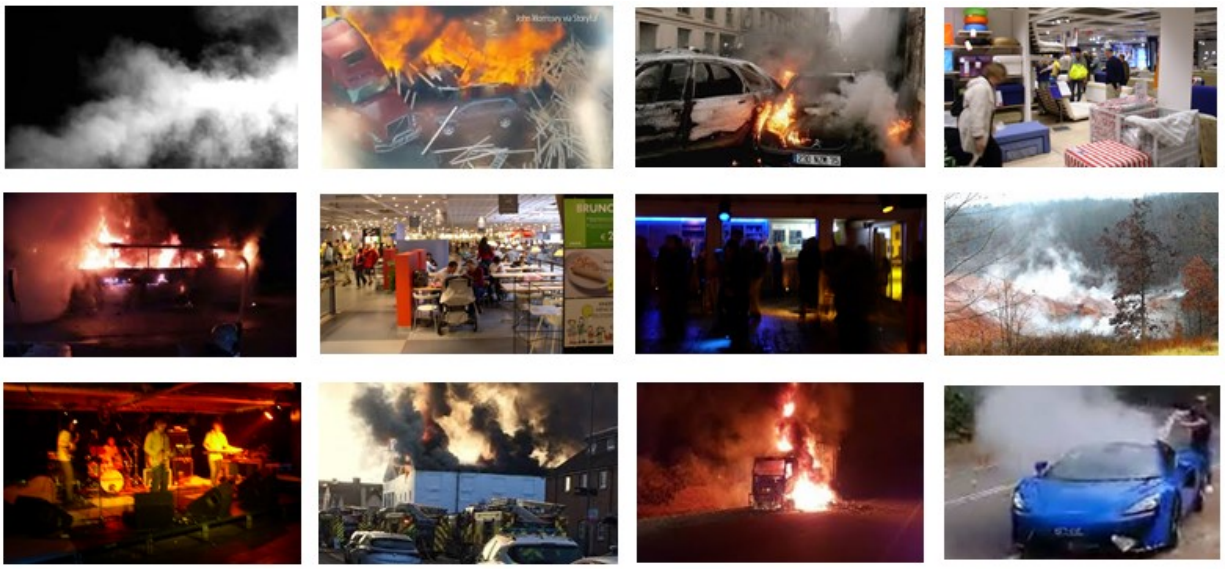
Sebagaimana yang sudah diuraikan sebelumnya, bahwa citra kebakaran selanjutnya dibagi ke dalam data latih dan data uji, dengan masing-masing di dalamnya terdapat kelas Fire_Condition and NonFire_Condition. Untuk data Fire_Condition terdiri dari gambar dengan citra api dan citra asap serta beragam kondisi kebakaran, seperti kebakaran rumah, kebakaran kendaraan, dan kebakaran hutan.

Berdasarkan hasil observasi citra yang diamati, Tabel 2 menunjukkan persentasi karakteristik masing-masing citra dari keseluruhan citra yang diamati.

Dari Tabel 2 dapat dilihat bahwa karakteristik citra yang tertinggi adalah citra kebakaran api di dalam ruangan dengan persentasi 20% dari total keseluruhan gambar. Sementara persentasi terendah adalah citra kebakaran asap kendaraan dengan persentasi sebanyak 8% dari total *dataset*.

B. Image Augmentation

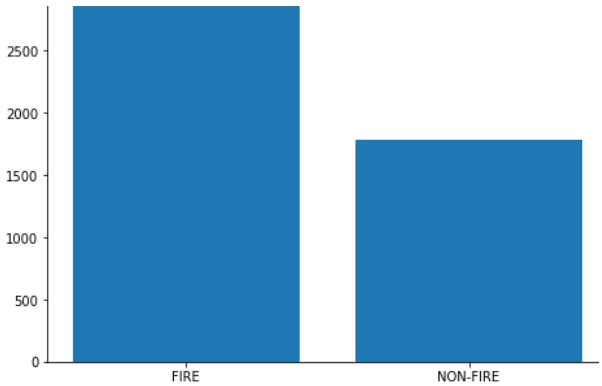
Image Augmentation dilakukan untuk menyeimbangkan data, antara kondisi dengan Fire_Condition dan NonFire_Condition. Gambar 4 menunjukkan citra sebelum *balancing* dan pada Gambar 5 menunjukkan citra setelah *balancing* data.



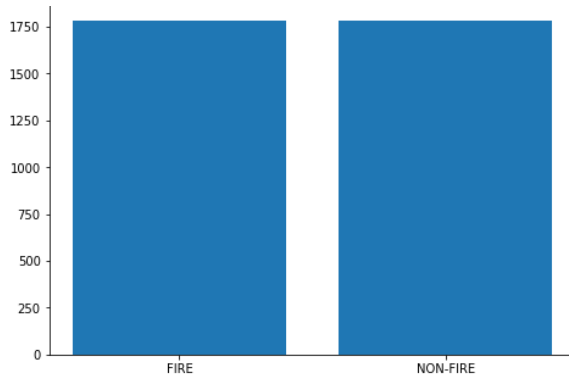
Gambar 3. Gambar citra yang diobservasi

Tabel 2. Karakteristik citra yang diamati

No	Karakteristik citra	Persentasi (%)
1	Kebakaran api ruangan	20
2	Kebakaran api hutan	10
3	Kebakaran api gedung	26
4	Kebakaran api kendaraan	12
5	Kebakaran asap gedung	11
6	Kebakaran asap ruangan	13
7	Kebakaran asap kendaraan	8
Total persentasi		100%



Gambar 4. Sebelum *data balancing*



Gambar 5. Setelah *data balancing*

Tabel 3. Pengaturan parameter dalam *pre-processing data*

No	Parameter	Nilai
1	*Zoom range*	0,2
2	*Shear range*	0,2
3	*Rotation range*	20
4	*Fill mode*	*Nearest*

Tabel 4. Detail *dense* dalam citra kebakaran

No	Dense	Jumlah layer	Jenis *activation*
1	Dense	32	relu
2	Dense_1	16	relu
3	Dense_2	32	relu
4	Dense_3	1	sigmoid

C. Pre-Processing Data

Dalam *pre-processing data* dilakukan beberapa hal, diantaranya *scaling pixel* yang dibuat terstandarisasi, penentuan nilai *rotation range*, *zoom range*, dan *shear range*. Dataset citra yang dicermati memiliki ragam karakteristik diantaranya 240 x 160 *pixel*, 184 x 274 *pixel*, 400 x 634 *pixel*, dan karakteristik *pixel* lainnya yang sangat beragam. Untuk menstandarisasi skala citra yang diamati maka dilakukan penyesuaian skala *pixel* menjadi 256 x 256. Misalnya jika ada citra dengan *pixel* 300 x 400, maka secara otomatis skala akan disesuikan menjadi 256 x 256.

Selain dengan pengaturan *scaling* sebagai bagian dari *pre-processing* data, juga dilakukan pengaturan untuk beberapa parameter. Tabel 3 menunjukkan pengaturan untuk beberapa parameter dalam *pre-processing* data.

Parameter yang diatur pada Tabel 3 terdiri dari *zoom range*, *shear range*, *rotation range*, dan *fill mode*. Untuk *zoom range* bernilai 0,2 yang artinya pemodelan akan memberikan toleransi terhadap citra yang memiliki ukuran *zoom* yang tidak berimbang dengan batas hingga 20%. Nilai *shear range* sebesar 0,2 mengindikasikan bahwa toleransi yang diberikan sebesar 20% terhadap kemencengan ragam citra. Sementara *rotation range* memberikan toleransi sebesar 20 derajat (ke kiri atau ke kanan).

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 254, 254, 32)      896

max_pooling2d (MaxPooling2D  (None, 127, 127, 32)      0
)

conv2d_1 (Conv2D)            (None, 125, 125, 16)      4624

max_pooling2d_1 (MaxPooling  (None, 62, 62, 16)        0
2D)

conv2d_2 (Conv2D)            (None, 60, 60, 32)        4640

max_pooling2d_2 (MaxPooling  (None, 30, 30, 32)        0
2D)

flatten (Flatten)            (None, 28800)             0

dense (Dense)                (None, 32)                921632

dense_1 (Dense)              (None, 16)                528

dense_2 (Dense)              (None, 32)                544

dense_3 (Dense)              (None, 1)                 33

=================================================================
Total params: 932,897
Trainable params: 932,897
Non-trainable params: 0
```

Gambar 6. *Trainable* parameter pemodelan CNN

D. Pemodelan Menggunakan CNN

Pemodelan yang dilakukan menggunakan CNN, dimana terjadi tiga tahap proses konvolusi dengan konvolusi pertama yakni Conv2D (32 (3, 3)) dimana hasil ekstraksi fitur map yang dari *input* berukuran 3 x 3 sebanyak 64. Sedangkan konvolusi kedua dan ketiga dengan Conv2D (16(3, 3)) dan Conv2D (32 (3, 3)) dimana masing-masing konvolusi memiliki ekstraksi fitur sebanyak 16 dan 32. Berikutnya adalah menambahkan jumlah *layer* dalam *hidden layer* melalui fungsi *dense*, dimana pada akhir *dense* terdapat 1 layer *output* yang berkaitan dengan *output* yang dihasilkan merupakan *binary classification*. Adapun detail *dense* dapat dilihat dalam Tabel 4.

Dalam mengkompilasi model, terdapat pengaturan parameter yakni *optimizer* = 'adam', *loss* = 'binary_crossentropy' dan *metrics* = ['accuracy']. Sehingga setelah men-*generate* pemodelan yang diimplementasikan, total *trainable* parameter dalam pemodelan CNN menghasilkan sebanyak 932.897 dengan detail keterangan dapat dilihat pada Gambar 6.

E. Epoch Hyperparameter Tuning

Setelah pemodelan CNN dilakukan tahap berikutnya adalah proses *epoch*. *Epoch* dilakukan dengan batasan seperti yang diuraikan dalam Tabel 5.

Tabel 5. Batasan dalam proses *epoch*

No	Uraian	Keterangan
1	*Bandwidth* internet	20 MBps
2	*Browser*	Google Chrome
3	Pemrosesan data	GPU Accelerator
4	Mode *epoch process*	Reset

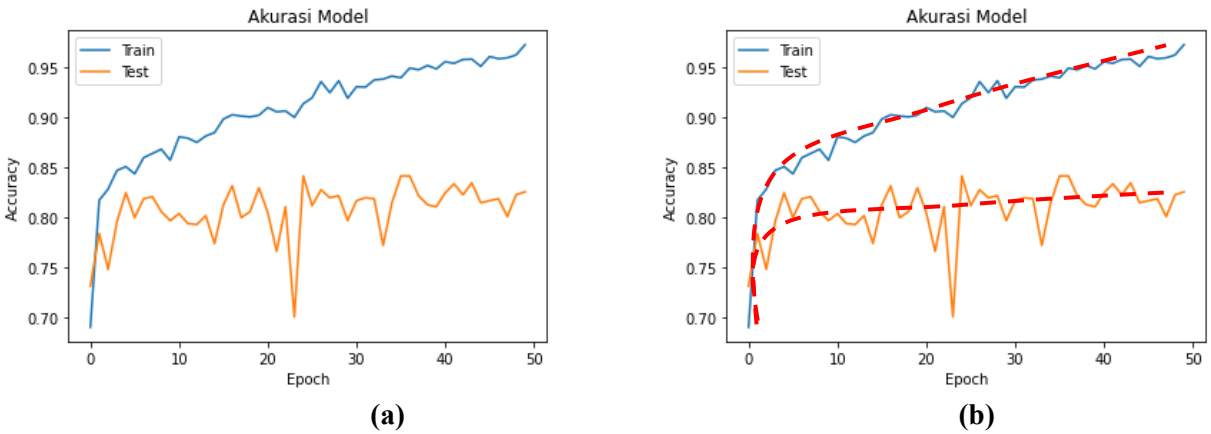
Dalam pemrosesan data, seperti yang diuraikan dalam Tabel 5 menggunakan fasilitas pendukung dari Google Colab, yakni GPU Accelerator. Hal ini dikarenakan untuk mengoptimalkan pemrosesan data, dimana data yang digunakan adalah citra visual. Sedangkan metode *epoch* menggunakan "reset" adalah setelah selesai *epoch* sesuai dengan skenario. Kemudian dilakukan proses reset dengan mengulang seluruh proses pemodelan dari awal akuisisi data hingga proses *epoch*.

Hasil akhir dalam proses *epoch* yang menjadi fokus dari penelitian ini adalah adalah tingkat akurasi model dalam mengenali citra dan validasi akurasi. Hasil proses *epoch* dalam eksekusi, misalnya dengan nilai *epoch* = 50, dapat dilihat pada Gambar 7, dimana pada *epoch* akan terhenti sesuai dengan nilai *epoch* yang dimasukan. Pada Gambar 7 di bagian paling bawah menunjukkan nilai *epoch* 50 dengan hasil *accuracy* 0,9725 dan *val accuracy* dengan nilai 0,8257. Hal ini dapat diartikan bahwa tingkat akurasi model dalam mengenali citra visual kebakaran adalah 0,9725 atau jika diubah ke dalam skala persentasi menjadi 97.25%. Sedangkan *val accuracy* menunjukkan nilai akurasi model mendeteksi citra kebakaran dalam validasi data.

Dengan menggunakan grafik *timeseries*, selama proses *epoch* dilakukan terjadi tren peningkatan tingkat akurasi, baik dari data latih dan data uji yang dikomparasi. Gambar 8 menunjukkan data tren kenaikan akurasi pada data latih dan data uji.

```
Epoch 39/50
73/73 [==============================] - 70s 949ms/step - loss: 0.1183 - accuracy: 0.9517 - val_loss: 0.8177 - val_accuracy: 0.8129
Epoch 40/50
73/73 [==============================] - 70s 950ms/step - loss: 0.1209 - accuracy: 0.9483 - val_loss: 0.6888 - val_accuracy: 0.8109
Epoch 41/50
73/73 [==============================] - 69s 937ms/step - loss: 0.1166 - accuracy: 0.9555 - val_loss: 0.7664 - val_accuracy: 0.8248
Epoch 42/50
73/73 [==============================] - 70s 950ms/step - loss: 0.1147 - accuracy: 0.9538 - val_loss: 0.7178 - val_accuracy: 0.8337
Epoch 43/50
73/73 [==============================] - 70s 958ms/step - loss: 0.1047 - accuracy: 0.9576 - val_loss: 0.8940 - val_accuracy: 0.8228
Epoch 44/50
73/73 [==============================] - 69s 941ms/step - loss: 0.0926 - accuracy: 0.9581 - val_loss: 0.6936 - val_accuracy: 0.8347
Epoch 45/50
73/73 [==============================] - 70s 947ms/step - loss: 0.1288 - accuracy: 0.9508 - val_loss: 0.7423 - val_accuracy: 0.8149
Epoch 46/50
73/73 [==============================] - 70s 954ms/step - loss: 0.1002 - accuracy: 0.9606 - val_loss: 0.8001 - val_accuracy: 0.8168
Epoch 47/50
73/73 [==============================] - 69s 939ms/step - loss: 0.1103 - accuracy: 0.9585 - val_loss: 0.7180 - val_accuracy: 0.8188
Epoch 48/50
73/73 [==============================] - 70s 949ms/step - loss: 0.1009 - accuracy: 0.9593 - val_loss: 0.9119 - val_accuracy: 0.8010
Epoch 49/50
73/73 [==============================] - 70s 956ms/step - loss: 0.0928 - accuracy: 0.9623 - val_loss: 0.7374 - val_accuracy: 0.8228
Epoch 50/50
73/73 [==============================] - 70s 951ms/step - loss: 0.0743 - accuracy: 0.9725 - val_loss: 0.8227 - val_accuracy: 0.8257
```

Gambar 7. *Epoch* 50 dalam pemodelan



(a) **(b)**

Gambar 8. (a) Akurasi model data latih dan data uji *epoch* 50; (b) Garis akurasi linear antara data latih dan data uji *epoch* 50

Tabel 6. Hasil akurasi skenario *epoch*

No	*Epoch*	*Accuracy*	*Val Accuracy*
1	*Epoch* 50	0,9725	0,8257
2	*Epoch* 55	0,9847	0,8149
3	*Epoch* 60	0,9733	0,8099
4	*Epoch* 65	0,9886	0,8040
5	*Epoch* 70	0,9750	0,8198
6	*Epoch* 75	0,9742	0,8317
7	*Epoch* 80	0,9903	0,8208
8	*Epoch* 85	0,9953	0,8495
9	*Epoch* 90	0,9805	0,8356
10	*Epoch* 95	0,9886	0,8198
11	*Epoch* 100	0,9852	0,8149

Jika dilihat dari Gambar 8 bagian (a), tren akurasi pada data latih ataupun data uji kecenderungannya adalah naik, walaupun jika dicermati lebih lanjut data uji terlihat tidak begitu signifikan. Namun pada Gambar 8 bagian (b) dengan garis bantu, tren dari data uji tetap naik. Sehingga dapat disimpulkan bahwa keduanya memiliki kecenderungan kemiripan dengan keduanya memiliki tren naik dalam akurasinya. Hal ini berlaku pada skenario *epoch* lainnya dimana akurasi model untuk data latih dan data uji memiliki kecenderungan meningkat untuk setiap siklus *epoch* pada skenario *epoch* yang diberikan.

F. Data Interpretation

Hasil keseluruhan skenario *epoch* yang terdiri dari *epoch* 50 hingga *epoch* 100 memiliki tingkatan akurasi di atas 0,97 dimana hasil akurasi terendah pada *epoch* 50 dengan akurasi 0,9725 atau 97,25% dan tertinggi pada *epoch* 85 dengan nilai akurasi 0,9953 atau 99,53%. Sebagai bagian dari *data interpretation*, hasil skenario *epoch* ditunjukkan pada Tabel 6.

Sebagaimana pada uraian sebelumnya mengenai akurasi tertinggi dan terendah, selanjutnya untuk hasil *val accuracy* terendah pada *epoch* 65 dengan nilai 0,8040 atau 80,40% sedangkan *val accuracy* tertinggi pada *epoch* 85 yakni 0,8495. Karena dalam penelitian ini terdapat 2 aspek yang menjadi acuan dalam melihat titik optimal dari *epoch* yang dilakukan dalam dataset citra kebakaran, maka perlu mempertimbangkan parameter *accuracy* dan *val accuracy*. Hal yang dilakukan adalah dengan mereratakan keduanya lalu diurutkan dari nilai tertinggi hingga terendah. Nilai tertinggi adalah yang diasumsikan merupakan titip optimal *hyperparameter tuning*. Tabel 7 menunjukkan perangkingan hasil rerata antara *accuracy* dan *val Accuracy*.

Tabel 7. Pengurutan nilai tertinggi hasil akurasi skenario *epoch*

No	*Epoch*	*Accuracy* (%)	*Val Accuracy* (%)	Rerata (%)
1	*Epoch* 85	99,53	84,95	92,24
2	*Epoch* 90	98,05	83,56	90,81
3	*Epoch* 80	99,03	82,08	90,56
4	*Epoch* 95	98,86	81,98	90,42
5	*Epoch* 75	97,42	83,17	90,30
6	*Epoch* 100	98,52	81,49	90,01
7	*Epoch* 55	98,47	81,49	89,98
8	*Epoch* 50	97,25	82,57	89,91
9	*Epoch* 70	97,50	81,98	89,74
10	*Epoch* 65	98,86	80,40	89,63
11	*Epoch* 60	97,33	80,99	89,16

Pada Tabel 7 nilai rerata tertinggi antara *accuracy* dan *val accuracy* terdapat pada skenario *epoch* 85 dengan persentasi 92,24% dan terendah terdapat dalam skenario *epoch* 60 dengan nilai persentasi 89,16%. Jika diamati lebih lanjut, terdapat 6 skenario *epoch* dengan nilai persentasi lebih dari 90%, antara lain *epoch* 85, *epoch* 90, *epoch* 80, *epoch* 95, *epoch* 75, dan *epoch* 100. Secara keseluruhan nilai rerata *accuracy* dan *val accuracy* sudah diatas nilai persentasi 89,16%. Dengan demikian, dapat disimpulkan *hyperparameter tuning epoch* yang optimal dengan melihat dari aspek *accuracy* dan *val accuracy* adalah pada *epoch* 85.

IV. KESIMPULAN

Hyperparameter tuning epoch untuk klasifikasi citra kebakaran telah berhasil dilakukan. Secara keseluruhan hasil skenario *epoch* memiliki rerata *accuracy* dan *val accuracy* di atas 89%, dengan terdapat 6 skenario *epoch* dengan persentasi rerata *accuracy* dan *val accuracy* di atas 90%, yakni *epoch* 85, *epoch* 90, *epoch* 80, *epoch* 95, *epoch* 75, dan *epoch* 100. Sementara untuk skenario *epoch* dengan nilai rerata *accuracy* dan *val accuracy* tertinggi terdapat pada *epoch* 85 dimana disimpulkan bahwa *epoch* tersebut merupakan titik optimal pada penelitian ini. Adapun saran dalam penelitian ini adalah dalam peningkatan akurasi dapat melakukan *hyperparameter tuning* terhadap aspek citra dengan melakukan *image augmentation* yang disesuaikan dengan karakteristik dari citra yang dicermati. Selain itu, untuk melihat efektifitas dari nilai *epoch* yang dihasilkan, perlu mempertimbangkan durasi *epoch* sebagai salah satu parameter yang digunakan dalam efektifitas *epoch* dalam satu siklus.

UCAPAN TERIMA KASIH

Ucapan terima kasih diberikan kepada Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Negeri Sriwijaya atas pendanaan kegiatan penelitian melalui skema hibah penelitian inovasi tahun 2022.

REFERENSI

[1] M. Wasil, Harianto, and Fathurrahman, "Pengaruh *Epoch* pada Akurasi menggunakan Convolutional Neural Network untuk Klasifikasi fashion dan Furniture," *Infotek J. Inform. dan Teknol.*, vol. 5, no. 1, pp. 53–61, 2022.

[2] S. Afaq and S. Rao, "Significance Of *Epoch*s On Training A Neural Network," *Int. J. Sci. Technol. Res.*, vol. 9, no. 6, pp. 485–488, 2020.

[3] A. E. Minarno, M. H. C. Mandiri, and M. R. Alfarizky, "Klasifikasi COVID-19 menggunakan Filter Gabor dan CNN dengan Hyperparameter Tuning," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 3, p. 493, 2021.

[4] G. A. P. Singh and P. K. Gupta, "Performance analysis of various machine learning-based approaches for detection and classification of lung cancer in humans," *Neural Comput. Appl.*, vol. 31, no. 10, pp. 6863–6877, 2019.

[5] T. Badriyah, D. B. Santoso, I. Syarif, and D. R. Syarif, "Improving stroke diagnosis accuracy using hyperparameter optimized deep learning," *Int. J. Adv. Intell. Informatics*, vol. 5, no. 3, pp. 256–272, 2019.

[6] S. M. Kumar, B. J. Sowmya, S. Priyanka, R. Sharma, S. Tej, and S. A. Karani, "Forest Fire Prediction Using Image Processing And Machine Learning," *Nat. Volatiles Essent. Oils*, vol. 8, no. 4, pp. 13116–13134, 2021.

[7] C. Boadi, S. K. Harvey, and A. Gyeke-dako, "Modelling of fire count data: fire disaster risk in Ghana," *Springer Plus J.*, vol. 4, no. 1, pp. 1–17, 2015.

[8] G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz, "An effective algorithm for hyperparameter optimization of neural networks," *IBM J. Res. Dev.*, vol. 61, no. 4, pp. 1–20, 2017.

[9] J. Wang and L. Perez, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," *ArXiv*, 2017.

[10] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020.

[11] S. R. Reynaldi, J. Apri, and S. A. Wahyu, "Klasifikasi Penyakit Kanker Kulit Menggunakan Metode Convolutional Neural Network," *Data Inst. Teknol. Telkom Purwokerto*, vol. 2, no. 1, pp. 52–57, 2022.