

Penerapan *Multi-Label Image Classification* Menggunakan Metode *Convolutional Neural Network (CNN)* Untuk Sortir Botol Minuman

Inggis Kurnia Trisiawan^{1*}, Yuliza², Fina Supegina², Said Attamimi²

¹PT. Control Systems Arena Para Nusa, Jakarta

²Teknik Elektro, Universitas Mercu Buana, Jakarta

*Inggiskurnia32@gmail.com

Abstrak—Dalam industri *Fast Moving Consumer Goods (FMCG)* seperti air minum kemasan, *quality control* berperan penting untuk menjaga produk yang baik, oleh karena itu diperlukan metode yang cepat dan andal untuk menganalisis data lapangan dan memberikan informasi yang dapat membantu dalam menentukan kualitas produk. Selama proses produksi di pabrik, botol air minum sering kali memiliki beberapa cacat misalnya, tutup dan label yang tidak ada atau tidak terpasang dengan benar, serta air yang kurang terisi pada botol. Semua masalah tersebut dapat menurunkan kualitas produk yang dikirim ke konsumen. Untuk mengatasinya, inspeksi visual menjadi teknik wajib di lini produksi. Metode inspeksi visual tradisional sering kali memerlukan beberapa kamera dan perangkat untuk mendeteksi beberapa cacat pada produk, masing-masing digunakan untuk mendeteksi jenis masalah yang berbeda. Dalam penelitian ini, digunakan metode *Multi-Label Image Classification* dengan *Convolutional Neural Network (CNN)* sebagai algoritma untuk mendeteksi beberapa cacat pada botol air sekaligus. Terdapat enam class label yaitu isi kurang, isi kosong, tutup rusak, tutup tidak ada, label rusak, dan label tidak ada. Masing-masing class label tersebut mewakili kemungkinan adanya cacat yang ada pada botol. Beberapa kombinasi *feature learning layer* dan *fully connected layer* digunakan untuk mengekstrak pola dan mengklasifikasikan gambar masukan. Untuk memfasilitasi penelitian ini, dikumpulkan dataset gambar botol air dengan skala besar, dimana didalam dataset tersebut merepresentasikan enam class label yang telah ditentukan. Saat diuji dengan dataset baru, model CNN mendapatkan hasil akurasi prediksi 98,526%, dan mendapat rata-rata akurasi sebesar 97,71% ketika diuji dengan *10-fold cross validation*.

Kata Kunci— *Artificial Intelligence (AI)*, *Multi-Label Image Classification*, *Convolutional Neural Network (CNN)*, *Sortir botol minuman*

DOI: 10.22441/jte.2022.v13i1.009

I. PENDAHULUAN

Air minum merupakan salah satu kebutuhan utama bagi manusia. Belakangan ini, telah berkembang tren dimana air minum ditempatkan ke dalam botol plastik terutama untuk air minum kemasan yang diproduksi secara massal. Didalam proses produksi air minum kemasan tersebut, tentunya tidak lepas dari adanya kesalahan dalam proses produksinya, contohnya adalah botol yang tidak ada label, kurang sesuai proses pemasangan label, tidak ada tutup, kurang sesuai pemasangan tutup dan botol yang tidak ada atau kurang dalam pengisian air minumannya.

Untuk menjaga kualitas produksi, harus dilakukan sortir terhadap produk yang telah jadi, agar jadi yang didapatkan sesuai standar dan layak untuk didistribusikan ke konsumen. [1]

Salah satu cara sortir produk jadi adalah dengan cara visual. Dengan sortir secara visual, kualitas produk dapat diketahui berdasarkan tampilan luarnya, apakah sudah sesuai standar atau mungkin ada beberapa bagian proses yang terlewat sehingga menyebabkan produk tersebut menjadi tidak layak. Saat ini telah berkembang beberapa metode yang dapat digunakan untuk pengecekan barang secara visual, antara lain dengan memanfaatkan *Convolutional Neural Network (CNN)*. [2]

Pada penelitian ini, fokus utamanya adalah penerapan CNN yang digunakan untuk klasifikasi gambar digital dari kamera, dimana gambar yang akan dianalisa merupakan sebuah botol minuman. Disini, CNN akan menganalisa dan memprediksi dari sebuah gambar botol minuman, apakah botol minuman tersebut terdapat kerusakan atau tidak.

II. PENELITIAN TERKAIT

Penelitian sebelumnya telah dilakukan oleh Valentina et al [1], hasilnya adalah ELU merupakan activation function yang paling sesuai dengan empat layer *max pooling*, tiga *fully connected layer* dan empat *layer konvolusi*. Dengan learning rate 0.00001, *dropout* 0.8 dan *epoch*-nya 50 kali dengan hasil akurasi sekitar 80%. Berikutnya Nugroho et al [2], membuat sistem yang dapat mengenali dan mengklasifikasi label merek pada botol plastik dengan 5 kelas berukuran 600ml. hasil akurasinya adalah 87,12%.

A. *Multi Label Image Classification*

Multi label image classification adalah salah satu jenis dari classification dimana dalam satu input gambar digital dapat diprediksi beberapa kemungkinan label yang paling mendekati. [3][4][5] *Multi label image classification* tentunya akan sangat relevan jika diterapkan pada pengolahan gambar digital di kehidupan sehari-hari, karena di dalam satu gambar biasanya tidak hanya terdapat satu label saja tapi bisa lebih dari dua label sekaligus. [6][7]

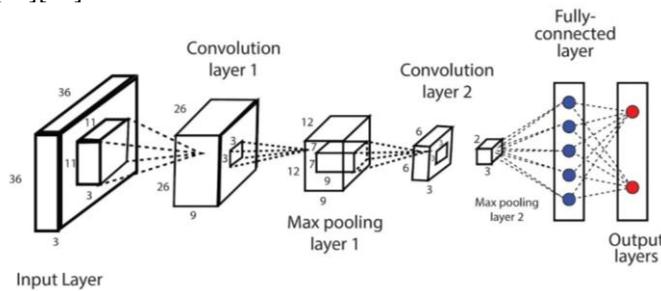
B. *Deep learning*

Deep learning merupakan pengembangan dari *neural network* yang memiliki jumlah layer yang banyak untuk memberikan ketepatan tugas seperti deteksi objek, pengenalan

suara, terjemahan bahasa dan lain lain. *Deep learning* memungkinkan model komputasi yang terdiri dari beberapa layer pengolahan untuk mempelajari representasi data dengan berbagai tingkat abstraksi. [8][9] Metode ini menemukan struktur yang rumit di dalam set data yang besar dengan menggunakan algoritma backpropagation untuk menunjukkan bagaimana mesin harus mengubah parameter internal yang digunakan untuk menghitung perwakilan di setiap layer dari perwakilan di layer sebelumnya. [10][11]

C. *Convolutional Neural Network (CNN)*

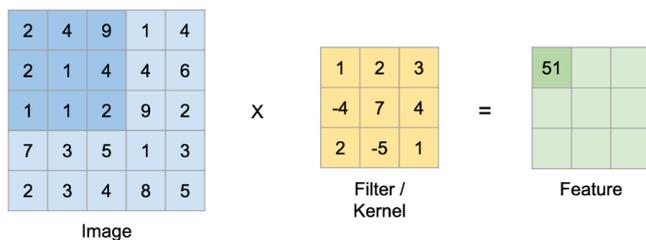
CNN adalah satu dari beberapa macam model yang ada pada *deep learning*. Struktur CNN terdiri dari *feature learning layer* dan *fully connected layer*. *Feature learning layer* dalam CNN terdiri dari beberapa layer, yaitu layer konvolusi, fungsi aktivasi (ReLU), *max pooling* dan *dropout regularization*. CNN bekerja secara hierarki, sehingga output pada layer konvolusi pertama digunakan sebagai input pada layer konvolusi selanjutnya. [12][13]



Gambar 1. *Convolutional Neural Network (CNN)*

D. *Layer Kovolusi*

Layer kovolusi merupakan layer pertama dalam arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan gambar, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada gambar di semua bagian yang memungkinkan seperti yang ditunjukkan pada Gambar 2 Kotak biru secara keseluruhan adalah gambar input. Kernel bergerak dari sudut kiri atas ke kanan bawah sehingga hasil konvolusi dari gambar tersebut berupa feature map (kotak hijau). [14]



Gambar 2. Operasi Konvolusi

$$G[m, n] = \sum_j \sum_k h[j, k] f[m - j, n - j] \dots \dots \dots (1)$$

Dimana:

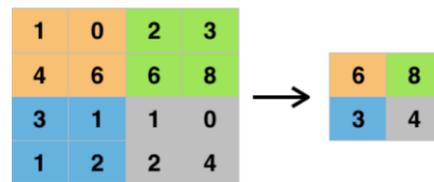
f = input

h = kernel

m = baris pada *feature map*
 n = kolom pada *feature map*

E. *Max Pooling*

Max pooling merupakan operasi yang membagi output dari layer konvolusi menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks gambar yang telah direduksi seperti pada Gambar 3, grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek gambar mengalami translasi (pergeseran).



Gambar 3. *Max Pooling*

F. *Rectified Linear Unit (ReLU)*

ReLU adalah salah satu jenis fungsi aktivasi yang bersifat non-linear. Nilai output dari neuron bisa dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input adalah positif, maka output dari neuron adalah nilai input aktivasi itu sendiri. Adapun persamaan untuk adalah sebagai berikut.

$$f(x) = \max(0, x) \dots \dots \dots (2)$$

G. *Dropout Regularization*

Dropout merupakan sebuah teknik regulasi jaringan syaraf tiruan dengan tujuan memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain neuron-neuron tersebut dinonaktifkan secara acak.

H. *Fully Connected Layer*

Fully connected layer merupakan layer dimana semua neuron aktivasi dari layer sebelumnya terhubung semua dengan neuron di layer selanjutnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari layer ini adalah untuk menyatukan semua node menjadi satu dimensi. [15]

I. *Binary Cross-Entropy*

Binary Cross-Entropy merupakan salah satu jenis *loss function* yang dipergunakan untuk mengetahui nilai *loss* dari sebuah model pengujian klasifikasi, terutama untuk klasifikasi biner. *Binary Cross-Entropy* memberikan nilai *loss* untuk setiap probabilitas klasifikasi binernya. Pada *Binary Cross-Entropy*, komponen vektor dari output berdiri sendiri dan tidak saling terikat satu sama lain, artinya *loss* yang dihitung untuk setiap komponen vektor keluaran CNN tidak dipengaruhi oleh komponen vektor lainnya. Maka dari itu *Binary Cross-Entropy* digunakan untuk klasifikasi *multi-label*, dimana nilai dari *class*

tertentu tidak mempengaruhi class lainnya. Adapun persamaannya adalah sebagai berikut:

$$Loss = \frac{1}{N} \sum_{i=1}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \dots (3)$$

Dimana:

\hat{y}_i = output neuron

y_i = target

J. Sigmoid

Sigmoid merupakan salah satu fungsi aktivasi non-linear dimana pada fungsi aktivasi ini akan menerima input berupa angka tunggal dan mengubah nilai tersebut menjadi sebuah nilai yang memiliki range mulai dari 0 sampai 1. Adapun persamaan untuk *sigmoid* adalah sebagai berikut

$$f(x_i) = \frac{1}{1+e^{-x_i}} \dots (4)$$

Dimana:

x_i = nilai input

III. METODOLOGI PENELITIAN

A. Metode Analisis Data

Digunakan bahasa pemrograman Python dengan menggunakan Google Colab sebagai text editor dan engine untuk menjalankan programnya. Google Colab merupakan sebuah layanan dimana berisi beberapa tools yang dapat dipergunakan untuk pemrograman Python berbasis cloud, artinya penyimpanan data, pemrosesan data dan lain-lain tersimpan di dalam server sehingga pengguna tidak perlu menginstal dan tinggal memakainya saja. Untuk penelitian ini digunakan framework TensorFlow dan library Keras untuk pembuatan arsitektur model CNN-nya.

B. Penentuan Class Label

Class label merupakan objek yang ingin dipresiksi di dalam sebuah model CNN. Penentuan class label dalam sebuah model CNN sangatlah penting karena akan mempengaruhi hasil prediksi dan proses pelatihan jaringan CNN. Berikut merupakan *class label* yang digunakan pada penelitian ini.

Tabel 1. *Class Label*

Class Label	Keterangan
Isi kurang	Isi dalam botol kurang dari 13 cm
Isi tidak ada	Tidak adanya isi di dalam botol
Tutup rusak	Tutup yang dipasang kurang kencang atau miring
Tutup tidak ada	Tidak adanya tutup di botol
Label rusak	Label yang dipasang miring atau tidak sesuai
Label tidak ada	Tidak adanya label di botol

C. Perancangan Dataset

Dataset yang digunakan dalam penelitian ini berupa data gambar. Di dalam metode CNN perlu digunakan dataset yang banyak untuk menghasilkan prediksi model yang baik. Dataset yang dipergunakan dalam penelitian ini berupa beberapa buah

gambar yang merupakan hasil tangkapan kamera terhadap botol minuman. Dalam penelitian ini pengambilan dataset dibagi berdasarkan setiap kemungkinan adanya class label didalam gambar, dimana secara garis besar dibagi menjadi tiga yaitu adanya satu class label terdeteksi, dua class label terdeteksi dan tiga class label terdeteksi. Untuk menghindari *imbalance dataset* yang dapat mengakibatkan prediksi model menjadi tidak sesuai, penulis menyamaratakan jumlah dataset gambar, yaitu dengan 100 buah gambar pada masing masing-kemungkinan.

Dalam penelitian ini ukuran aspek rasio saat pengambilan gambar dataset adalah 3 x 4 dan memanfaatkan background hijau untuk mengurangi *noise* pada gambar dataset. Untuk mengoptimalkan hasil model prediksi, setiap dataset gambar bersifat unik dan berbeda antara satu dengan yang lain, dimana proses pengambilan gambar dilakukan dengan memperhatikan beberapa kemungkinan posisi ataupun letak *class label* didalam sebuah gambar.



Gambar 4. Contoh Dataset

Tabel 2. Sebaran Dataset

No	Class Label Pada Gambar	Jumlah Gambar
1	Semua oke	100
2	Isi kurang	100
3	Isi tidak ada	100
4	Tutup rusak	100
5	Tutup tidak ada	100
6	Label rusak	100
7	Label tidak ada	100
8	Isi kurang, Tutup rusak	100
9	Isi kurang, Tutup tidak ada	100
10	Isi kurang, Label rusak	100
11	Isi kurang, Label tidak ada	100
12	Isi tidak ada, Tutup rusak	100
13	Isi tidak ada, Tutup tidak ada	100
14	Isi tidak ada, Label rusak	100
15	Isi tidak ada, Label tidak ada	100
16	Label rusak, Tutup rusak	100
17	Label rusak, Tutup tidak ada	100
18	Label tidak ada, Tutup rusak	100
19	Label tidak ada, Tutup tidak ada	100
20	Isi kurang, Tutup rusak, Label rusak	100
21	Isi kurang, Tutup rusak, Label tidak ada	100
22	Isi kurang, Tutup tidak ada, Label rusak	100
23	Isi kurang, Tutup tidak ada, Label tidak ada	100
24	Isi tidak ada, Tutup rusak, Label rusak	100
25	Isi tidak ada, Tutup rusak, Label tidak ada	100
26	Isi tidak ada, Tutup tidak ada, Label rusak	100
27	Isi tidak ada, Tutup tidak ada, Label tidak ada	100
Total Keseluruhan		2700

D. Perancangan One Hot Encoding

One-hot encoding merupakan sebuah proses untuk mengkategorikan sebuah class label yang ada pada neural network kedalam bilangan biner. Angka 1 merepresentasikan bahwa class label tersebut ada, dan angka 0 merepresentasikan bahwa class label tersebut tidak ada. Hasil pengumpulan dataset gambar sebelumnya akan di mapping ke dalam sebuah file dengan format CSV dimana dalam file tersebut termuat nama file gambar, label yang ada (isi kurang, isi tidak ada, tutup rusak, tutup tidak ada, label rusak, label tidak ada) beserta keterangan ada tidaknya class label tersebut pada masing-masing gambar. Dataset yang sudah dibuat kemudian dimasukan ke dalam Google Colab untuk diolah datanya ke dalam program CNN.

E. Pengolahan Dataset

Dataset yang digunakan dalam program ini terbagi menjadi dua jenis yaitu file .csv yang berisi dataset one-hot encoder dimana didalamnya terdapat nama gambar dan keterangan class label yang ada pada gambar serta file .jpg yang berisi data gambar dari dataset itu sendiri.

Untuk dataset gambar, sebelum masuk ke jaringan CNN akan diolah terlebih dahulu. Pertama gambar tersebut akan dirubah ukurannya agar seragam dengan resolusi 200 x 200 piksel dengan tiga buah channel RGB, kemudian gambar akan dirubah ke dalam bentuk array yang merepresentasikan nilai dari setiap piksel yang ada pada gambar tersebut. Setelah itu nilai dari array tersebut akan dirubah range nilainya dari yang sebelumnya 0-255 menjadi 0-1 agar nilai tidak terlalu besar dan mempermudah pada saat proses pelatihan model CNN.

F. Perancangan Model CNN

Penentuan model arsitektur dari CNN merupakan hal yang sangat penting untuk dilakukan. Baik tidaknya hasil prediksi output akan sangat bergantung dari arsitektur yang dipakai. Suatu model dikatakan baik jika tidak *overfitted* ataupun *underfitted*. *Overfitted* artinya model menghafal semua data pelatihan, tanpa menganalisis pola trend dari data. Sedangkan *underfitted* artinya model tidak dapat menganalisis data pelatihan secara keseluruhan sehingga menghasilkan prediksi yang kurang sesuai. Model CNN yang baik mampu mengeneraliskan data inputan, artinya model tersebut dapat memprediksi output dari dataset yang berbeda dengan dataset yang sudah dipakai untuk proses pelatihan.

Saat ini masih belum ada paduan yang baku tentang bagaimana menentukan model arsitektur yang optimal karena penentuan model arsitektur tersebut sangat bergantung pada banyak hal, salah satunya jumlah output prediksi, ukuran dataset, jenis dataset dll. Di dalam CNN terdapat beberapa *hyperparameter* contohnya adalah jumlah *epoch*, filter, ukuran kernel, ukuran *batch*, jumlah *feature learning layer*, jumlah *neuron fully connected layer* dll, sehingga ada kemungkinan tak hingga untuk setiap kombinasi *hyperparameter*-nya. Untuk menentukan model arsitektur yang optimal, perlu dilakukannya percobaan pada setiap kombinasi arsitektur tersebut. Dalam penelitian ini, model arsitektur yang dipakai adalah sebagai berikut:

Tabel 3. Model Arsitektur CNN

Layer	Deskripsi
1	Conv2D with 16 filters, kernel size (5,5), activation = ReLu
	Batch Normalization
	Max Pooling with pool size (2,2)
	Dropout regularization (0,2)
2	Conv2D with 32 filters, kernel size (5,5), activation = ReLu
	Batch Normalization
	Max Pooling with pool size (2,2)
	Dropout regularization (0,2)
3	Conv2D with 64 filters, kernel size (5,5), activation = ReLu
	Batch Normalization
	Max Pooling with pool size (2,2)
	Dropout regularization (0,2)
4	Conv2D with 128 filters, kernel size (5,5), activation = ReLu
	Batch Normalization
	Max Pooling with pool size (2,2)
	Dropout regularization (0,2)
5	Conv2D with 300 filters, kernel size (5,5), activation = ReLu
	Batch Normalization
	Max Pooling with pool size (2,2)
	Dropout regularization (0,2)
6	Dense layer with 128 neuron, activation = ReLu
	Dropout regularization (0,5)
7	Dense layer with 64 neuron, activation = ReLu
	Dropout regularization (0,5)
8	Dense layer with 32 neuron, activation = ReLu
	Dropout regularization (0,5)
9	Dense layer with 6 neuron, activation = Sigmoid
	Dropout regularization (0,5)

G. Pengujian Dengan K-Fold Cross Validation

K-fold cross validation digunakan untuk mengevaluasi kinerja model dengan menggunakan dataset yang sudah dipersiapkan. Metode ini akan membagi data latih menjadi k sub data. Tiap-tiap sub data akan menjadi data validasi dan sisanya akan menjadi data latih. Proses ini akan terus diulangi hingga semua k sub data telah menjadi data validasi.

H. Pengujian Dengan Dataset Baru

Pengujian dengan dataset baru dilakukan dengan tujuan untuk mengetahui performa model jika diuji dengan dataset yang bukan berasal dari dataset untuk proses pelatihan apakah sudah dapat memprediksi dengan bagus, atau model CNN yang hanya menghafal data yang sudah ada saja.

IV. HASIL DAN ANALISA

Pada penelitian ini dilakukan tiga tahap untuk implementasi dari model CNN. Pertama adalah pelatihan dan validasi. Kedua pengujian dengan metode *K-fold cross validation* untuk mengetahui nilai akurasi rata-rata pada setiap dataset untuk pelatihan dan validasi. Ketiga, pengujian dengan dataset baru, dimana digunakan dataset yang berbeda dengan dataset untuk pelatihan dan validasi.

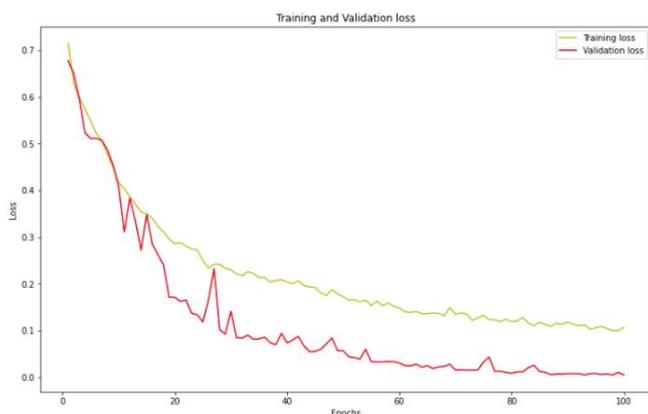
A. Pelatihan dan Validasi

Pada penelitian ini, pelatihan model akan dihentikan jika sudah mencapai *epoch* sebanyak 100. Selain itu, ukuran *batch* yang dipakai untuk proses pelatihan adalah sebesar 90. Digunakan dataset sejumlah 2700 gambar, pembagiannya adalah 1890 dataset gambar untuk pelatihan dan 810 dataset gambar untuk proses validasi. Karena digunakan ukuran *batch* 90, maka total *batch* yang ada untuk dataset pelatihan adalah sebanyak 21 *batch* dimana akan dilakukan *update* nilai *weight* pada masing masing *batch* tersebut. Proses pelatihan ini dilakukan dengan menggunakan *optimizer* Adam dan *initial learning rate* sebesar 0,001. Berikut merupakan nilai *loss* dan akurasi per 10 *epoch* untuk proses pelatihan dan validasi yang sudah dilakukan.

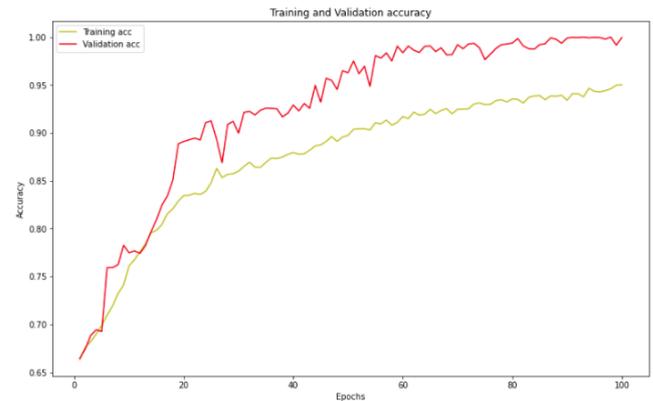
Tabel 4. Hasil Pelatihan Model

Epoch	Pelatihan		Validasi	
	Loss	Akurasi	Loss	Akurasi
1	0,7149	0,6632	0,6773	0,6644
10	0,4163	0,7612	0,4081	0,7748
20	0,2853	0,8346	0,1711	0,8911
30	0,2295	0,8600	0,1409	0,8998
40	0,2037	0,8795	0,0733	0,9291
50	0,1723	0,8974	0,0563	0,9627
60	0,1482	0,9170	0,0300	0,9837
70	0,1346	0,9247	0,0152	0,9921
80	0,1187	0,9355	0,0085	0,9938
90	0,1179	0,9341	0,0076	0,9990
100	0,1064	0,9502	0,0046	0,9995

Adapun grafik *loss* dan akurasi untuk pelatihan dan validasi dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Grafik *Loss* Pelatihan dan Validasi



Gambar 6. Grafik Akurasi Pelatihan dan Validasi

B. K-Fold Cross Validation

Pada pengujian untuk *cross validation* nilai parameter *k* yang ditentukan adalah sebesar 10, dimana dataset akan dilatih dan dilakukan validasi sebanyak 10 kali dengan menggunakan pembagian dataset yang berbeda tiap *fold*-nya. Tujuan dari *cross validation* adalah untuk mengetahui performa model dengan skema pembagian dataset untuk pelatihan dan validasi yang berbeda. Pelatihan untuk *cross validation* dilakukan dengan menggunakan ukuran *batch* 90 dan *epoch* sebesar 100. Adapun hasil dari *cross validation* yang sudah dilakukan dapat dilihat pada Tabel 5 berikut ini:

Tabel 5. Akurasi *Cross Validation*

Fold	Akurasi
1	100%
2	96,36%
3	94,51%
4	100%
5	99,20%
6	93,53%
7	97,10%
8	99,32%
9	97,16%
10	100%
Rata-rata	97,71%
SD	+2,25%

C. Pengujian Dengan Dataset Baru

Pada pengujian dengan dataset baru ini, penulis mempersiapkan total 260 gambar yang digunakan untuk melakukan pengujian. Dataset tersebut secara garis besar dibagi menjadi tiga bagian yaitu jika terdapat 1 *class label*, 2 *class label*, dan 3 *class label*, dimana untuk mengetahui performa model secara keseluruhan, jumlah gambar untuk masing masing kemungkinan adanya *class label* tersebut disamaratakan yaitu 10 gambar. Hasil dari pengujian dengan dataset baru dapat dilihat pada Tabel 6.

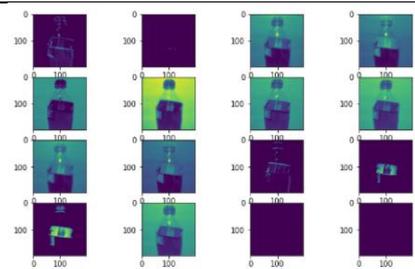
Tabel 6. Hasil Pengujian Dengan Dataset Baru

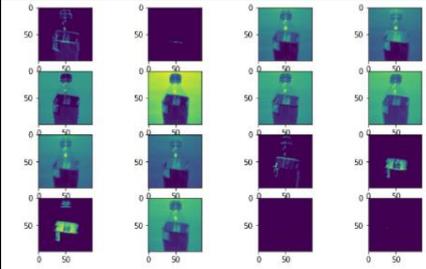
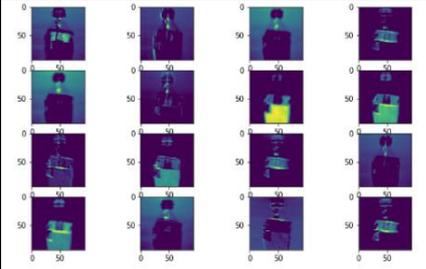
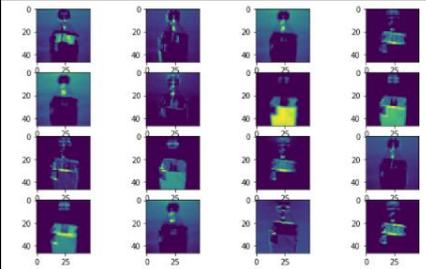
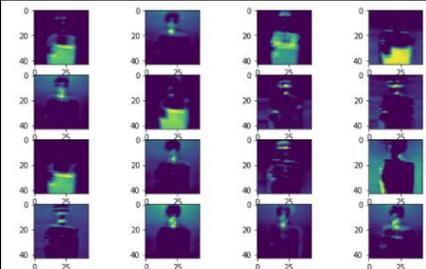
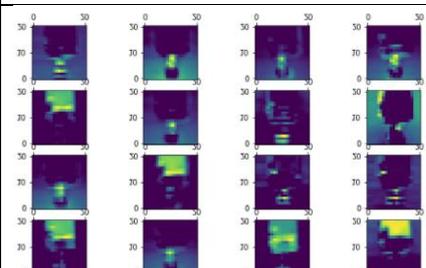
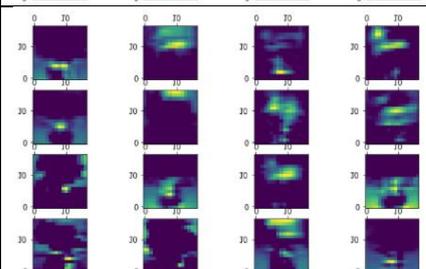
No	Class Label	Prediksi Benar	Prediksi Salah	Akurasi
1	Isi kurang	260	0	100%
2	Isi tidak ada	260	0	100%
3	Tutup rusak	255	5	98,077%
4	Tutup tidak ada	260	0	100%
5	Label rusak	242	18	93,077%
6	Label tidak ada	260	0	100%
Rata-rata				98,526%

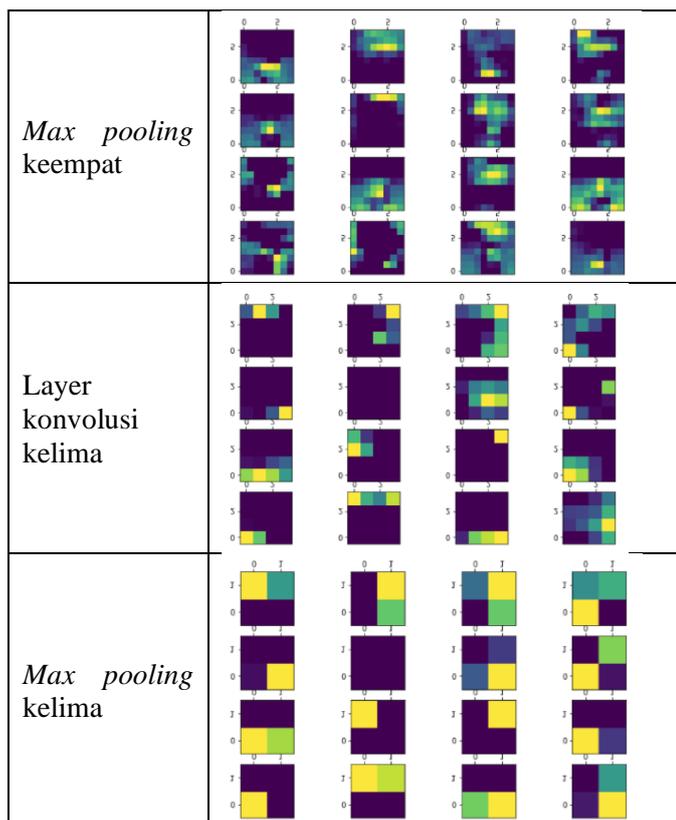
D. Visualisasi Hasil

Visualisasi proses training pada tiap layer dilakukan dengan cara memproyeksikan kembali fitur aktivasi ke awal proses training jaringan. Pada penelitian ini digunakan total 9 layer utama dimana dibagi menjadi dua, yaitu 5 layer untuk *feature learning layer* dan 4 layer untuk *fully connected layer*. Dalam *feature learning layer*, ukuran filter pada setiap layer konvolusi akan semakin membesar pada setiap layernya. Pada layer pertama bertujuan untuk menangkap beberapa informasi relevan dari data piksel mentah yang masih memiliki *noise*. Setelah hasil ekstraksi gambar mentah yang berguna telah diekstraksi, maka akan ditambahkan jumlah filter pada layer kedua, ketiga dan setelahnya untuk mendeteksi pola yang ada pada gambar dengan lebih detail. Operasi *max pooling* digunakan untuk mendapatkan hasil *feature* yang paling besar dan untuk mereduksi hasil daripada *feature map*. Contoh hasil ekstraksi pada layer konvolusi dan *max pooling* dapat dilihat pada Tabel 7.

Tabel 7. Hasil Ekstraksi *Feature Learning Layer*

Input Gambar	
Layer konvolusi pertama	

<i>Max pooling pertama</i>	
Layer konvolusi kedua	
<i>Max pooling kedua</i>	
Layer konvolusi ketiga	
<i>Max pooling ketiga</i>	
Layer konvolusi keempat	



V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dengan melalui beberapa tahap perencanaan, perancangan, pembuatan serta pengujian maka dapat diambil kesimpulan sebagai berikut:

1. Dalam penelitian ini, CNN berhasil untuk mengklasifikasikan gambar ke dalam enam buah *class label* yang sudah ditentukan yaitu “isi kurang”, “isi tidak ada”, “tutup rusak”, “tutup tidak ada”, “label rusak” dan “label tidak ada”.
2. Model arsitektur CNN yang dipakai dalam penelitian ini yaitu menggunakan 5 buah *feature learning layer* dan 4 *fully connected layer*, dimana dalam proses pelatihan, didapatkan nilai akurasi sebesar 95,02% dengan nilai loss-nya yaitu 0,1064.
3. Rata-rata akurasi prediksi yang didapatkan pada pengujian menggunakan metode *10-fold cross-validation* adalah 97,71% dan standar deviasi yang didapat yaitu 2,25%.
4. Rata-rata akurasi prediksi yang didapatkan pada pengujian dengan dataset baru yaitu 98,526%.

UCAPAN TERIMA KASIH

Pada bagian ini dituliskan ucapan terima kasih terhadap pihak-pihak yang membantu terselesaikannya penelitian ini

serta ucapan terima kasih terhadap tim editorial Jurnal Teknologi Elektro atas dipublikasikannya penelitian ini.

DAFTAR PUSTAKA

- [1] R. Valentina, S. Rostianingsih, and Alvin Nathaniel Tjondrowiguno, “Pengenalan Gambar Botol Plastik dan Kaleng Minuman Menggunakan Metode Convolutional Neural Network,” *Jurnal Infra*, vol. 8, no. 1, pp. 249–254, 2020, Accessed: Feb. 05, 2022. [Online]. Available: <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/9805>.
- [2] A. S. Nugroho, “Sistem Pengenalan Botol Plastik Berdasarkan Label Merek Menggunakan Faster-RCNN,” *Techno (Jurnal Fakultas Teknik, Universitas Muhammadiyah Purwokerto)*, vol. 21, no. 2, p. 111, Nov. 2020, doi: 10.30595/techno.v21i2.8635.
- [3] I. M. Baltruschat, “Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification,” *Scientific Reports*, vol. 9, p. 6381, 2019. <https://doi.org/0.1038/s41598-019-42294-8>
- [4] L. Gaobo, “Combining Convolutional Neural Network with Recursive Neural Network for Blood Cell Image Classification,” *IEEE Access*, vol. 6, pp. 36188–36197, 2018. <https://doi.org/10.1109/ACCESS.2018.2846685>
- [5] Y. Huang, “Multi-Task Deep Neural Network for Multi-Label Learning,” *IEEE International Conference on Image Processing*, pp. 2897–2900, 2013. <https://doi.org/10.1109/ICIP.2013.6738596>
- [6] A. L. Katole, “Hierarchical Deep Learning Architecture for 10K Objects Classification,” *Computer Science & Information Technology (CS &IT)*, p. 77–93, 2015. <https://doi.org/10.5121/csit.2015.51408>
- [7] I. Naoto, “Multi-Label Fashion Image Classification with Minimal Human Supervision,” *International Convergence on Computer Vision*, pp. 2261–2267, 2017. <https://doi.org/10.1109/ICCVW.2017.265>
- [8] Y. LeCun, B. Yoshua and H. Geoffrey, “Deep Learning,” *Nature*, vol. 521, pp. 436–444, 2015. <https://doi.org/10.1038/nature14539>
- [9] F. Hu, G. S. Xia, J. Hu and L. Zhang, “Transferring Deep Convolutional Neural Networks For The Scene Classification Of High-Resolution Remote Sensing Imagery,” *Remote Sensing*, vol. 7, no. 11, p. 14680–14707, 2015. <https://doi.org/10.3390/rs71114680>
- [10] F. Hu, G. S. Xia, J. Hu and L. Zhang, “Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery,” *Remote Sensing*, vol. 7, no. 11, p. 14680–14707, 2015. <https://doi.org/10.3390/rs71114680>
- [11] H. W. Mwangi and M. Mokoena, “Using Deep Learning to Detect Polyethylene Terephthalate (PET) Bottle Status for Recycling,” *Double Blind Peer Reviewed International Research Journal*, vol. 19, no. 4, 2019. <https://doi.org/10.34257/GJCSTGVOL19IS4PG27>
- [12] M. H. Romario, E. Ihsanto, and T. M. Kadarina, “Sistem Hitung dan Klasifikasi Objek dengan Metode Convolutional Neural Network,” *Jurnal Teknologi Elektro*, vol. 11, no. 2, p. 108, Jun. 2020, doi: 10.22441/jte.2020.v11i2.007.
- [13] M. Utami, Julpri Andika, and Said Attamimi, “Artificial Intelligence For Banana’s Ripeness Detection Using Conventional Neural Network Algorithm,” *Jurnal Teknologi Elektro*, vol. 12, no. 2, pp. 73–79, 2021, doi: 10.22441/jte.2021.v12i2.005.
- [14] J. Kim, O. Sangjun, Y. Kim and M. Lee, “Convolutional Neural Network with Biologically Inspired Retinal Structure,” *Procedia Computer Science*, vol. 88, pp. 145–154, 2016. <https://doi.org/10.1016/j.procs.2016.07.418>
- [15] S. Albelwi and A. Mahmood, “A Framework for Designing the Architectures of Deep Convolutional Neural Networks,” *Entropy*, vol. 19, 2017. <https://doi.org/10.3390/e19060242>