



Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah

Rima Dias Ramadhani¹, Afandi Nur Aziz Thohari², Condro Kartiko³, Apri Junaidi⁴, Tri Ginanjar Laksana⁵,
Novanda Alim Setya Nugraha⁶

^{1,4}Program Studi Sains Data, Fakultas Informatika, Institut Teknologi Telkom Purwokerto

²Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Semarang

³Program Studi Rekayasa Perangkat Lunak, Fakultas Informatika, Institut Teknologi Telkom Purwokerto

^{5,6}Program Studi Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom Purwokerto

¹rima@ittelkom-pwt.ac.id, ²afandi@polines.ac.id, ³condro.kartiko@ittelkom-pwt.ac.id, ⁴apri@ittelkom-pwt.ac.id,

⁵anjarlaksana@ittelkom-pwt.ac.id, ⁶novanda@ittelkom-pwt.ac.id

Abstract

Waste is goods / materials that have no value in the scope of production, where in some cases the waste is disposed of carelessly and can damage the environment. The Indonesian government in 2019 recorded waste reaching 66-67 million tons, which is higher than the previous year, which was 64 million tons. Waste is differentiated based on its type, namely organic and anorganic waste. In the field of computer science, the process of sensing the type waste can be done using a camera and the Convolutional Neural Networks (CNN) method, which is a type of neural network that works by receiving input in the form of images. The input will be trained using CNN architecture so that it will produce output that can recognize the object being inputted. This study optimizes the use of the CNN method to obtain accurate results in identifying types of waste. Optimization is done by adding several hyperparameters to the CNN architecture. By adding hyperparameters, the accuracy value is 91.2%. Meanwhile, if the hyperparameter is not used, the accuracy value is only 67.6%. There are three hyperparameters used to increase the accuracy value of the model. They are dropout, padding, and stride. 20% increase in dropout to increase training overfit. Whereas padding and stride are used to speed up the model training process.

Keywords: CNN, dropout, model, padding, waste, stride

Abstrak

Sampah merupakan barang/bahan yang tidak memiliki nilai dalam lingkup produksi, dimana dalam beberapa kasus sampah dibuang sembarangan dan dapat merusak lingkungan. Pemerintah Indonesia tahun 2019 mencatat sampah mencapai 66-67 juta ton, dimana lebih tinggi dibandingkan tahun sebelumnya yaitu 64 juta ton. Sampah dibedakan berdasarkan jenisnya yaitu sampah organik dan anorganik. Pada bidang ilmu komputer, proses penginderaan jenis dan bentuk sampah dapat dilakukan menggunakan kamera dan metode *Convolutional Neural Networks* (CNN) yang merupakan jenis *neural network* yang bekerja dengan cara menerima masukan berupa citra. Masukan tersebut akan di *training* menggunakan arsitektur CNN sehingga akan menghasilkan output yang dapat mengenali objek yang diinputkan. Pada penelitian ini dilakukan optimasi penggunaan metode CNN untuk mendapatkan hasil yang akurat dalam mengidentifikasi jenis sampah. Optimasi dilakukan dengan menambah beberapa *hyperparameter* pada arsitektur CNN. Dengan menambahkan *hyperparameter* diperoleh nilai akurasi yang tinggi yaitu 91,2%. Sedangkan apabila tidak menggunakan *hyperparameter* nilai akurasi hanya sebesar 67,6%. Terdapat tiga *hyperparameter* yang digunakan untuk menaikkan nilai akurasi model yaitu *dropout*, *padding*, dan *stride*. Penambahan *dropout* sebesar 20% untuk meningkatkan overfitting saat pelatihan. Sedangkan *padding* dan *stride* digunakan untuk mempercepat proses pelatihan model.

Kata kunci: CNN, dropout, model, padding, sampah, stride.

1. Pendahuluan

Sampah merupakan barang/bahan yang tidak memiliki nilai yang digunakan secara biasa maupun khusus

dalam lingkup produksi, barang rusak selama manufaktur, atau materi berlebihan yang sebagian besar berasal dari rumah tangga [1]. Sampah tersebut dibuang sembarangan ke berbagai tempat atau dibakar disekitar

tempat tinggal warga yang efeknya akan merusak lingkungan sekitar [2]. Di Indonesia, sampah merupakan salah satu permasalahan yang sangat serius. Hal ini dibuktikan dengan data yang diperoleh dari Badan Pusat Statistik (BPS) Tahun 2018 dimana masyarakat Indonesia masih sulit dalam melakukan pemilahan sampah dan tidak tahu manfaat dari sampah yang dibuang. Data yang diperoleh dari Kementerian Lingkungan Hidup dan Kehutanan pada tahun 2019 sampah di Indonesia mencapai 66-67 juta ton, dimana jumlah ini lebih tinggi dibandingkan dengan tahun sebelumnya yang mencapai 64 juta ton. Persentase sampah organik sebesar 60%, sampah plastik mencapai 15%.

Menurut penelitian [2] dalam penanganan dan pengolahannya sampah digolongkan menjadi dua jenis yaitu sampah organik dan anorganik. Sampah organik umumnya sampah yang mengandung senyawa organik yang dapat terurai oleh mikroorganisme seperti sisa makanan, karton, kain, karet, kulit, sampah halaman, dan lainnya. Selain itu, sampah organik biasanya berwarna hijau atau coklat, dan bentuknya tidak beraturan. Sampah anorganik mengandung bahan yang bersifat anorganik dan sulit terurai oleh mikroorganisme seperti kaca, kaleng, aluminium, debu, dan logam lainnya. Selain itu, sampah anorganik biasanya berwarna putih atau biru, bentuk padat atau lebih solid.

Meskipun sampah sudah diklasifikasikan masih terdapat pembuangan yang tidak sesuai dengan jenisnya sehingga membuat tumpukan sampah menjadi meningkat serta tidak diimbangi dengan pengolahan yang baik akan menimbulkan berbagai permasalahan kembali [2]. Berdasarkan permasalahan dalam pengelolaan jenis sampah diperlukan sebuah edukasi dalam deteksi gambar jenis sampah agar dalam pembuangan dan pengelolaannya sesuai dengan peruntukannya menggunakan sebuah aplikasi.

Berdasarkan permasalahan tersebut, solusi yang ditawarkan dalam penelitian ini adalah dengan memanfaatkan teknologi sistem cerdas yang dapat mengidentifikasi gambar sampah menggunakan aplikasi mobile. Terdapat beberapa algoritma untuk klasifikasi gambar *K-means*, *Support Vector Machine* (SVM), dan *Convolutional Neural Network* (CNN) [3]. Namun, dari ketiga algoritma tersebut CNN yang paling banyak di gunakan untuk mendeteksi Gambar [4].

CNN bekerja dengan cara menerima input berupa *image*, input akan di *training* dalam beberapa layer seperti *softmax* sehingga menghasilkan output yang dapat mengenali object yang di inputkan [5]. CNN merupakan klasifikasi gambar yang diambil dari sebuah inputan gambar yang kemudian diproses dan diklasifikasikan. CNN terdiri dari neuron yang memiliki *weight*, bias dan *activation function* [6]. Namun yang

membedakan dengan *neural network* biasa adalah jika menggunakan *neural network* biasa mungkin hanya memuat skala panjang dan tinggi. Namun, CNN bisa memuat semua informasi dari keseluruhan skala yang bisa mengklasifikasikan objek dengan lebih akurat karena bisa menggunakan skala lebarnya juga [6] [7].

Adapun pada penelitian [8] CNN telah digunakan dalam membantu meningkatkan sistem pengelolaan sampah dan membantu menciptakan *smart city*. Dua *Convolutional Neural Networks* (CNN), keduanya didasarkan pada arsitektur jaringan AlexNet, dikembangkan untuk mencari objek sampah dalam gambar dan memisahkan item yang dapat didaur ulang dari objek sampah TPA. Sistem CNN dua tahap pertama kali dilatih dan diuji pada kumpulan data gambar dalam ruangan TrashNet *benchmark* dengan akurasi yang mencapai 93,6%. Kemudian sistem dilatih dan diuji pada gambar luar ruangan yang diambil oleh penulis di lingkungan penggunaan yang dimaksudkan menggunakan dataset <https://github.com/garythung/trashnet> yang berisi 2527 data yang berisi gambar luar ruangan dan 2390 berasal dari dalam ruangan yang berisi berbagai macam barang sampah seperti (gelas, kertas, *cardboard*, plastik, logam, dan sampah lainnya) dengan akurasi berkisar antara 89,7% hingga 93,4% dan secara keseluruhan 92%.

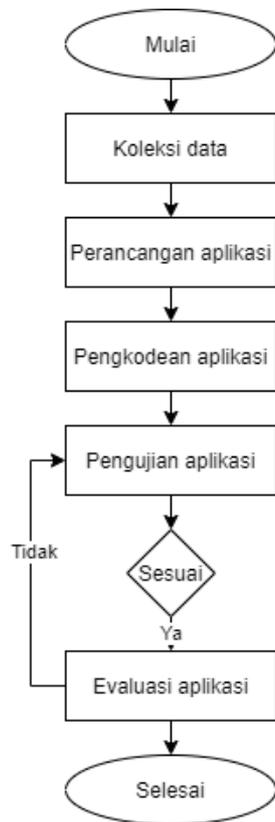
Pada penelitian [9] yang mendeteksi gambar daur ulang atau sampah dan mengklasifikasikannya menjadi enam kelas yang terdiri dari gelas, kertas, logam, plastik, karton dan sampah. Dalam penelitian tersebut juga menciptakan dataset masing-masing berisi sekitar 400-500 gambar. Model yang digunakan adalah *Support Vector Machine* (SVM) dengan *Scale-Invariant Feature Transformations* (SIFT) dan *Convolutional Neural Networks* (CNN). Hasil yang diperoleh, dalam penelitian tersebut SVM bekerja lebih baik dari CNN dikarenakan CNN tidak sepenuhnya terlatih kemampuan karena kesulitan menemukan *hyperparameter* yang optimal. Selain itu, CNN juga dapat diterapkan di bidang *Natural Processing Language* (NLP), dimana memanfaatkan berbagai layer dengan *filter convolving* yang diaplikasikan untuk *fitur local* [10]. Dalam penelitian tersebut model CNN telah terbukti efektif untuk NLP dan telah mencapai hasil yang sangat baik dalam penguraian semantik, pencarian permintaan pencarian, pemodelan kalimat, dan tugas NLP tradisional lainnya.

Berdasarkan latar belakang di atas, pada penelitian ini dilakukan optimasi penggunaan metode CNN untuk mendapatkan hasil yang akurat dalam mengidentifikasi jenis sampah yaitu sampah organik dan anorganik. Optimasi dilakukan dengan penambahan beberapa *hyperparameter* pada arsitektur CNN. Terdapat tiga *hyperparameter* yang digunakan untuk meningkatkan nilai akurasi model yaitu *dropout*, *padding*, dan *stride*.

Selain itu, penelitian ini juga menjelaskan hasil analisis *loss* dan akurasi atas penambahan *hyperparameter* dan tanpa *hyperparameter*.

2. Metode Penelitian

Metode yang digunakan dalam penelitian ini meliputi beberapa tahap yang dapat dilihat pada Gambar 1, diantaranya:



Gambar 1. Diagram Alur Penelitian

2.1. Proses Persiapan

Berdasarkan proses persiapannya dalam penelitian dibagi menjadi 2 (dua) yaitu proses koleksi data dan perancangan aplikasi. Koleksi data dilakukan dari sumber dataset yang diperoleh dari *google image* dan diambil dengan *extension chrome fatkun batch image downloader* yang bertujuan agar menghasilkan scrapping data yang lebih banyak dari *image* yang diunduh. Adapun dataset tersebut kemudian dilakukan *preprocessing* yang digunakan untuk menghilangkan *noise* pada data.

Adapun konsep akhir yang dirancang diarahkan ke dalam perancangan aplikasi dengan memetakan kebutuhan *design* antarmuka aplikasi menggunakan *Adobe Photoshop* untuk rancangan *design* dan *usability* sesuai dengan kebutuhan pengguna. Selain itu, dalam proses perancangannya dalam *machine learning*

menggunakan python dengan library *Tensor Flow* dan *Keras*.

2.2. Pengkodean

Berdasarkan hasil koleksi data dan perancangan aplikasi, hasil akhir yang diharapkan adalah adanya klasifikasi jenis sampah yaitu organik dan anorganik. Pada penelitian ini algoritma CNN diimplementasikan dengan library *Tensor Flow* dan *Keras*. Gambar 2 menunjukkan proses pemodelan CNN yang digunakan dalam penelitian ini hingga penerapan ke dalam model Android.

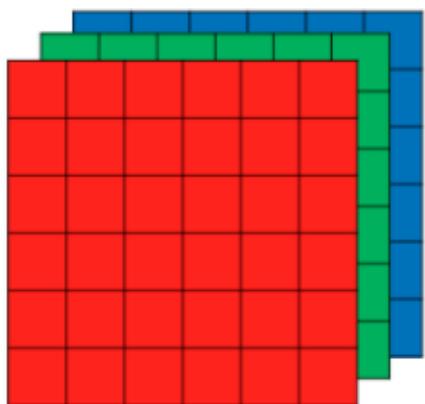


Gambar 2. Pemodelan CNN

Dataset yang digunakan berupa gambar sampah yang belum diklasifikasikan yang diambil dari objek sampah sebanyak organik sebanyak 840 gambar dan anorganik sebanyak 840 gambar. Setelah mendapatkan data citra sampah organik dan anorganik. Langkah selanjutnya adalah membagi dataset tersebut kedalam *training* dan *validation*. Perbandingan data *training* dan *validation* adalah 80% untuk *training* dan 20% untuk *validation*. Pembagian dataset ini dilakukan secara random menggunakan library *split_folder*.

Selanjutnya dilakukan *preprocessing* agar citra sampah yang dimiliki memiliki ukuran dan pola yang sama. Teknik *preprocessing* dilakukan dengan menggunakan library ImageDataGenerator, dimana citra akan di *resize*, *flip*, *rotate* untuk mendapatkan ukuran dan bentuk yang sama. Tujuan tahap *preprocessing* ini agar citra yang digunakan sebagai input dapat langsung di proses oleh ke dalam algoritma CNN. Pada proses ini ukuran citra di set adalah 150x150 piksel.

Berbagai macam penelitian mengenai metode deep learning saat ini sudah masuk di area objek deteksi termasuk ekstraksi fitur dan klasifikasi menggunakan *Convolutional Neural Networks* (CNN) dengan akurasi yang cukup tinggi dari hasil penelitian yang ada. Berbagai permasalahan yang muncul seperti gambar yang terlalu besar menyebabkan akurasi dari klasifikasi tidak terlalu baik. Pelabelan dengan tepat menjadi kunci klasifikasi berjalan dengan baik [11]. *Convolutional Neural Network* (CNN) banyak di gunakan untuk mendeteksi *image* [4] CCN berkerja dengan cara menerima input berupa *image*, input akan di *training* dalam beberapa layer seperti *softmax* sehingga menghasilkan output yang dapat mengenali object yang di inputkan [5]. CNN merupakan klasifikasi gambar yang diambil dari sebuah inputan gambar yang kemudian diproses dan diklasifikasikan. Program akan membaca data tersebut sebagai input gambar berupa array dari beberapa piksel dan resolusi dalam perbesar 6 x 6 x 3 seperti yang terlihat pada Gambar 3.

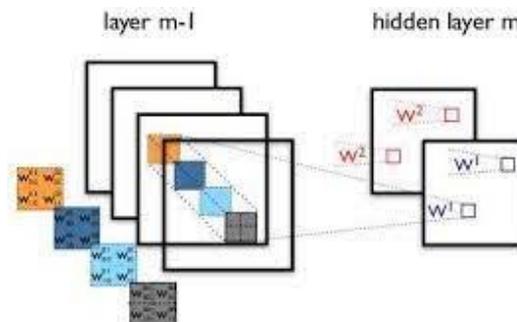


Gambar 3. Dimensi 6 x 6 x 3

CNN akan melatih dan menguji dataset, setiap masukan gambar akan melalui sekelompok *convolution layer* dengan nilai probabilitas antara 0 dan 1, Karena sifat proses konvolusi, maka CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara, seperti yang ditunjukkan pada Gambar 4.

Confusion matrix diterapkan dalam pengukuran pencarian hasil akurasi untuk mengetahui keandalan dari algoritma yang telah dijalankan khususnya pada pemodelan klasifikasi. Pada dasarnya *confusion matrix*

mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya [10]. Pada pengukuran kinerja menggunakan *confusion matrix*, seperti yang terlihat pada Tabel 1.



Gambar 4. Proses Konvolui pada CNN

Tabel 1. Confussion Matrix

Class	Kelas Positif	Kelas Negatif
Positif	TP (True Positive)	TN (True Negative)
Negatif	FP (False Positive)	FN (False Negative)

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (1)$$

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad (2)$$

$$Recall = \frac{TP}{FN+TP} * 100\% \quad (3)$$

Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah seperti, nilai True Negative (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, nilai False Positive (FP) merupakan data negatif namun terdeteksi sebagai data positif, nilai True Positive (TP) merupakan data positif yang terdeteksi benar, dan nilai False Negative (FN) merupakan kebalikan dari True Positive, sehingga data positif, namun terdeteksi sebagai data negatif.

2.3. Pengujian Aplikasi

Uji aplikasi dilakukan melalui perangkat *smartphone* android yang memiliki spesifikasi dan layar berbeda untuk mendapatkan bug dari aplikasi tanpa harus melakukan uji coba menggunakan banyak perangkat. Alat pengujian menggunakan Espresso.

Pada tahap ini akan terlihat hasil dari akurasi model yang digunakan. Selain itu, akan dievaluasi juga berdasarkan hasil dari pengujian aplikasi dengan melihat efektifitas implementasi aplikasi yang diusulkan, jika ditemukan ketidakefektifan dari aplikasi ini maka dilakukan perbaikan. Namun, jika aplikasi ini dinilai efektif maka akan dilakukan penambahan fitur lainnya sesuai dengan kebutuhan.

3. Hasil dan Pembahasan

Setelah dilakukan perancangan arsitektur CNN. Langkah selanjutnya adalah melakukan pelatihan pada data yang diperoleh untuk mendapatkan model. Pada saat pertama kali dilakukan pelatihan (belum dilakukan optimasi), diperoleh model yang belum akurat. Setelah itu dilakukan optimasi dengan menambah beberapa *hyperparameter*, sehingga didapatkan hasil yang akurat.

3.1. Perbandingan Loss dan Akurasi Pelatihan

Proses pelatihan merupakan proses untuk mendapatkan model pembelajaran mesin. Pada saat pelatihan telah diatur untuk jumlah *step* per epoch adalah 25, dengan waktu menyelesaikan epoch adalah 3 detik. Penggunaan *padding* dan *stride* merupakan penyebab waktu eksekusi menjadi lebih cepat. Sebab pengujian tanpa menggunakan *padding* dan *stide* membuat waktu eksekusi per epoch sebesar 7 detik. Hasil pengujian *loss* dan akurasi tiap epoch pada model tanpa *hyperparameter* ditunjukkan pada Tabel 2, sedangkan model dengan *hyperparameter* ditunjukkan pada Tabel 3.

Tabel 2. Loss dan Akurasi Model Tanpa *Hyperparameter*

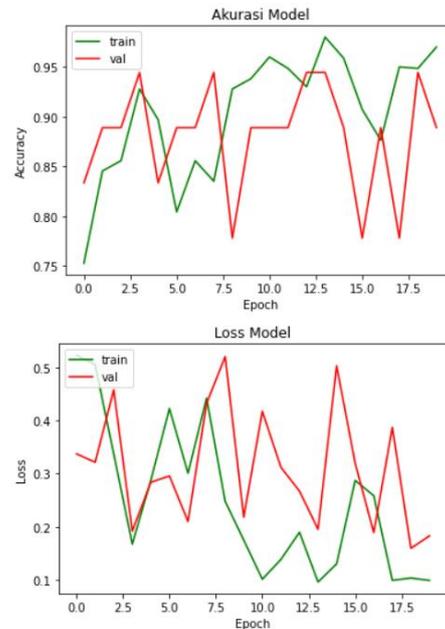
Epoch	Waktu	Loss dan Akurasi
1/20	8s 295ms/step	loss: 0.6126 - accuracy: 0.6161
2/20	7s 267ms/step	loss: 0.6248 - accuracy: 0.8132
3/20	7s 262ms/step	loss: 0.2948 - accuracy: 0.9137
4/20	7s 265ms/step	loss: 0.2423 - accuracy: 0.8920
5/20	7s 261ms/step	loss: 0.2868 - accuracy: 0.8932
6/20	7s 263ms/step	loss: 0.2803 - accuracy: 0.8931
7/20	7s 261ms/step	loss: 0.3300 - accuracy: 0.8537
8/20	7s 262ms/step	loss: 0.3555 - accuracy: 0.8433
9/20	7s 268ms/step	loss: 0.3375 - accuracy: 0.9247
10/20	7s 268ms/step	loss: 0.1861 - accuracy: 0.9125
11/20	7s 271ms/step	loss: 0.0917 - accuracy: 0.9780
12/20	7s 268ms/step	loss: 0.1073 - accuracy: 0.9481
13/20	7s 274ms/step	loss: 0.1600 - accuracy: 0.9304
14/20	7s 269ms/step	loss: 0.0909 - accuracy: 0.9829
15/20	7s 268ms/step	loss: 0.0941 - accuracy: 0.9705
16/20	7s 268ms/step	loss: 0.2872 - accuracy: 0.9278
17/20	7s 264ms/step	loss: 0.2979 - accuracy: 0.8411
18/20	7s 272ms/step	loss: 0.1101 - accuracy: 0.9493
19/20	7s 264ms/step	loss: 0.0962 - accuracy: 0.9527
20/20	7s 273ms/step	loss: 0.0945 - accuracy: 0.9768

Tabel 3. Loss dan Akurasi Model dengan *Hyperparameter*

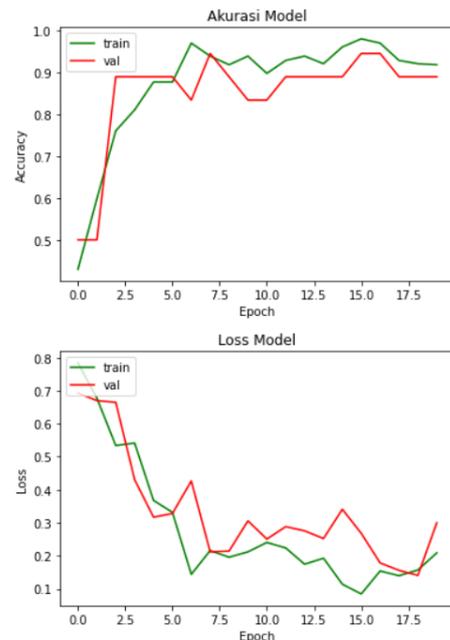
Epoch	Waktu	Loss dan Akurasi
1/20	3s 118ms/step	loss: 0.7041 - accuracy: 0.5464
2/20	3s 107ms/step	loss: 0.5835 - accuracy: 0.6600
3/20	3s 106ms/step	loss: 0.3141 - accuracy: 0.8800
4/20	3s 103ms/step	loss: 0.2544 - accuracy: 0.9072
5/20	3s 101ms/step	loss: 0.3095 - accuracy: 0.8660
6/20	3s 103ms/step	loss: 0.2388 - accuracy: 0.9200
7/20	3s 107ms/step	loss: 0.1178 - accuracy: 0.9800
8/20	3s 107ms/step	loss: 0.3476 - accuracy: 0.8866
9/20	3s 106ms/step	loss: 0.2837 - accuracy: 0.8763
10/20	3s 103ms/step	loss: 0.1668 - accuracy: 0.9400
11/20	3s 108ms/step	loss: 0.1986 - accuracy: 0.9278
12/20	3s 104ms/step	loss: 0.1147 - accuracy: 0.9691
13/20	3s 101ms/step	loss: 0.5159 - accuracy: 0.7938
14/20	3s 101ms/step	loss: 0.3673 - accuracy: 0.8866
15/20	3s 107ms/step	loss: 0.2004 - accuracy: 0.9300
16/20	3s 105ms/step	loss: 0.2506 - accuracy: 0.9278

17/20	3s 106ms/step	loss: 0.2640 - accuracy: 0.9200
18/20	3s 102ms/step	loss: 0.1596 - accuracy: 0.9500
19/20	3s 104ms/step	loss: 0.1824 - accuracy: 0.9278
20/20	3s 105ms/step	loss: 0.1593 - accuracy: 0.9400

Berdasarkan Tabel 2 dan Tabel 3 dapat diketahui bahwa dengan penambahan *hyperparameter* dapat mempercepat waktu *training* sampai lebih dari 50%. Kemudian untuk visualisasi grafik dari masing-masing nilai *loss* dan akurasi dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Visualisasi nilai akurasi dan loss model tanpa menggunakan *hyperparameter*



Gambar 6. Visualisasi nilai akurasi dan loss model dengan menggunakan *hyperparameter*

Berdasarkan grafik pada Gambar 5 diketahui bahwa tanpa menggunakan *hyperparameter* maka nilai *loss* dan akurasi tidak konsisten naik atau turun. Hal ini berakibat terjadinya *overfitting* yaitu keadaan dimana data yang digunakan untuk pelatihan itu adalah yang "terbaik", namun jika dilakukan tes dengan menggunakan data yang berbeda dapat mengurangi akurasi. Sedangkan Gambar 6 grafik yang dihasilkan menunjukkan hasil yang konsisten, walaupun naik turun namun tidak signifikan. Sehingga dapat disimpulkan penggunaan *hyperparameter* dapat membuat model menjadi *goodfit*.

3.2. Model Sebelum dilakukan Optimasi

Pada saat pelatihan menggunakan arsitektur CNN dengan tiga layer konvolusi, menggunakan *max pooling* dan *flatten*. Hasilnya pada arsitektur parameter yang digunakan lebih dari 3 juta parameter. Lebih detailnya dapat dilihat pada Tabel 4.

Tabel 4. Arsitektur Model Tanpa *Hyperparameter*

Layer (Type)	Output Shape	Param#
conv2d_40 (Conv2D)	(None,148,148, 32)	896
max_pooling2d_40(MaxPooling g)	(None,74,74, 32)	0
conv2d_1 (Conv2D)	(None,72, 72, 64)	18496
max_pooling2d_1(MaxPooling)	(None,36, 36, 64)	0
conv2d_2 (Conv2D)	(None,34,34,128)	73856
max_pooling2d_2(MaxPooling)	(None,17,17, 128)	0
conv2d_3 (Conv2D)	(None,15,15, 128)	147584
Max_pooling2d_3(MaxPooling)	(None,7,7,128)	0
Flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776
dense_1 (Dense)	(None, 3)	519

Kemudian dilakukan pengujian terhadap citra uji, hasilnya salah dalam mengidentifikasi citra sampah. Hal ini ditunjukkan pada Gambar 7, dimana model salah menebak citra anorganik. Secara keseluruhan dilakukan pengujian menggunakan citra uji yang hasilnya ditunjukkan pada Tabel 5.



Gambar 7. Deteksi Sampah Menggunakan Sebelum dilakukan Optimasi Model

Tabel 5. Pengujian Model Sebelum di Optimasi

Sampah	Prediksi		Total
	Benar	Salah	
Organik	14	3	17
Anorganik	9	8	17
Total	23	11	34
Akurasi			67,6%

Menggunakan data test yang dipakai yaitu sebanyak 34 citra. Diperoleh nilai akurasi model dengan membandingkan citra yang diprediksi benar dengan total citra sebesar 67,6%. Nilai tersebut tentunya masih belum akurat dalam memprediksi citra. Oleh karena itu perlu dilakukan optimasi agar hasil akurasi lebih besar.

3.3. Model Setelah dilakukan Optimasi

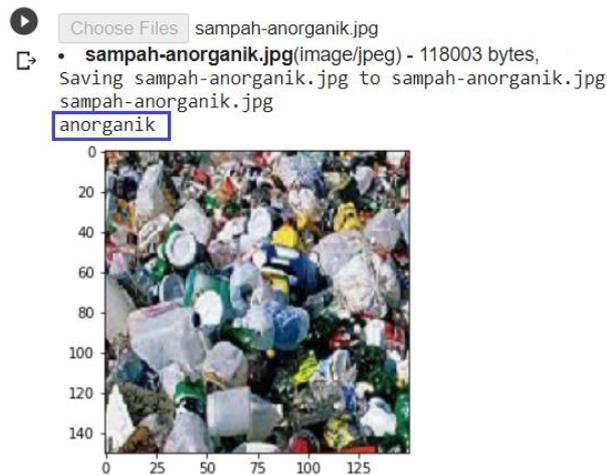
Dikarenakan model belum optimasi dalam memprediksi sampah, maka dilakukan optimasi dengan cara menambah *hyperparameter* pada saat pelatihan. *Hyperparameter* yang ditambahkan adalah *dropout*, *padding*, dan *strides*. Penambahan *hyperparameter* *dropout* mengurangi jumlah parameter pelatihan seperti ditunjukkan pada Tabel 6. Hal ini dikarenakan *dropout* menghapus *hidden* layer yang tidak terpakai pada saat pelatihan. Nilai *dropout* yang diberikan pada saat pelatihan adalah sebesar 20%.

Tabel 6. Arsitektur Model Menggunakan *Hyperparameter*

Layer (Type)	Output Shape	Param#
conv2d_40 (Conv2D)	(None,75,75, 32)	896
max_pooling2d_40(MaxPooling)	(None,37,37, 32)	0
dropout (Dropout)	(None,37, 37, 32)	0
conv2d_35 (Conv2D)	(None,35, 35, 64)	18496
max_pooling2d_41(MaxPooling)	(None,17, 17, 64)	0
dropout_41 (Dropout)	(None,17, 17, 64)	0
conv2d_42 (Conv2D)	(None,15,15,128)	73856
max_pooling2d_42(MaxPooling)	(None, 7, 7, 128)	0
dropout_37 (Dropout)	(None, 7, 7, 128)	0
conv2d_43 (Conv2D)	(None, 5, 5, 128)	147584
max_pooling2d_43(MaxPooling)	(None, 2, 2, 128)	0
dropout_38 (Dropout)	(None, 2, 2, 128)	0
flatten_10 (Flatten)	(None, 512)	0
dense_20 (Dense)	(None, 512)	262656
dense_21 (Dense)	(None, 3)	519

Dikarenakan jumlah parameter yang dipakai lebih sedikit, menyebabkan waktu pelatihan model di Tabel 6 menjadi lebih cepat. Penambahan *dropout* juga menyebabkan model yang dihasilkan tidak terjadi *overfitting*. Selain itu juga diberikan *hyperparameter* *padding* dan *stride* untuk meningkatkan akurasi model. *Padding* yang diberikan adalah 'same' dan dimensi *stride* yang di pakai adalah 2x2 piksel.

Kemudian setelah didapatkan model, selanjutnya dilakukan pengujian pendeteksian sampah dengan mengambil sampel sampah anorganik yang hasilnya ditunjukkan pada Gambar 8.



Gambar 8. Deteksi Sampah Menggunakan Model dengan *Hyperparameter*

Berdasarkan Gambar 8 diketahui bahwa model dengan *hyperparameter* dihasilkan mampu untuk mendeteksi sampah anorganik dengan benar. Setelah dilakukan menggunakan seluruh citra uji yang berjumlah 34 citra. Diperoleh nilai akurasi dari model adalah sebesar 91,2% seperti ditunjukkan pada Tabel 7.

Tabel 7. Pengujian Model Setelah di Optimasi

Sampah	Prediksi		Total
	Benar	Salah	
Organik	15	2	17
Anorganik	26	1	17
Total	31	3	34
Akurasi			91,2%

4. Kesimpulan

Convolutional Neural Network (CNN) merupakan metode kecerdasan buatan yang dapat digunakan untuk mengidentifikasi objek dari suatu citra berdasarkan polanya. Namun tidak semua citra dapat diidentifikasi secara akurat menggunakan CNN. Contohnya citra yang berisi objek berupa sampah organik dan anorganik yang memiliki banyak pola seperti ukuran, bentuk, dan warna. Menggunakan arsitektur CNN yang biasa tidak akurat dalam menentukan sampah organik dan anorganik. Oleh karena itu perlu dilakukan optimasi pada arsitektur CNN untuk mendapatkan hasil akurasi yang akurat. Teknik optimasi yang dilakukan pada penelitian ini adalah dengan menambahkan *hyperparameter* pada arsitektur CNN. *Hyperparameter* yang digunakan adalah *dropout*,

padding, dan *stride*. *Dropout* digunakan untuk meningkatkan akurasi dan menghindari *overfitting*, sedangkan *padding* dan *stride* digunakan untuk mempercepat proses pelatihan. Hasil dari optimasi menunjukkan kenaikan tingkat akurasi model sebesar 91,2%, dimana sebelum dilakukan optimasi nilai akurasi model sebesar 67,6%.

Ucapan Terimakasih

Terima kasih kepada Direktorat Riset dan Pengabdian Masyarakat (DRPM) Kementerian Riset dan Teknologi/Badan Riset dan Inovasi Nasional yang telah memberikan Skema Penelitian Dosen Pemula (PDP) Tahun Anggaran 2020. Ucapan terima kasih pula kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) IT Telkom Purwokerto yang telah mendukung penelitian ini.

Daftar Rujukan

- [1] D. Hoornweg, P. Bhada-Tata, and C. Kennedy, "Environment: Waste production must peak this century.," *Nature*, vol. 502, no. 7473, pp. 615–617, Oct. 2013, doi: 10.1038/502615a.
- [2] K. Fatmawati, E. Sabna, and Y. Irawan, "Rancang Bangun Tempat Sampah Pintar Menggunakan Sensor Jarak Berbasis Mikrokontroler Arduino," *Riau J. Comput. Sci.*, vol. 6, no. 2, pp. 124–134, 2020.
- [3] C. Chen, A. J. Barnett, and J. Su, "This Looks Like That : Deep Learning for Interpretable Image Recognition," no. NeurIPS, pp. 1–12, 2019.
- [4] H. Yanagisawa, T. Yamashita, and H. Watanabe, "A study on object detection method from manga images using CNN," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, Jan. 2018, pp. 1–4, doi: 10.1109/IWAIT.2018.8369633.
- [5] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, 2018, doi: 10.1109/TPAMI.2017.2711011.
- [6] J. Koushik, "Understanding Convolutional Neural Networks," 2016, [Online]. Available: <http://arxiv.org/abs/1605.09081>.
- [7] *Using Convolutional Neural Networks for Image Recognition.*
- [8] R. Sultana, R. D. Adams, Y. Yan, P. M. Yanik, and M. L. Tanaka, "Trash and Recycled Material Identification using Convolutional Neural Networks (CNN)," in *2020 SoutheastCon*, 2020, pp. 1–8, doi: 10.1109/SoutheastCon44009.2020.9249739.
- [9] M. Yang and G. Thung, "Classification of Trash for Recyclability Status," pp. 1–6.
- [10] Y. Kim, "Convolutional Neural Networks for Sentence Classification," pp. 1746–1751, 2014.
- [11] D. Montserrat, Q. Lin, J. Allebach, and E. Delp, "Training Object Detection And Recognition CNN Models Using Data Augmentation," *Electron. imaging*, vol. 2017, pp. 27–36, 2017