

Klasifikasi Gambar Batu-Kertas-Gunting Menggunakan *Convolutional Neural Network* dengan Fungsi *Callback* untuk Mencegah *Overfitting*

Samidin^{*1}, Akhmad Fadjeri²

^{1,2}Teknik Informatika, Fakultas Teknik, Universitas Ma'arif Nahdlatul Ulama Kebumen, Indonesia
Email: ¹samidinsampoerna@gmail.com, ²fadjeri.akhmadfadjeri@gmail.com

Abstrak

Dalam era modern, kecerdasan buatan (AI) dan *machine learning*, khususnya teknik *machine learning*, telah menjadi integral dalam perkembangan teknologi. Implementasi *Convolutional Neural Network* (CNN) dalam klasifikasi gambar sering mengalami masalah *overfitting*, di mana model terlalu beradaptasi dengan data latih dan kurang mampu menggeneralisasi data uji. Untuk mencegah *overfitting*, penelitian ini menggunakan *callback*, seperti *Early Stopping*, *Model Checkpointing*, dan *CSV Logging*. Dataset gambar batu-kertas-gunting diperoleh dari Github Dicoding dan diproses dengan pembagian subset, *augmentasi* data, dan *generator* data. Model CNN dirancang dengan lapisan konvolusi, lapisan *pooling*, dan lapisan *Dense*. *Optimizer*, fungsi *loss*, dan metrik evaluasi dipilih sesuai tugas klasifikasi gambar. Melalui pelatihan model, akurasi meningkat dari 48.75% menjadi 98.75%, sementara *loss* menurun dari 0.9794 menjadi 0.0448. Evaluasi pada data validasi juga menunjukkan peningkatan konsisten. Pelatihan dihentikan pada *epoch* ke-18 setelah mencapai akurasi di atas 98%, untuk mencegah *overfitting* dan mengoptimalkan waktu pelatihan. Kesimpulan menunjukkan bahwa penggunaan *callback* efektif dalam mencegah *overfitting* dan meningkatkan akurasi model CNN dalam klasifikasi gambar batu-kertas-gunting. Ini menjadi strategi penting dalam meningkatkan kualitas hasil akhir model.

Kata kunci: *Callback, Convolutional Neural Network, Kecerdasan Buatan, Klasifikasi Gambar, Machine Learning, Overfitting*

Abstract

In the modern era, artificial intelligence (AI) and machine learning techniques have become integral in technological advancements. The implementation of Convolutional Neural Networks (CNN) in image classification often faces the issue of overfitting, where the model adapts too closely to the training data and fails to generalize well to test data. To mitigate overfitting, this research employs callbacks, such as Early Stopping, Model Checkpointing, and CSV Logging. The rock-paper-scissors image dataset is obtained from Github Dicoding and processed through subset division, data augmentation, and data generators. The CNN model is designed with convolutional layers, pooling layers, and Dense layers. Optimizers, loss functions, and evaluation metrics are selected according to the image classification task. Through model training, accuracy increases from 48.75% to 98.75%, while loss decreases from 0.9794 to 0.0448. Evaluation on validation data also shows consistent improvement. Training is halted at the 18th epoch after achieving accuracy above 98%, to prevent overfitting and optimize training time. The conclusion indicates that the use of callbacks is effective in preventing overfitting and enhancing the accuracy of CNN models in rock-paper-scissors image classification. This becomes a crucial strategy in improving the quality of the final model outcomes.

Keywords: *Artificial Intelligence, Callback, Convolutional Neural Network, Image Classification, Machine Learning, Overfitting*

1. PENDAHULUAN

Pada era modern, kecerdasan buatan *Artificial Intelligence*, (AI) dan *machine learning* telah menjadi bagian integral dari perkembangan teknologi. Khususnya, teknik *machine learning* telah mendorong peningkatan luar biasa dalam pengolahan citra dan pengenalan pola (Fadjeri, 2020). AI dan *machine learning* telah membuka peluang baru dalam berbagai bidang, termasuk pengenalan wajah (Kurniawan & Kusriani, 2023), otomasi proses industri, dan klasifikasi gambar. Dalam konteks

klasifikasi gambar, *AI* dan *machine learning* telah membawa dampak yang signifikan. Misalnya, penggunaan teknik-teknik *machine learning* telah menjadi pilihan utama dalam memecahkan berbagai tugas klasifikasi gambar (Sarker, 2021). Teknik ini telah digunakan dalam aplikasi praktis, seperti kendaraan otonom yang menggunakan visi komputer untuk mengenali rambu lalu lintas dan objek di sekitarnya (Liyanthi, 2024). Dalam hal ini, klasifikasi gambar batu-kertas-gunting menjadi satu contoh yang menarik, yang memiliki potensi aplikasi dalam permainan, pengenalan gestur, dan bahkan pengendalian perangkat berbasis citra (Fadjeri, 2023).

Beberapa penelitian terdahulu juga telah mengaplikasikan *CNN* dalam klasifikasi gambar dengan konteks yang mirip, seperti klasifikasi citra permainan batu-kertas-gunting (Naufal et al., 2021). dan klasifikasi citra permainan batu-kertas-gunting menggunakan *Probabilistic Neural Network* (Siregar et al., 2021). Namun, permasalahan yang umum terjadi dalam implementasi *CNN* adalah *Overfitting* terjadi ketika model terlalu cocok dengan data pelatihan dan kehilangan kemampuan untuk secara tepat menggeneralisasi terhadap data pengujian. Pengembangan metode untuk mencegah *overfitting* menjadi hal yang sangat penting dalam meningkatkan performa model. Salah satu pendekatan yang dapat digunakan adalah penggunaan fungsi *callback* yang memonitor proses pelatihan dan memberhentikan pelatihan ketika ditemukan tanda-tanda *overfitting* (Ikhsal et al., 2023).

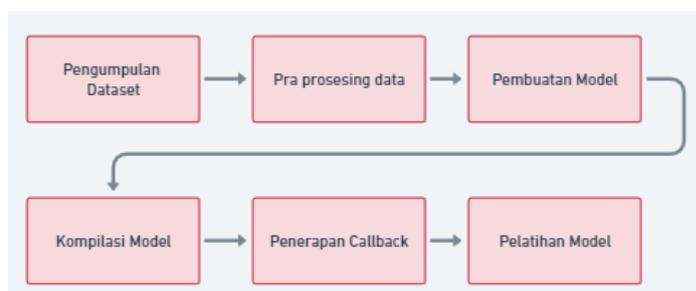
Sebagai contoh, Aristyanto dan (Aristyanto & Kurniawan, 2021) telah mengembangkan metode Neural Machine Translation berdasarkan hyperparameter neural network dengan mengambil pendekatan yang serupa dalam mengatasi kompleksitas model. (Fadillah et al., 2021) juga menggunakan teknik data *augmentasi* untuk mengatasi keterbatasan data pada model penerjemah bahasa isyarat Indonesia. Melihat penelitian-penelitian terdahulu dan tantangan yang dihadapi, penelitian ini bertujuan untuk mengembangkan metode klasifikasi gambar batu-kertas-gunting menggunakan *Convolutional Neural Network* (*CNN*) dengan penerapan fungsi *callback* untuk mencegah *overfitting*. Referensi dari penelitian terkait, seperti klasifikasi citra permainan batu-kertas-gunting (Fauzy et al., 2023) dan penggunaan *CNN* dalam konteks klasifikasi gambar (Siregar et al., 2021), akan menjadi landasan teoritis dalam pengembangan dan peningkatan metode yang diusulkan.

Penelitian ini juga akan merujuk pada penelitian-penelitian terkait lainnya yang berhasil mengatasi permasalahan sejenis, seperti penggunaan *Recurrent Neural Network* (Tarkus et al., 2020) dalam klasifikasi kualitas telur puyuh dan penggunaan *CNN* dalam mendeteksi api (WICAKSONO, 2020), untuk mendapatkan wawasan lebih lanjut mengenai pengaplikasian teknik kecerdasan buatan dalam berbagai konteks.

Dalam penelitian ini, gap research yang ditemukan adalah kekurangan metode klasifikasi gambar yang efektif dan handal dalam mengatasi permasalahan *overfitting*. Oleh karena itu, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan metode klasifikasi gambar batu-kertas-gunting yang handal dan efektif, serta membantu mendorong penggunaan teknologi kecerdasan buatan dalam berbagai aplikasi praktis.

2. METODE PENELITIAN

Metode penelitian yang dilakukan dalam penelitian ini terdiri dari empat langkah, yaitu pengumpulan Dataset, pra prosesi data, pembuatan model *CNN*, *compile* model, penerapan *callback* dan pelatihan model yang dapat di lihat pada Gambar 1.



Gambar 1. Metode Penelitian

2.1. Pengumpulan Dataset

Proses pengumpulan data melibatkan serangkaian langkah untuk menghimpun informasi yang diperlukan dalam mendukung penelitian (Yudhana et al., n.d.). Dataset yang digunakan dalam studi ini terdiri dari kumpulan gambar yang mencakup kategori *Rock*, *Paper*, dan *Scissor*, yang telah diperoleh dari repositori GitHub Dicoding. Kemudian, data tersebut diekstraksi.

2.2. Pra prosesing Data

Preprocessing merupakan langkah krusial dalam persiapan data untuk analisis atau model *machine learning*, yang melibatkan membersihkan, menormalkan, dan mempersiapkan data agar dapat diolah dengan lebih efisien oleh algoritma (Putra et al., 2024). Salah satu tahap dalam mempersiapkan data untuk pelatihan atau pengujian model adalah membagi dataset menjadi subset untuk pelatihan dan pengujian menggunakan paket *split folders*. Setelah itu, dilakukan *augmentasi* data dengan menggunakan objek *Image Data Generator*, di mana berbagai transformasi kecil seperti rotasi, pemutaran, dan pembalikan diterapkan pada gambar-gambar dalam dataset pelatihan. Tujuannya adalah untuk meningkatkan variasi data dan mencegah *overfitting* selama proses pelatihan model. Akhirnya, generator data dibuat menggunakan objek *Image Data Generator* yang telah dikonfigurasi sebelumnya. *Generator* ini akan memuat gambar-gambar dalam batch ke dalam model selama proses pelatihan atau pengujian.

2.3. Pembuatan Model

CNN merupakan jenis *neural network* yang dirancang khusus untuk mengolah data dua dimensi. Menurut (Nugroho et al., 2020), *CNN* termasuk dalam kategori *Deep Neural Network* karena memiliki struktur yang dalam dan sering digunakan untuk menganalisis data citra. Pembuatan arsitektur model *Convolutional Neural Network* (*CNN*) melibatkan penggabungan beberapa lapisan konvolusi dan *pooling*, yang kemudian diikuti oleh lapisan-lapisan *Dense* untuk tujuan klasifikasi. Desain arsitektur ini harus dipertimbangkan dengan cermat, mengingat kompleksitas dataset yang digunakan serta tujuan akhir dari proses klasifikasi.

2.4 Kompilasi Model

Setelah merancang arsitektur model, langkah selanjutnya adalah melakukan kompilasi model dengan mengatur parameter tertentu. Menurut (Omar et al., 2021), dalam konteks ini, model akan di-*compile* dengan menggunakan *optimizer Adam* dan *loss function* berupa *sparse categorical cross entropy*. Pemilihan *optimizer* yang tepat dan fungsi *loss* yang sesuai sangat penting dalam kasus klasifikasi gambar batu-kertas-gunting untuk mencapai hasil yang optimal.

2.5 Penerapan *Callback*

Dalam pelatihan model, penggunaan *callback* sangat penting untuk memantau dan mengontrol proses agar menghindari *overfitting* dan meningkatkan kualitas model secara keseluruhan, sebagaimana dijelaskan oleh (Arisandi & Satya, 2022). Salah satu *callback* yang berguna adalah *Early Stopping*, yang memungkinkan proses pelatihan dihentikan secara otomatis jika akurasi data telah mencapai nilai tertentu, sehingga menghindari *overfitting*. Selain itu, *Model Checkpointing* dan *CSV Logging* juga merupakan *callback* yang membantu dalam memantau dan mencatat kemajuan pelatihan model. Dengan menggunakan *callback* ini secara bijak, kita dapat mengoptimalkan proses pelatihan dan memperoleh model yang lebih baik.

2.6 Pelatihan Model

Dalam proses pelatihan model *CNN*, data *training* digunakan untuk mengajari sistem agar dapat mengenali pola yang diinginkan, seperti yang dijelaskan (Saputra et al., 2021). Tahapan ini melibatkan penggunaan dataset pelatihan yang telah dibagi dan diperluas. Prosesnya melibatkan iterasi berulang di mana parameter model disesuaikan agar sesuai dengan data pelatihan. Selama iterasi ini, model akan

menyesuaikan bobotnya untuk meminimalkan *loss* dan meningkatkan akurasi. Dengan demikian, model dilatih menggunakan dataset pelatihan yang telah disiapkan sebelumnya, dengan harapan dapat menghasilkan model yang mampu mengenali pola dengan lebih baik.

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan pada penelitian ini menggunakan Metode penelitian yang dilakukan dalam penelitian ini terdiri dari empat langkah, yaitu Dataset, pra prosesing data, pembuatan model *CNN*, *compile* model, penerapan *callback* dan pelatihan model.

3.1. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini terdiri dari sampel gambar Rock, Paper, dan *Scissor* dapat di lihat pada Gambar 2 yang diperoleh dari Github Dicoding. data tersebut diekstrak ke direktori /tmp. Proses ini melibatkan membuka *file zip* tersebut, mengekstrak seluruh isinya, dan kemudian menutup *file zip* setelah selesai. Setelah ekstraksi, dataset tersebut siap untuk diproses lebih lanjut



Gambar 2. Gunting Batu Kertas

Dataset ini terbagi menjadi 3 folder dengan total masing-masing 726 gambar batu, 712 gambar kertas, dan 750 gambar gunting yang dapat di lihat pada tabel 1.

Tabel 1. Sampel Dataset

Class	Jumlah
Batu	726
Kertas	712
Gunting	750

Untuk memperluas variasi data dan mencegah *overfitting*, teknik augmentasi juga diterapkan dengan menggunakan *Image Data Generator*. *Rescale* dilakukan dengan faktor 1./255, sedangkan augmentasi lainnya termasuk rotasi hingga 20 derajat, *flip horizontal*, sudut *shear* hingga 0.2, dan pengisian piksel kosong dengan piksel terdekat dapat di lihat tabel 2.

Tabel 2. Tipe Proses Data *Augmentation*

Data Augmentasi	Nilai
rescale	1./255
Rotation range	20 derajat
Horisontal flip	true
Shear range	0.2
Fill mode	Nearest

Generator data ini yang terdapat di tabel 3 digunakan untuk memuat gambar-gambar dalam dataset yang terorganisir dalam direktori-direktori pelatihan dan validasi. Gambar-gambar tersebut diubah ukurannya menjadi 150x150 piksel, dibagi menjadi *batch-batch* dengan ukuran 32 gambar untuk setiap iterasi pelatihan, dan label-labelnya dihasilkan dalam format *one hot encoding* karena terdapat beberapa

kelas yang harus diprediksi. Hal ini mempersiapkan data untuk pelatihan model pengenalan gambar tangan berdasarkan bentuk batu, gunting, atau kertas.

Tabel 3. Image Data Generator

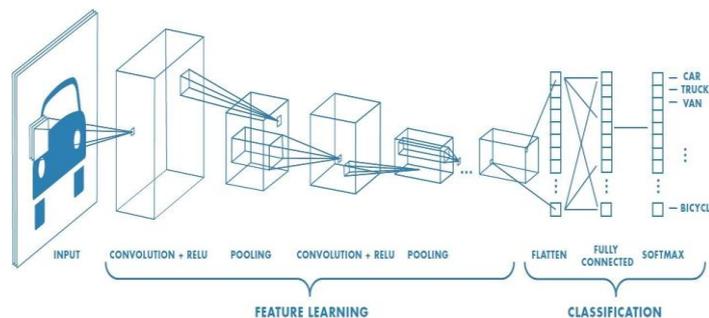
Generator	Directory	Target Size	Batch Size	Class Mode
Train generator	train_dir	(150, 150)	32	categorical
Validation generator	validation_dir	(150, 150)	32	categorical

3.2. Pembuatan Model

Pada model *CNN* terdapat empat lapisan konvolusi dengan jumlah filter berturut-turut 32, 64, 128, dan 256. Setiap filter memiliki ukuran 3x3 dan menggunakan fungsi aktivasi *ReLU*. Lapisan-lapisan konvolusi yang terdapat pada tabel 4, yang bertujuan untuk mengekstraksi fitur-fitur penting dari gambar. Setiap lapisan konvolusi menggunakan filter untuk mengaplikasikan operasi konvolusi pada gambar input. Lapisan-lapisan *MaxPooling* digunakan untuk mengurangi dimensi spasial gambar, sehingga mengurangi jumlah parameter dan komputasi yang diperlukan dalam model. Setelah setiap lapisan konvolusi, terdapat lapisan *MaxPooling* dengan ukuran pool 2x2. Lapisan *Flatten* bertugas untuk mengubah hasil dari lapisan konvolusi dan *pooling* menjadi *vektor* satu dimensi, yang kemudian akan menjadi input untuk lapisan-lapisan *Dense* di bagian selanjutnya dari model. Lapisan-lapisan *Dense* atau *fully connected layers* bertujuan untuk melakukan klasifikasi berdasarkan fitur-fitur yang diekstraksi oleh lapisan-lapisan konvolusi sebelumnya. Dalam model ini, terdapat dua lapisan *Dense* dengan 512 unit pada masing-masingnya, yang menggunakan fungsi aktivasi *ReLU*. Lapisan *output* terakhir menerapkan fungsi aktivasi *softmax* untuk mengklasifikasikan gambar ke dalam salah satu dari tiga kelas: batu, kertas, atau gunting.

Tabel 4. Struktur Model CNN

Layer (type)	Output Shape	Param#
conv2d (Conv2D)	(None, 148, 148, 32)	Filter size=3x3; Act=ReLU
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	Pool size=2x2
conv2d_1 (Conv2D)	(None, 72, 72, 64)	Filter size=3x3; Act=ReLU
conv2d_1 (Conv2D)	(None, 72, 72, 64)	Filter size=3x3; Act=ReLU
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	Pool size=2x2
conv2d_2 (Conv2D)	(None, 34, 34, 128)	Filter size=3x3; Act=ReLU
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	Pool size=2x2
conv2d_3 (Conv2D)	(None, 15, 15, 256)	Filter size=3x3; Act=ReLU
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 256)	Pool size=2x2
flatten (Flatten)	(None, 12544)	-
dense (Dense)	(None, 512)	Act=ReLU



Gambar 3. Arsitektur Model CNN

Dapat di lihat pada Gambar 3 Dalam model *Convolutional Neural Network (CNN)* fungsi aktivasi *ReLU (Rectified Linear Unit)* digunakan pada lapisan-lapisan konvolusi dan lapisan *Dense* sepenuhnya terhubung. Fungsi ini mengubah nilai *input* menjadi 0 jika nilainya *negatif*, sementara nilai *positif* tidak berubah. Di sisi lain, fungsi aktivasi *Softmax* digunakan pada lapisan *output* untuk menghasilkan probabilitas kelas dalam masalah klasifikasi multi-kelas. *Softmax* memperhitungkan nilai *eksponensial* dari setiap nilai input dan membaginya dengan jumlah *eksponensial* dari semua nilai *input*, menghasilkan distribusi *probabilitas* yang menggambarkan kepercayaan model terhadap setiap kelas

3.3. Compile Model

Metode *compile* digunakan untuk mengonfigurasi model untuk proses pelatihan. *Optimizer adam* Ini adalah algoritma optimasi yang digunakan untuk menyesuaikan bobot jaringan selama pelatihan. Algoritma Adam merupakan *optimizer* yang sering digunakan dalam pelatihan jaringan *neural* karena keefektifannya. *Loss categorical_crossentropy* Ini adalah fungsi kerugian yang digunakan selama pelatihan. Dalam kasus klasifikasi multi-kelas seperti yang dilakukan di sini, *categorical_crossentropy* adalah pilihan yang umum. *Metrics accuracy* Ini adalah *metrik* evaluasi yang akan dipantau selama pelatihan dan evaluasi model. Dalam hal ini, akan dipantau akurasi persentase kelas yang diprediksi dengan benar untuk mengevaluasi kinerja model.

3.4. Penerapan Callback

Pada penelitian ini, *callback* digunakan sebagai salah satu strategi yang efektif dalam mencegah *overfitting* dan meningkatkan akurasi model *Convolutional Neural Network (CNN)* dalam klasifikasi gambar batu-kertas-gunting. Dengan penerapan *callback*, model dapat dihentikan pada titik optimal untuk mencegah *overfitting*, yaitu kondisi di mana model terlalu mempelajari data pelatihan dan kehilangan kemampuan untuk generalisasi pada data baru. Gambaran visual dari penerapan *callback* yang dapat di lihat pada gambar 4 menunjukkan bahwa pelatihan model dihentikan setelah *accuracy* dan *validation accuracy* di tentukan yaitu mencapai tingkat akurasi lebih dari 0.98 dan *validation accuracy* lebih dari sama dengan 0.97, sehingga model tidak terlalu beradaptasi dengan data pelatihan. Selain itu, *Model Checkpointing* dan *CSV Logging* juga merupakan *callback* yang membantu dalam memantau dan mencatat kemajuan pelatihan model. Dengan demikian, *callback* berperan penting dalam meningkatkan kemampuan *generalisasi* model *CNN* yang dilatih, sehingga model dapat memberikan performa yang baik saat diimplementasikan dalam situasi dunia nyata.

```
#Dengan Menggunakan Callback & menghentikan training setelah akurasi terpenuhi
class callbackFunction(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('accuracy')>=0.98 and logs.get('val_accuracy')>=0.97):
            print("\nAccuracy above 98%, stop training!")
            self.model.stop_training=True

callbacks = callbackFunction()
csvlogger = CSVLogger('/content/training.log', separator=',', append=False)
```

Gambar 4. Function Callback

3.4. Pelatihan Model

Kemudian, digunakan metode *fit* untuk memulai proses pelatihan model. Berikut adalah penjelasan tentang argumen yang diatur *train generator* Ini adalah *generator* data untuk data pelatihan. *Steps per epoch = 25* ini adalah jumlah *batch* yang dieksekusi pada setiap *epoch* pelatihan. Dalam hal ini, telah diatur menjadi 25. Jumlah ini dipilih secara *arbitrer* dan dapat disesuaikan sesuai dengan kebutuhan. *Epochs = 20* ini adalah jumlah *epochs* iterasi lengkap melalui seluruh dataset yang dieksekusi selama pelatihan. *Validation data validation generator* ini adalah *generator* data untuk data validasi. *Validation steps = 5* ini adalah jumlah *batch* yang dieksekusi pada setiap *epoch* evaluasi validasi. Dalam hal ini, telah diatur menjadi 5. Seperti *steps per epoch*, jumlah ini dapat disesuaikan sesuai dengan kebutuhan. *Verbose 1* Ini adalah level detail dari *output* yang ditampilkan selama pelatihan. *CSV logger* Merupakan

objek *callback* yang digunakan untuk mencatat metrik pelatihan ke *file CSV* yang dapat di lihat di table 5. Pilihan ini mengatur agar *output* ditampilkan untuk setiap *epoch*. Setelah menjalankan kode ini, model akan melalui proses pelatihan dan evaluasi, dan histori pelatihan akan disimpan dalam *variabel history*. Histori ini dapat digunakan untuk memvisualisasikan kinerja pelatihan model.

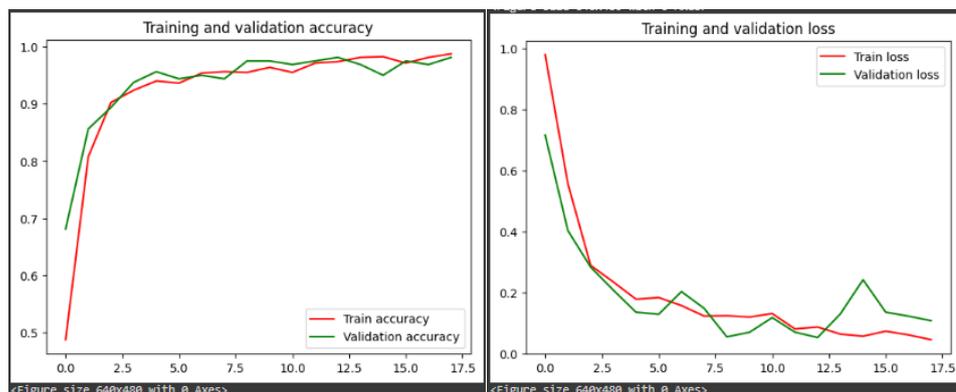
Tabel 5. Training Model

Parameter	Nilai
Epoch	20
Batch Size	25
Validation Step	5
Optimizer	Adam
Optimizer Metric	Accuracy
Loss Function	Categorical Cross Entropy
Verbose	1
Callbacks	CSV logger

Pelatihan model *Convolutional Neural Network (CNN)* untuk klasifikasi gambar batu-kertas-gunting, terlihat bahwa model mengalami peningkatan performa seiring dengan berjalannya *epoch*. Pada awal pelatihan yang dapat dilihat pada table 6, akurasi model hanya sekitar 48.75%, tetapi meningkat secara signifikan menjadi 98.75% pada akhir pelatihan. Demikian pula, *loss* model menurun dari 0.9794 menjadi 0.0448, menunjukkan bahwa model semakin mempelajari pola yang ada dalam data. Di sisi lain, evaluasi pada data validasi juga menunjukkan peningkatan yang, dengan akurasi meningkat dari 68.12% menjadi 98.12%. Hasil ini menunjukkan bahwa model telah belajar untuk secara efektif membedakan antara gambar batu, kertas, dan gunting, dan mampu melakukan klasifikasi dengan sangat baik pada data yang tidak dilihat sebelumnya. Penting untuk dicatat bahwa pelatihan model dihentikan lebih awal pada *epoch* ke-18 karena akurasi telah mencapai di atas 98%, sehingga menghindari *overfitting* dan mengoptimalkan waktu pelatihan. *Overall*, hasil pelatihan ini konsisten yang dapat di lihat pada Gambar 5 menunjukkan bahwa model *CNN* berhasil mengatasi tugas klasifikasi dengan baik dan siap untuk digunakan dalam aplikasi praktis.

Tabel 6. Hasil Pelatihan Model

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy	Duration/Epoch	Duration/Step
1	0.9794	0.4875	0.7155	0.6812	65s	3s
2	0.5562	0.8075	0.4026	0.8562	57s	2s
3	0.2888	0.9025	0.2831	0.8938	58s	2s
4	0.2343	0.9237	0.2076	0.9375	56s	2s
5	0.1776	0.9400	0.1351	0.9563	58s	2s
6	0.1831	0.9362	0.1283	0.9438	56s	2s
7	0.1569	0.9538	0.2026	0.9500	56s	2s
8	0.1224	0.9563	0.1477	0.9438	61s	2s
9	0.1235	0.9550	0.0543	0.9750	56s	2s
10	0.1191	0.9638	0.0693	0.9750	62s	3s
11	0.1306	0.9550	0.1173	0.9688	55s	2s
12	0.0803	0.9712	0.0696	0.9750	57s	2s
13	0.0865	0.9737	0.0522	0.9812	57s	2s
14	0.0632	0.9812	0.1293	0.9688	56s	2s
15	0.0564	0.9825	0.2415	0.9500	55s	2s
16	0.0728	0.9712	0.1351	0.9750	56s	2s
17	0.0608	0.9812	0.1221	0.9688	61s	2s
18	0.0448	0.9875	0.1072	0.9812	54s	2s



Gambar 5. Grafik Akurasi dan Model Los Validasi

4. KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa penggunaan *callback* dalam proses pelatihan model *Convolutional Neural Network* (CNN) efektif dalam mencegah *overfitting* dan meningkatkan akurasi model. Dengan mengimplementasikan *callback*, model dapat dihentikan saat mencapai tingkat akurasi yang tinggi yaitu 98%, menghindari *overfitting*, dan meningkatkan kemampuan generalisasi model pada data baru. Hal ini membantu dalam meningkatkan performa model CNN dalam klasifikasi gambar batu-kertas-gunting. *Callback* menjadi strategi yang penting dalam mengoptimalkan proses pelatihan model dan meningkatkan kualitas hasil akhir model.

DAFTAR PUSTAKA

- Arisandi, L., & Satya, B. (2022). Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network. *Jurnal Sistem Cerdas*, 5(3), 135–146. <https://doi.org/10.37396/jsc.v5i3.262>
- Aristyanto, M. Y., & Kurniawan, R. (2021). Pengembangan Metode Neural Machine Translation Berdasarkan Hyperparameter Neural Network. *Seminar Nasional Official Statistics, 2021*(1), 935–946. <https://doi.org/10.34123/semnasoffstat.v2021i1.789>
- Fadillah, R. Z., Irawan, A., Susanty, M., & Artikel, I. (2021). Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO). *Jurnal Informatika*, 8(2), 208–214. <https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/10768>
- Fadjeri, A. (2020). Pengolahan Citra Digital Untuk Menghitung Ekstraksi Ciri Greenbean Kopi Robusta Dan Arabika (Studi Kasus: Kopi Temanggung). *Indonesian Journal of Applied Informatics*, 4(2), 92. <https://doi.org/10.20961/ijai.v4i2.39253>
- Fadjeri, A. (2023). Klasifikasi Biji Kopi Berdasarkan Bentuk Menggunakan Image Processing dan K-NN. *Jurnal Ilmiah SINUS*, 21(2), 55. <https://doi.org/10.30646/sinus.v21i2.726>
- Fauzy, S. R., . A., & Azzahra, F. F. (2023). Implementasi Game Development Life Cycle Dalam Pembuatan Game Buana Ruh. *Indexia*, 5(01), 19. <https://doi.org/10.30587/indexia.v5i01.5215>
- Ikhsal, M. F., Dermawan, B. A., & Adam, R. I. (2023). Peningkatan Deteksi Kecelakaan di Jalan Raya Menggunakan Real-ESRGAN pada Citra CCTV Persimpangan Jalan. *Journal of Applied Informatics and Computing*, 7(1), 51–56. <https://doi.org/10.30871/jaic.v7i1.5562>
- Kurniawan, H., & Kusriani, K. (2023). Klasifikasi Pengenalan Wajah Siswa Pada Sistem Kehadiran dengan Menggunakan Metode Convolutional Neural Network. *Jurnal Media Informatika Budidarma*, 7(2), 846–856. <https://doi.org/10.30865/mib.v7i2.5958>
- Liyanthi, M. (2024). Perancangan Data Pipeline Computer Vision untuk Mendukung Pelatihan Agen Otonom dalam Mengenali Rambu Lalu Lintas. *Journal of Computer and Information Systems ...*, 5(1), 14–25. <https://www.journal-computing.org/index.php/journal->

- cisa/article/view/450%0Ahttps://www.journal-computing.org/index.php/journal-cisa/article/download/450/208
- Naufal, M. F., Huda, S., Budilaksono, A., Yustisia, W. A., Arius, A. A., Miranti, F. A., & Prayoga, F. A. T. (2021). Klasifikasi Citra Game Batu Kertas Gunting Menggunakan Convolutional Neural Network. *Techno.Com*, 20(1), 166–174. <https://doi.org/10.33633/tc.v20i1.4273>
- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia. *Algor*, 2(1), 12–21.
- Omar, J., Husna Shabrina, N., Bhakti, A. N., & Patria, A. (2021). Emotion Recognition using Convolutional Neural Network on Virtual Meeting Image. *Ultima Computing: Jurnal Sistem Komputer*, 13(1).
- Putra, F., Tahiyat, H. F., Ihsan, R. M., Rahmadden, R., & Efrizoni, L. (2024). Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 273–281. <https://doi.org/10.57152/malcom.v4i1.1085>
- Saputra, I., AJI PAMBUDI, R. S., DARONO, H. E., AMSURY, F., FAHDIA, M. R., RAMADHAN, B., & ARDIANSYAH, A. (2021). Analisis Sentimen Pengguna Marketplace Bukalapak dan Tokopedia di Twitter Menggunakan Machine Learning. *Faktor Exacta*, 13(4), 200. <https://doi.org/10.30998/faktorexacta.v13i4.7074>
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 1–21. <https://doi.org/10.1007/s42979-021-00592-x>
- Siregar, S. J., Lubis, A. I., & Ginting, E. F. (2021). Penerapan Neural Network Dalam Klasifikasi Citra Permainan Batu Kertas Gunting dengan Probabilistic Neural Network. *Building of Informatics, Technology and Science (BITS)*, 3(3), 420–425. <https://doi.org/10.47065/bits.v3i3.1143>
- Tarkus, A. D., Sompie, S. R. U. A., & Jacobus, A. (2020). Implementasi Metode Recurrent Neural Network pada Pengklasifikasian Kualitas Telur Puyuh. *Jurnal Teknik Informatika*, 15(2), 137–144.
- WICAKSONO, M. I. (2020). Pendeteksian Api Berbasis Pengolahan Citra Digital Menggunakan Metode Convolutional Neural Network. *Etd.Repository.Ugm.Ac.Id*. https://etd.repository.ugm.ac.id/home/detail_pencarian_downloadfiles/492280
- Yudhana, A., Umar, R., & Ahmadi, A. (n.d.). *Akuisisi Data Forensik Google Drive Pada Android Dengan Metode National Institute of Justice (NIJ)*. X(X), 8–13.

Halaman Ini Dikosongkan