

KLASIFIKASI TINTA CATRIDGE BERDASARKAN WARNA PADA GUDANG BARANG MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK

Aji Nurrohman^{1,*}, Berliyanto², Leni Asrar³, Sigit Wibisono⁴, Wibisono⁵, Suryadarma⁶, Triyono Budi Santoso⁷

^{1,7}Department of Electrical Engineering, Department of Informatics Engineering, Institut Teknologi Budi Utomo, Indonesia
ajinurrohman7@gmail.com

Article Info

Article history:

Kata Kunci:

Cartridge
Convolutional neural network
Python

ABSTRAK

Penggunaan printer dan tinta cartridge untuk mencetak data dalam bentuk teks ataupun gambar masih sering digunakan. Printer yang sering digunakan akan membuat tinta cartridge cepat habis, biasanya pengguna printer akan memesan kepada toko melalui aplikasi e-commerce. Toko akan mengirimkan melalui gudang barang, namun seringkali terjadi kesalahan barang antara yang dipesan dan diterima user. Hal ini dapat mempengaruhi kepuasan user, oleh karena itu penting bagi toko dalam melakukan cek manual terhadap barang yang akan dikirim. Untuk mengatasi masalah tersebut, dibuatlah sebuah model machine learning menggunakan Python dan Convolutional neural network yang dapat mengklasifikasikan citra hasil tangkapan kamera ponsel fokus pada penelitian ini mengklasifikasikan citra tinta cartridge HP 680 black color dan tri color. Dimulai dengan mengumpulkan dataset yang sudah ada berdasarkan pengalaman penulis yang pernah bekerja sebagai checker pada gudang barang. Kemudian langkah preprocessing agar citra siap untuk diinputkan kedalam model, lalu membuat arsitektur model dan melakukan pelatihan. Di akhir penelitian akan dilakukan evaluasi untuk melihat akurasi dari prediksi pada citra data test. peneliti menargetkan nilai akurasi yang ingin dicapai adalah 90%. Dapat disimpulkan berdasarkan akurasi tersebut bahwa akurasi model yang didapat pada penelitian ini sudah sesuai target yang ingin dicapai.



Corresponding Author:

Aji Nurrohman,
Institut Teknologi Budi Utomo,
Email: * ajinurrohman7@gmail.com

1. PENDAHULUAN

Penggunaan printer dan tinta cartridge untuk mencetak dokumen ditingkat pendidikan ataupun perkantoran masih banyak dilakukan. Printer yang sering digunakan akan membuat tinta cartridge cepat habis, membuat tinta cartridge menjadi bagian paling penting dalam hal mencetak dokumen dalam bentuk fisik. Hal ini membuat demand pembeli terhadap tinta cartridge masih tinggi, menyebabkan bisnis jual beli tinta cartridge masih banyak dilakukan (I. Cholissodin and A. A. Soebroto:2021)

Dalam bisnis jual beli tinta cartridge, tinta biasanya disimpan pada gudang barang jadi sebelum dijual kepada pembeli. Barang masuk ke gudang akan dilakukan pengecekan jenis tinta secara manual oleh tim penerima barang untuk diletakkan sesuai posisi agar mempermudah proses pengelolaan inventory, tujuannya agar pada saat barang keluar tim pic mudah dalam mengambil barang dan tidak terjadi kesalahan dalam menyiapkan barang keluar. Pengecekan barang secara manual akan memakan banyak waktu jika barang masuk berjumlah besar, hal ini juga akan mempengaruhi tingkat akurasi dalam pengecekan barang. Ketepatan serta konsistensi dalam proses pengecekan barang menjadi unsur penting dalam pengelolaan inventory (F. Loekman :2023)

Permasalahan akurasi, efisien dan konsistensi dalam pengecekan barang masuk jika dilakukan secara manual bisa diselesaikan dengan Deep Learning. Deep Learning adalah salah satu cabang dari machine learning yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. Deep learning sangat baik untuk diterapkan dalam berbagai aplikasi seperti pengenalan citra, suara, klasifikasi teks, dan sebagainya (Cholissodin and A. A. Soebroto: 2021). Pada bidang Deep Learning terdapat metode pengenalan citra digital yaitu Convolutional Neural Network (CNN), metode CNN ini telah digunakan untuk berbagai pengenalan citra digital (Awaludin & Rahwanto, 2017). Convolutional Neural Network adalah salah satu kelas yang ada didalam deep learning yang berfungsi untuk melakukan tugas pengenalan dan klasifikasi citra. Diantaranya, untuk pengenalan terhadap data medis, pengenalan wajah dan pengenalan barang.

2. METODE

2.1 Jenis Penelitian

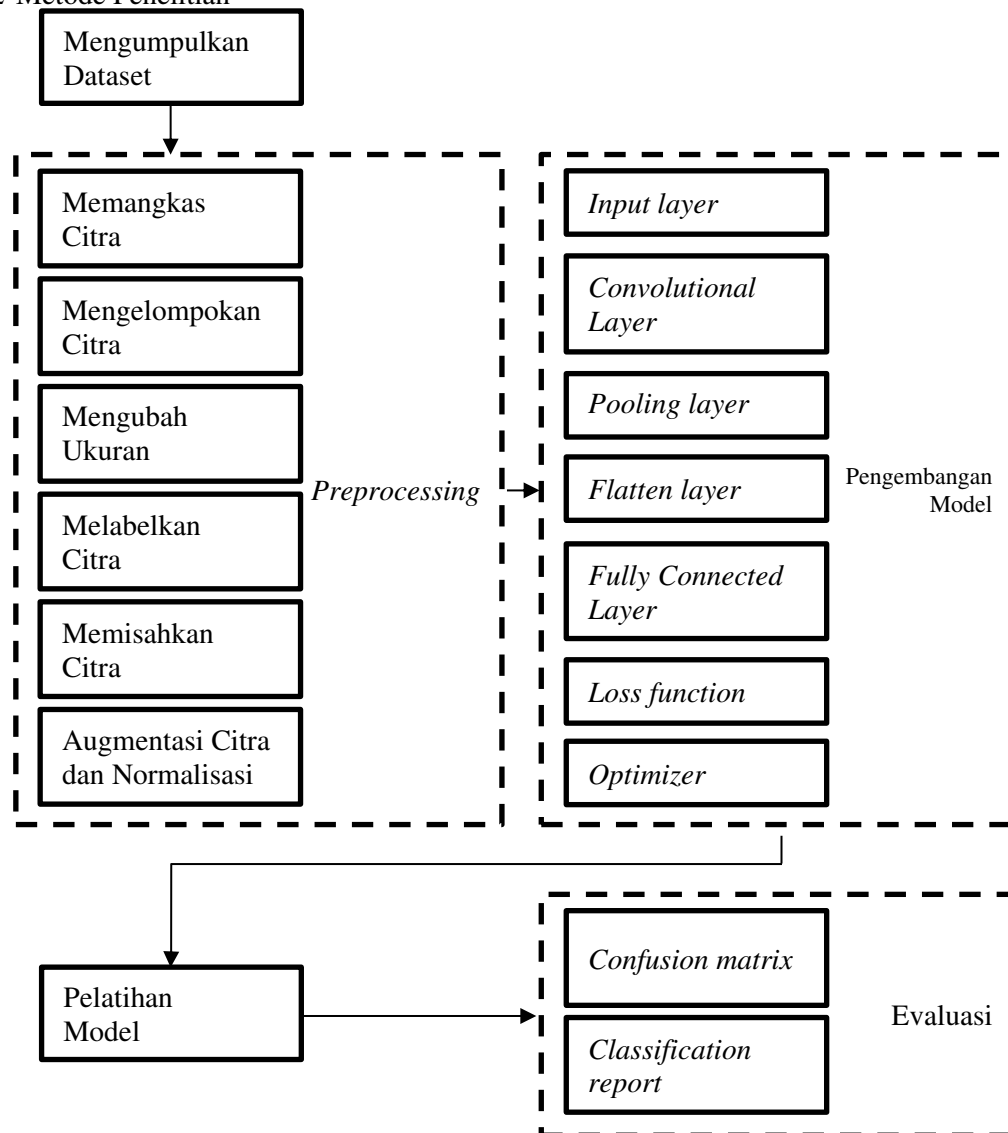
Penelitian ini menggunakan jenis penelitian kuantitatif dikarenakan jumlah data yang besar dan menghasilkan nilai numerik berupa akurasi. Pada penelitian ini data berupa citra digital yang berisikan informasi dalam bentuk matriks yang kemudian dilakukan perhitungan matematis, tujuan dari proses tersebut untuk mengklasifikasikan jenis tinta cartridge berdasarkan warna.

Jenis data pada penelitian ini adalah jenis data primer yang diambil menggunakan metode dokumentasi berdasarkan pengalaman penulis yang pernah bekerja sebagai checker barang keluar pada gudang barang jadi yang menjual tinta cartridge. Mengumpulkan dataset adalah langkah penulis melakukan pengambilan Citra barang keluar yang di ambil menggunakan kamera ponsel berdasarkan pengalaman penulis bekerja di gudang barang jadi, Dataset yang dikumpulkan berupa citra tinta cartridge HP 680 black color dan tri color.

Pada penelitian ini terbagi menjadi 2 percobaan, percobaan A menggunakan 500 citra HP 680 black color dan 500 citra HP 680 tri color, percobaan B menggunakan 600 citra HP 680 black color dan 600 citra HP 680 tri color.

Pada metode penelitian ini peneliti menjabarkan dalam dua bentuk yaitu diagram dan tabel, diagram untuk memvisualisasikan setiap langkah yang dilakukan dan tabel untuk penjelasan yang lebih detail.

2.2 Metode Penelitian



Gambar 1 Diagram Metode Penelitian

Tabel 1 Metode Penelitian

No	Langkah Penelitian	Metode/Proses	Input	Output
1	Mengumpulkan Dataset	Dokumentasi	Citra barang keluar yang di ambil menggunakan kamera ponsel berdasarkan pengalaman penulis bekerja di gudang barang jadi	Citra yang berjenis tinta <i>cartridge</i> HP 680 <i>black</i> dan 680 <i>tri color</i>
2	Memangkas Citra	<i>Manual Cropping</i>	Citra yang berjenis tinta <i>cartridge</i> HP 680 <i>black</i> dan 680 <i>tri color</i>	Citra tinta <i>cartridge</i> 680 <i>black</i> dan 680 <i>tri color</i> yang sudah dipangkas jadi per <i>item</i>
3	Mengelompokkan Citra	<i>Manual Classification</i>	Citra tinta <i>cartridge</i> 680 <i>black</i> dan 680 <i>tri color</i> yang sudah	Citra tinta <i>cartridge</i> yang sudah di

			dipangkas jadi per <i>item</i>	kelompokan kedalam <i>folder</i>
4	Mengubah Ukuran Citra	<i>Mass Resizing, Coding</i>	Citra tinta <i>cartridge</i> yang sudah di kelompokan	Citra tinta <i>cartridge</i> yang sudah dirubah ukurannya
5	Melabelkan citra	<i>Folder-based Labeling</i>	Citra tinta <i>cartridge</i> yang sudah dirubah ukurannya	Citra tinta <i>cartridge</i> yang sudah dilabelkan berdasarkan <i>folder</i>
6	Memisahkan Citra	<i>Random Splitting</i>	Citra tinta <i>cartridge</i> yang sudah dilabelkan berdasarkan <i>folder</i>	Citra tinta <i>cartridge</i> yang sudah di bagi secara <i>random</i>
7	Augmentasi Citra	<i>Rotation, Width Shift, Height Shift, Shear, Fill mode</i>	Citra tinta <i>cartridge</i> yang sudah di bagi secara <i>random</i>	Citra tinta <i>cartridge</i> yang bervariasi dari citra aslinya
8	Normalisasi	<i>Rescale</i>	Citra tinta <i>cartridge</i> yang bervariasi dari citra aslinya	Citra tinta <i>cartridge</i> yang sudah dirubah menjadi numerik
9	Memodelkan Arsitektur	<i>Input layer, Convolutional layer, Activation function, Pooling layer, Flatten layer, Fully connected layer, Optimizer, Loss function</i>	Konfigurasi arsitektur model	Model yang sudah dikonfigurasi arsitekturnya
10	Pelatihan Model	<i>Fitting, Epoch</i>	Citra tinta <i>cartridge</i> yang sudah dirubah menjadi numerik dan Model yang sudah dikonfigurasi arsitekturnya	Model yang sudah di latih
11	Evaluasi	<i>Training Validation Metrics Plotting, Confusion matrix, Classification report, Accuracy Evaluation</i>	Model yang sudah dilatih	Akurasi dan kesalahan untuk setiap label

Tabel 1 diatas adalah diagram alur yang digunakan dalam penelitian ini untuk menjawab pertanyaan penelitian yang ada, langkah utama pada kerangka penelitian dibagi jadi setiap langkah-langkah kecil. Pada tabel 3.1 diatas adalah penjelasan mendetail dari langkah penelitian yang dilakukan, terdapat input dan output dari setiap langkah.

2.3 Metode Pengumpulan Data

Metode pengumpulan data pada penelitian ini adalah metode dokumentasi berdasarkan pengalaman penulis yang pernah bekerja sebagai checker pada gudang barang jadi yang menjual tinta cartridge, data yang tercapture dan terkumpul pada ponsel penulis sebagai bukti untuk barang tersebut sesuai dengan barang yang dikirim.

Data dokumentasi ini bukan berupa asset perusahaan yang berharga dikarenakan data citra digital tinta cartridge ini bisa didapatkan di internet. Data yang digunakan pada penelitian ini terbagi dua yaitu sejumlah 600 citra black color dan 600 citra tri color.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Dataset

Dataset pada penelitian ini adalah citra tinta cartridge HP 680 black color dan tri color, dataset citra dikumpulkan dengan cara memotret dengan kamera ponsel. Citra dikumpulkan berdasarkan pengalaman penulis yang pernah bekerja sebagai checker barang keluar disebuah gudang yang menjual tinta cartridge HP 680 black color dan tri color.



Gambar 2 Hasil Tangkapan Citra

Pada gambar 2 diatas adalah hasil tangkapan citra barang keluar menggunakan ponsel. dalam satu citra bisa terdapat dua jenis black color dan tri color atau jumlah kuantitas yang lebih dari satu, maka dari itu penulis melakukan pemangkasan citra. Jumlah dataset yang akan digunakan pada penelitian ini adalah :

1. Percobaan A : 500 black color dan 500 tri color
2. Percobaan B : 600 black color dan 600 tri color

Perbedaan jumlah ini dikarenakan saran dari pembimbing skripsi agar ditambahkan jumlah datasetnya, dengan menggunakan Convolutional Neural Network , semakin banyak dataset yang gunakan untuk pelatihan machine learning maka semakin akurat pada hasil machine learning. semakin banyak dataset dan bervariasi maka komputer akan semakin banyak mengenal berbagai bentuk citra pixel yang bisa diprediksi nantinya.

3.2 Hasil *Preprocessing*

Preprocessing adalah pemrosesan data yakni gambar asli dini pada data sebelum diproses oleh Algoritma Convolutional Neural Network (E. Hermawan: 2021). Tujuan dari *preprocessing* adalah agar menghasilkan citra dengan kualitas yang lebih baik untuk proses berikutnya (A. Antoni, T. Rohana, and A. R. Pratama : 2023).

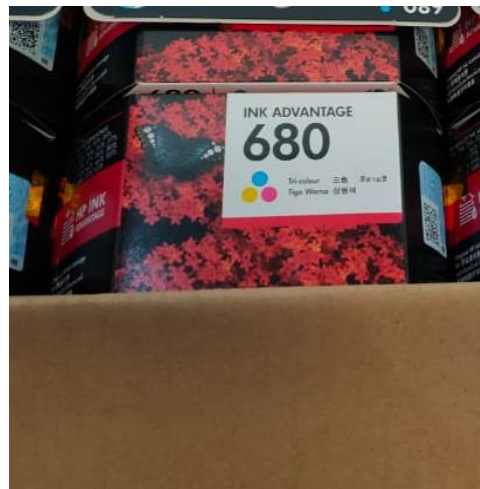
Preprocessing dilakukan kepada keseluruhan citra yang terdapat di dataset, *preprocessing* bertujuan untuk menggeneralisirkan citra yang bentuknya berbagai macam agar selaras dan bisa diinputkan kedalam model untuk dilakukan pelatihan. Metode yang digunakan yaitu memangkas citra, mengelompokkan citra, mengubah ukuran citra, melabelkan citra, memisahkan citra, augmentasi citra dan normalisasi.

3.2.1 Memangkas Citra

Pemangkasan citra dilakukan karena hasil citra yang diambil menggunakan ponsel masih tergabung antara black color dan tri color dalam satu citra serta jumlah kuantitas dari setiap jenis pada citra lebih dari satu, pemangkasan ini dilakukan agar dalam satu citra terdapat satu jenis dan satu kuantitas tinta cartridge HP 680. Pada langkah ini menggunakan tools yang terdapat di windows 10 yaitu snipping tools berguna untuk memangkas citra.



Gambar 3 Citra Sebelum Dipangkas



Gambar 4 Citra Sesudah Dipangkas

Pada gambar 3 diatas adalah hasil tangkapan citra menggunakan kamera ponsel yang belum dipangkas, pada gambar 4 adalah hasil dari langkah pemangkasan citra. Pada langkah ini penulis menggunakan metode manual cropping yaitu melakukan pemangkasan pada semua citra dengan satu persatu menggunakan snipping tools.

3.2.2 Mengelompokkan Citra

Pada langkah ini citra yang sudah dipangkas dikelompokkan ke dalam direktori sesuai dari jenis pada citranya, mengelompokkan citra bertujuan untuk mempermudah penulis dalam melakukan proses coding.



Gambar 5 Hasil Pengelompokan Citra Black Color



Gambar 6 Hasil Pengelompokan Citra Tri Color

Pada gambar 5 dan gambar 6 adalah hasil pada langkah pengelompokan citra, terdapat dua direktori yaitu black dan tri_color. Pada langkah ini penulis menggunakan metode manual classification yaitu melihat satu persatu dari citra yang sudah dipangkas lalu mengelompokkan sesuai jenisnya.

3.2.3 Mengubah Ukuran Citra

Pada langkah ini penulis mengubah ukuran citra untuk menselaraskan ukuran panjang dan lebar dari keseluruhan citra, mengubah ukuran citra dikarenakan hasil dari pemangkasan pada langkah sebelumnya tidak selaras ukuran dari keseluruhan citra. Penulis mengubah ukuran citra dilakukan dengan dua cara yaitu pada percobaan A menggunakan bahasa pemrograman python karena dirubah dilevel code dan pada percobaan B menggunakan tools powertoys yang terdapat di windows 10.

height=250

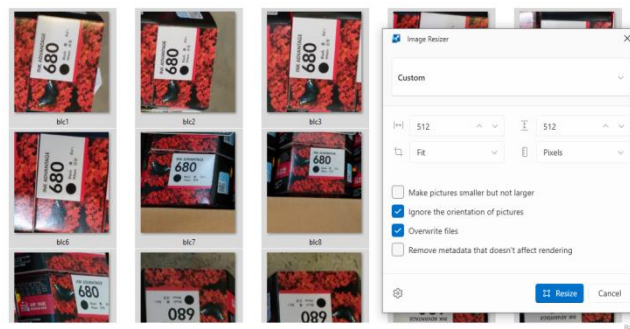
width=250

channels=3

batch_size=60

img_shape=(height, width, channels)

img_size=(height, width)



Gambar 7 Mengubah Ukuran Citra Percobaan B

Pada gambar 7 adalah langkah mengubah ukuran citra pada percobaan A dilevel code, terdapat keseluruhan citra diubah menjadi 250 x 250 pixel. Pada gambar 4.7 adalah langkah mengubah ukuran citra pada percobaan B menggunakan tools powertoys yang terdapat fitur image resizernya, terdapat keseluruhan citra diubah menjadi 512 x 512 pixel. Perbedaan ukuran dari setiap percobaan ini dikarenakan saran pembimbing skripsi untuk mengubah ukuran pada percobaan B menjadi 512 x 512 pixel berdasarkan ukuran dataset citra di komunitas kaggle yang banyak digunakan. berdasarkan sumber argument paper berikut menjelaskan resolusi 512 pixel lebih baik dari 256 pixel, <https://www.kaggle.com/competitions/google-research-identify-contrails-reduce-global-warming/discussion/430471> (diakses, 31 Juli 2024).

3.3 Pelatihan Model

Pada langkah ini penulis melakukan pelatihan pada model yang sudah dibuat arsitektur pada langkah sebelumnya dengan citra hasil preprocessing.

```
epochs=50
history=model.fit(x=train_gen, epochs=epochs, validation_data=valid_gen)
```

adalah code untuk melakukan pelatihan model dengan memanggil model.fit(x = train_gen, epochs = epochs, validation_data = valid_gen), pada code tersebut terdapat parameter x = train_gen berfungsi untuk memasukan input citra hasil preprocessing, parameter epochs berfungsi untuk mengkonfigurasi berapa banyak iterasi yang akan dilakukan selama pelatihan, dan parameter validation_data berfungsi untuk memasukan input citra validation.

```
14/14 ————— 97s 5s/step - accuracy: 0.5028 - loss: 0.6937 - val_accuracy: 0.5600 - val_loss: 0.6855
Epoch 2/50
14/14 ————— 89s 5s/step - accuracy: 0.5559 - loss: 0.6906 - val_accuracy: 0.5200 - val_loss: 0.6844
Epoch 3/50
14/14 ————— 90s 5s/step - accuracy: 0.5200 - loss: 0.6780 - val_accuracy: 0.6100 - val_loss: 0.6262
Epoch 4/50
14/14 ————— 89s 5s/step - accuracy: 0.6302 - loss: 0.6233 - val_accuracy: 0.6700 - val_loss: 0.5877
Epoch 5/50
14/14 ————— 96s 5s/step - accuracy: 0.6519 - loss: 0.6015 - val_accuracy: 0.6400 - val_loss: 0.6096
Epoch 46/50
14/14 ————— 89s 5s/step - accuracy: 0.9226 - loss: 0.1977 - val_accuracy: 0.9200 - val_loss: 0.2199
Epoch 47/50
14/14 ————— 88s 4s/step - accuracy: 0.9433 - loss: 0.1533 - val_accuracy: 0.9500 - val_loss: 0.1952
Epoch 48/50
14/14 ————— 90s 5s/step - accuracy: 0.9266 - loss: 0.2031 - val_accuracy: 0.9200 - val_loss: 0.1949
Epoch 49/50
14/14 ————— 90s 5s/step - accuracy: 0.9132 - loss: 0.2234 - val_accuracy: 0.9300 - val_loss: 0.2339
Epoch 50/50
14/14 ————— 89s 5s/step - accuracy: 0.9497 - loss: 0.1539 - val_accuracy: 0.9000 - val_loss: 0.4256
```

Gambar 8 Hasil Pelatihan Model

Pada gambar 8 adalah hasil dari pelatihan model yaitu model akan melakukan pelatihan dan mengembalikan output nilai accuracy serta loss pada pelatihan dan validasi disetiap iterasinya.

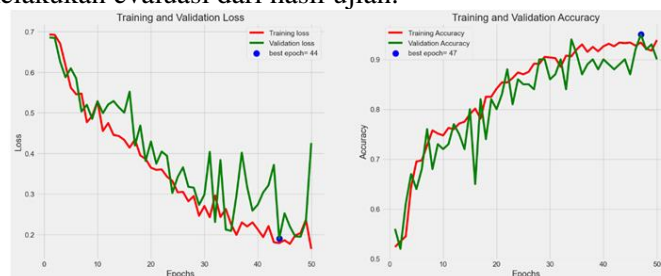
3.4 Evaluasi

Pada langkah ini penulis melakukan ujian pada hasil model yang sudah dilatih di langkah sebelumnya, model akan diuji menggunakan citra yang belum pernah dilihat atau dipelajari model selama pelatihan yaitu data test.

Pada langkah ini juga penulis melakukan evaluasi dari hasil pelatihan dan ujian.

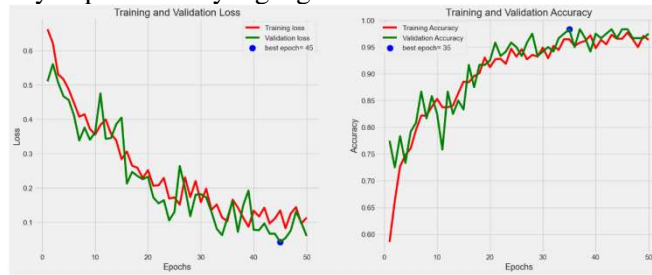
```
tr_plot(history,0)
save_dir=r'./'
subject='tinta'
acc=model.evaluate( test_gen, batch_size=test_batch_size, verbose=1, steps=test_steps,
return_dict=False)[1]*100
msg=f'accuracy on the test set is {acc:5.2f} %'
print_in_color(msg, (0,255,0),(55,65,80))
save_id=str (model_name + '-' + subject + '-' + str(acc)[:str(acc).rfind('.')+3] + '.h5')
save_loc=os.path.join(save_dir, save_id)
```

adalah code untuk memanggil function tr_plot yang berfungsi untuk menampilkan visualisasi data dari hasil pelatihan setiap epochsnya. Pada code tersebut juga memanggil model.evaluate berfungsi untuk melakukan evaluasi dari hasil ujian.



Gambar 9 Hasil Evaluasi Pelatihan dan Validasi Percobaan A

Pada gambar 9 adalah hasil visualisasi data accuracy dan loss pada saat pelatihan dan validasi percobaan A, terlihat untuk accuracy dan loss pada pelatihan cukup baik tidak terjadi perbedaan yang signifikan pada setiap epochsnya. Pada accuracy dan loss pada validasi disetiap epochsnya terlalu banyak perbedaan yang signifikan.



Gambar 10 Hasil Evaluasi Pelatihan dan Validasi Percobaan B

Pada gambar 10 adalah hasil visualisasi data accuracy dan loss pada saat pelatihan dan validasi percobaan B, terlihat untuk accuracy dan loss pada pelatihan cukup baik tidak terjadi perbedaan yang signifikan pada setiap epochsnya. Pada accuracy dan loss pada validasi disetiap epochsnya tidak terlalu banyak perbedaan yang signifikan walaupun hasil lossnya tetap turun dan hasil accuracynya tetap naik.

2/2 ————— 7s 3s/step - accuracy: 0.9533 - loss: 0.1992
accuracy on the test set is 96.00 %

Gambar 11 Hasil Evaluasi Ujian Percobaan A

2/2 ————— 143s 44s/step - accuracy: 0.9722 - loss: 0.1861
accuracy on the test set is 96.67 %

Gambar 12 Hasil Evaluasi Ujian Percobaan B

3.5 Analisis Hasil

Dari pembahasan dan hasil diatas terdapat perbedaan jumlah dataset, jumlah batch size, ukuran citra dan arsitektur model yang bisa dilihat pada table 2.

Tabel 2 Analisis Perbedaan

Perbedaan	Percobaan A	Percobaan B
Jumlah Dataset	500 citra <i>black color</i> dan 500 citra <i>tri color</i>	600 citra <i>black color</i> dan 600 citra <i>tri color</i>
Jumlah <i>Batch size</i>	<i>Batch_size</i> = 60	<i>Batch_size</i> = 24
Ukuran Citra	250 x 250 pixel	512 x 512 pixel
Arsitektur Model	Tidak menggunakan metode <i>Dropout</i>	Menggunakan metode <i>Dropout</i>

Pada tabel 2 terdapat beberapa perbedaan pada setiap percobaan jumlah dataset yang berfungsi sebagai input citra untuk melatih model, jumlah batch size pada percobaan B turun menjadi 24 karena ukuran citra pixel yang besar menjadikan perangkat keras yang digunakan penulis belum mampu untuk menggunakan batch size seperti percobaan A, ukuran citra yang berubah sesuai saran dari dosen pembimbing, dan penambahan metode dropout pada percobaan B untuk mengurangi overfitting. Berdasarkan hasil dan perbedaan diatas accuracy terbaik didapat oleh model percobaan B dengan accuracy 96.67% sedangkan percobaan A dengan accuracy 96.00%. berdasarkan hasil penelitian relevan terdahulu nilai tersebut baik dalam klasifikasi citra. Peneliti mentargetkan akurasi yang ingin dicapai 90% keatas sesuai hasil penelitian terdahulu untuk mengklasifikasikan barang yang rata-rata masih diantara 80% sampai 95%.

4. KESIMPULAN

Penelitian yang dilakukan dapat disimpulkan, bahwa dataset yang digunakan untuk mengembangkan model Deep Learning klasifikasi tinta cartridge adalah citra tinta cartridge merk HP 680 black color dan HP 680 tri color. Citra dikumpulkan dengan cara memotret barang dengan kamera ponsel, dataset dari penelitian ini dapat diakses secara public diplatform github. Pada penelitian ini terdapat dua dataset, yaitu percobaan A dengan dataset berjumlah 500 black color dan 500 tri color, sedangkan percobaan B dataset berjumlah 600 black color dan 600 tri color.

Dalam mengembangkan model Convolutional Neural Network yang sesuai untuk klasifikasi citra tinta cartridge berdasarkan penelitian ini, yaitu preprocessing, memodelkan arsitektur, pelatihan model dan evaluasi. Pada langkah preprocessing dilakukan dengan cara memangkas citra, mengelompokkan citra, mengubah ukuran citra, melabelkan citra, memisahkan citra, augmentasi citra dan normalisasi. Pada langkah memodelkan arsitektur, arsitektur model yang digunakan, yaitu input layer, convolutional layer, pooling layer, dropout, flatten layer, fully connected layer, loss function dan optimizer. Pada langkah pelatihan model, hasil arsitektur model yang sudah dibuat akan dilatih dengan citra hasil preprocessing, pada percobaan A dan percobaan B pelatihan dilakukan sebanyak 50 kali. Sedangkan langkah evaluasi dilakukan dengan melihat hasil akurasi model dalam memprediksi data test menggunakan confusion matrix dan classification report.

Hasil akurasi pada penelitian ini terbagi menjadi dua berdasarkan percobaan A dan percobaan B. terdapat beberapa perbedaan pada setiap percobaan jumlah dataset yang berfungsi sebagai input citra untuk melatih model, jumlah batch size pada percobaan B turun menjadi 24 karena ukuran citra pixel yang besar menjadikan perangkat keras yang digunakan penulis belum mampu untuk menggunakan batch size seperti dipercobaan A, ukuran citra yang berubah sesuai saran dari dosen pembimbing, dan penambahan metode dropout pada percobaan B untuk mengurangi overfitting. Berdasarkan hasil dan perbedaan diatas akurasi terbaik didapat oleh model percobaan B dengan akurasi 96.67% sedangkan percobaan A dengan akurasi 96.00%, berdasarkan penelitian relevan terdahulu peneliti mentargetkan nilai akurasi yang ingin dicapai adalah 90%. Dapat disimpulkan berdasarkan akurasi tersebut bahwa akurasi model yang didapat pada penelitian ini sudah sesuai target yang ingin dicapai.

DAFTAR PUSTAKA

- Awaludin, M., & Rahwanto, Y. Z. (2017). *Pengembangan Algoritma Neural Network Berdasarkan Rentang*. 10(2).
- I. Cholissodin and A. A. Soebroto, "AI , MACHINE LEARNING & DEEP LEARNING (Teori & Implementasi)," no. January, 2021.
- E. Hermawan, "Klasifikasi Pengenalan Wajah Menggunakan Masker atau Tidak Dengan Mengimplementasikan Metode CNN (Convolutional Neural Network)," *J. Ind. Kreat. Dan Inform.*, vol. 1, pp. 33–43, 2021.
- A. Antoni, T. Rohana, and A. R. Pratama, "Implementasi Algoritma Convolutional Neural Network Untuk Klasifikasi Citra Kemasan Kardus Defect dan No Defect," *Build. Informatics, Technol. Sci.*, vol. 4, no. 4, pp. 1941–1950, 2023, doi: 10.47065/bits.v4i4.3270.
- I. Cholissodin and A. A. Soebroto, "AI , MACHINE LEARNING & DEEP LEARNING (Teori & Implementasi)," no. January, 2021.
- B. Mahesh, "Machine Learning Algorithms - A Review," no. October, 2020, doi: 10.21275/ART20203995.
- N. Jain and R. Kumar, "A Review on Machine Learning & It ' s Algorithms," no. December, 2022, doi: 10.35940/ijsc.E3583.1112522.
- S. Yuliany, Aradea, and Andi Nur Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *J. Buana Inform.*, vol. 13, no. 1, pp. 54–65, 2022, doi: 10.24002/jbi.v13i1.5022.
- Y. B. E. Purba, N. F. Saragih, A. P. Silalahi, and ..., "Perancangan Alat Pendeteksi Kematangan Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode Convolutional Neural Network (CNN)," *J. Ilm. Tek. ...*, vol. 2, no. 1, pp. 13–21, 2022, [Online]. Available: <https://ojs.fikom-methodist.net/index.php/methotika/article/view/43%0Ahttps://ojs.fikom-methodist.net/index.php/methotika/article/download/43/39>
- [M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, "A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery," *Remote Sens.*, vol. 9, no. 2, 2017, doi: 10.3390/rs9020100.
- T. Zhi, L. Y. Duan, Y. Wang, and T. Huang, "Two-stage pooling of deep convolutional features for image retrieval," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2016-Augus, pp. 2465–2469, 2016, doi: 10.1109/ICIP.2016.7532802.
- F. Loekman, "Sistem Manajemen Inventori Dengan Pengenalan Barang Secara Otomatis Menggunakan Metode Convolutional Neural Network Inventory Management System with Automatic Recognition of Goods Using the Convolutional Neural Network Method," vol. 12, no. 1, pp. 47–56, 2023, doi: 10.34148/teknika.v12i1.596.