

# Implementasi Algoritma *Convolution Neural Network* pada Klasifikasi Limbah dengan Arsitektur *MobileNet*

Hery Oktafiandi<sup>1\*</sup>, Winarnie<sup>2</sup>

<sup>1,2</sup>Teknologi Rekayasa Perangkat Lunak, Politeknik Sawunggali Aji

<sup>2</sup>Magister Teknologi Informasi, Universitas Amikom Yogyakarta

<sup>1</sup>Jl. Wismoaji no.08 Kutoarjo, Purworejo, Jawa Tengah, Indonesia

<sup>2</sup>Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta, Indonesia

E-mail: heryokta@gmail.com<sup>1</sup>, winzahwa@gmail.com<sup>2</sup>

\*penulis korespondensi

Published: 30 Maret 2024

**Abstrak** - Proses daur ulang sampah bisa digunakan sebagai solusi dalam menangani masalah sampah secara umum. Proses daur ulang sampah membutuhkan proses pemilahan sampah berdasarkan jenis sampah. Limbah banyak memberikan dampak negatif pada lingkungan hidup. Dampak negatif yang diberikan bisa berupa pencemaran udara, air, dan tanah. Dengan menggunakan teknologi berbasis *deep learning* sampah bisa diklasifikasikan berdasarkan jenisnya. *Convolutional Neural Network* (CNN) bagian metode dari *deep learning* yang banyak digunakan dalam pengklasifikasian image. Pengklasifikasian limbah bisa membantu masyarakat dan pemerintah dalam menanggulangi dampak negatif limbah. Pada penelitian ini menggunakan arsitektur pada CNN yaitu MobileNet guna pengenalan citra limbah. Dengan penggunaan MobileNet, hasil akurasi yang diperoleh sebesar 93%.

**Kata kunci:** deep learning, CNN, MobileNet

**Abstract** - The waste recycling process can be used as a solution to dealing with waste problems in general. The waste recycling process requires a waste sorting process based on the type of waste. Waste has a lot of negative impacts on the environment. The negative impacts can be in the form of air, air and land pollution. By using deep learning-based technology, waste can be classified based on its type. Convolutional Neural Network (CNN) is a deep learning method that is widely used in image classification. Waste classification can help society and the government overcome the negative impacts of waste. This research uses a CNN architecture, namely MobileNet, for waste image recognition. By using MobileNet, the accuracy results obtained were 93%.

**Keywords:** deep learning, CNN, MobileNet

## 1. PENDAHULUAN

Limbah dan sampah merupakan permasalahan yang menjadi pekerjaan rumah bersama untuk menyelesaikannya. Diantara solusi yang dapat digunakan adalah dengan mendaur ulang sampah atau limbah tersebut. Limbah banyak memberikan dampak negatif pada lingkungan hidup. Dampak negatif yang diberikan bisa berupa pencemaran udara, air, dan tanah. Efek dari dari pencemaran adalah terganggunya kesehatan pada manusia [1]. Permasalahan diatas bisa diatasi dengan cara mengklasifikasikan limbah dan mengidentifikasi limbah berdasarkan jenisnya [2]. Pengklasifikasikan limbah bisa membantu masyarakat dan pemerintah dalam menanggulangi dampak negatif limbah. Perlu dilakukan pengklasifikasikan limbah secara efektif [3], dengan memanfaatkan teknologi machine learning untuk mengklasifikasikan dan mendeteksi limbah dibandingkan dengan cara manual yang sangat terbatas karena keterbatasan manusia yang melakukannya [4].

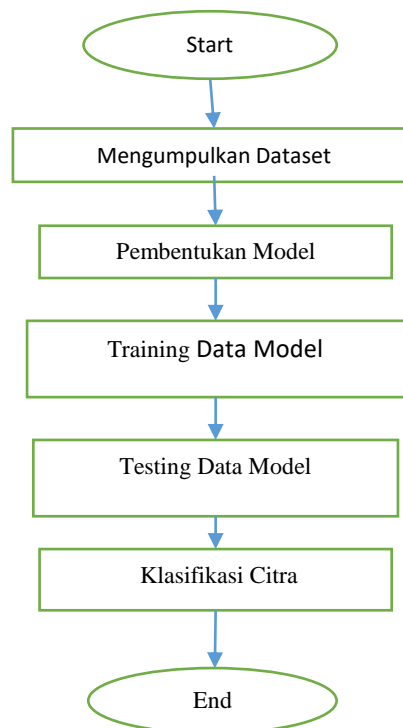
*Deep Learning* merupakan bagian dari *Artificial Intelligent* yang merupakan pengetahuan menggunakan beberapa lapisan representasi data dan ekstraksi fitur. Aplikasi pada machine learning dapat dibedakan berdasarkan aplikasi yang digunakan. Aplikasi dalam *machine learning* dapat digunakan pada pengenalan ucapan, visual, konversi ucapan ke teks, pengenalan citra, prediksi, pengelompokan, dan lain-lain [5][6][7]. *Convolutional Neural Network* (CNN) merupakan cara klasifikasi image yang terpopuler, dengan objek tersegmentasi [8], CNN digunakan untuk mendeteksi dan mengenali sebuah image yang telah dilatih sebelumnya [9]. CNN bekerja memanfaatkan proses konvolusi pada image menggunakan kernel konvolusi untuk mendapatkan hasil perkalian image dengan filter yang digunakan [10].

Penelitian sebelumnya dilakukan oleh Suang Liang. D.k.k [11] yang melakukan penelitian untuk mengoptimasikan identifikasi limbah dengan menggunakan algoritma CNN dan arsitektur MTLA (*Multy Task Learning Architecture*) mendapatkan nilai F1 sebesar 95%, *Confidence* 95%, dan *Precision* sebesar 81.5%. Penelitian lain mendeteksi sampah juga dilakukan oleh Cong Wang d.k.k [12] melakukan percobaan dengan menggunakan dataset sebanyak 6 class dan menggunakan algoritma CNN dengan arsitektur MobileNetV3, mendapatkan akurasi sebesar 94.26%. Zhuang Kang d.k.k [13] melakukan percobaan dengan CNN dan ResNet-34, menghasilkan nilai akurasi sebesar 99%. Uday Kulkarni d.k.k [14] meneliti deteksi sampah dengan algoritma CNN dan arsitektur MobileNetV2 dan MobileNetV3, mendapatkan nilai akurasi V2 sebesar 73.36%, dan V3 sebesar 69.51%. Penelitian lain juga dilakukan oleh Zhichao Chen d.k.k [15] mengklasifikasikan sampah dengan menggunakan GCNet, menghasilkan akurasi sebesar 97.9%

Penelitian ini melakukan klasifikasi limbah dengan menggunakan algoritma CNN dengan arsitektur MobileNet. Dalam penelitian ini menggunakan CNN beserta arsitektur MobileNet. Dataset yang digunakan didapat dari kaggle dataset pada bagian *garbage classification*, terdiri dari 2527 data yang terbagi dari 6 katagori.

## 2. METODE

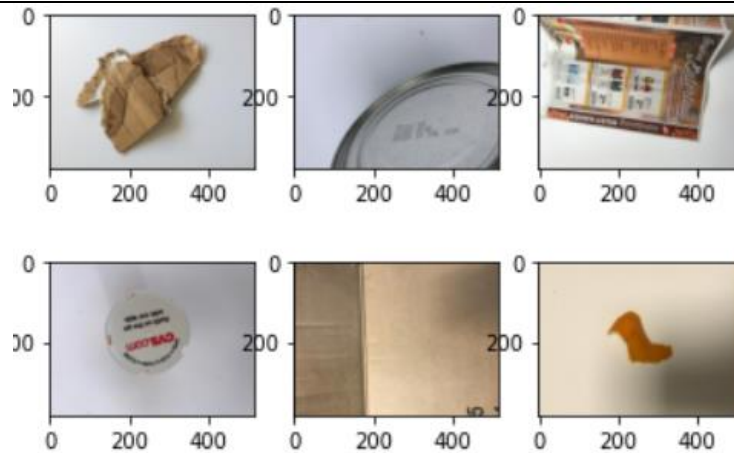
Pada penelitian ini menerapkan metode *Convolution Neural Network* (CNN) dan tahapan yang dilakukan ditunjukkan pada gambar 1. Tahap awal dimulai dengan mengumpulkan dataset, lalu pembentukan model, dilanjutkan dengan training dan melakukan testing sehingga menghasilkan klasifikasi citra limbah.



Gambar 1. Langkah Metode Penelitian

### 2.1 Mengumpulkan Dataset

Penelitian ini menggunakan dataset *garbage classification* dari kaggle yang terdiri dari 6 class kategori yaitu kardus sebanyak 393, kaca sebanyak 491, logam sebanyak 400, kertas sebanyak 584, plastic sebanyak 472, dan sampah sebanyak 127. Total dataset yang digunakan adalah 2527 data [13]. Contoh gambar dataset ada pada gambar 2. Pemrograman bahasa python digunakan untuk perancangan dan pemodelan sistem.



Gambar 2. Contoh Dataset

## 2.2 Pembentukan Model

Sebelum pemodelan, terdapat 3 bagian data yaitu data *training*, validasi, dan pengujian dengan menggunakan sampel data secara acak. Jumlah pelatihan adalah 70%, validasi adalah 15%, dan pengujian adalah 15%. Kemudian proses pemrosesan gambar dilakukan pada augmentasi.[16]. Struktur parameter *Convolution Neural Network* (CNN) terdiri dari jenis konvolusi dan jumlah lapisan dense [5]. Gambar 3 menunjukkan struktur parameter CNN. Parameter yang diuji dengan pemodelan MobileNet terdiri dari jenis konvolusi, *batchnorm* dan RELU. Konvolusi MobileNet terdiri *depthwise* konvolusi dan *pointwise* konvolusi. Batchnorm terdiri dari *batchnorm depthwise* dan *batchnorm pointwise*. RELU terdiri dari RELU *depthwise* dan RELU *pointwise*. Struktur parameter dari MobileNet ditunjukkan pada gambar 4.

| Layer (type)                                      | Output Shape | Param # |
|---|--------------|---------|
| global_average_pooling2d (GlobalAveragePooling2D) | (None, 1024) | 0       |
| dense_2 (Dense)                                   | (None, 1024) | 1049600 |
| dropout (Dropout)                                 | (None, 1024) | 0       |
| dense_3 (Dense)                                   | (None, 1024) | 1049600 |
| dropout_1 (Dropout)                               | (None, 1024) | 0       |
| dense_4 (Dense)                                   | (None, 512)  | 524800  |
| dropout_2 (Dropout)                               | (None, 512)  | 0       |
| dense_5 (Dense)                                   | (None, 6)    | 3078    |
| Total params: 5,855,942                           |              |         |
| Trainable params: 5,820,038                       |              |         |
| Non-trainable params: 35,904                      |              |         |

Gambar 3. Struktur Parameter CNN

| Layer (type)                       | Output Shape       | Param # |
|------------------------------------|--------------------|---------|
| conv_dw_13_bn (BatchNormalization) | (None, 7, 7, 1024) | 4096    |
| conv_dw_13_relu (ReLU)             | (None, 7, 7, 1024) | 0       |
| conv_pw_13 (Conv2D)                | (None, 7, 7, 1024) | 1048576 |
| conv_pw_13_bn (BatchNormalization) | (None, 7, 7, 1024) | 4096    |
| conv_pw_13_relu (ReLU)             | (None, 7, 7, 1024) | 0       |
| Total params: 3,228,864            |                    |         |
| Trainable params: 3,206,976        |                    |         |
| Non-trainable params: 21,888       |                    |         |

Gambar 4. Struktur Parameter MobileNet

### 2.3 Training Data Model

Tahap Terdapat 1766 data, atau 70% dari total dataset, digunakan untuk tahap *training* model. Training model adalah proses menggunakan pembelajaran mesin agar algoritma yang kita buat dapat mengingat pola kelas tertentu pada data yang kita latih. Model CNN dengan arsitektur MobileNet menggunakan alat augmentasi keras python[15]. Pada proses augmentasi terdapat parameter yang digunakan dan dapat dilihat pada tabel 1. Yang berisi parameter dan nilai yang terkandung. Setelah meningkatkan proses training dengan melakukan epoch sebanyak lima puluh kali, proses epoch dapat dilihat pada gambar 5 dengan tools yang tersedia untuk python.

Tabel 1. Parameter Augmentasi

| Parameter          | Nilai   |
|--------------------|---------|
| Fill Mode          | Nearest |
| Horizontal Flip    | True    |
| Parameter          | Nilai   |
| Share Range        | 0.2     |
| Rotation Range     | 40      |
| Width Shift Range  | 0.2     |
| Height Shift Range | 0.2     |
| Zoom Range         | 0.2     |

```
print("[INFO] training head...")
H = model.fit_generator(
    train_generator,
    steps_per_epoch=totalTrain // BATCH_SIZE,
    validation_data=validation_generator,
    validation_steps=totalVal // BATCH_SIZE,
    epochs=50)
```

Gambar 5. Training Data dengan Epoch

## 2.4 Testing Data Model

Gambar 6 menunjukkan proses evaluasi model, yang menggunakan 383 data pengujian, yang merupakan 15% dari data total. Proses ini dilakukan dengan Python.

```
# reset the testing generator and evaluate the network after
# fine-tuning just the network head
print("[INFO] evaluating after fine-tuning network head...")
test_generator.reset()
predIdxs = model.predict_generator(test_generator,
    steps=(totalTest // BATCH_SIZE) + 1)
predIdxs = np.argmax(predIdxs, axis=1)
print(classification_report(test_generator.classes, predIdxs,
    target_names=test_generator.class_indices.keys()))
# plot_training(H, 50, config.WARMUP_PLOT_PATH)
```

Gambar 6. Proses Evaluasi Model

## 2.5 Klasifikasi Citra

Setelah melakukan pemodelan, proses traing, dan testing model maka proses selanjutnya adalah mendesain fungsi prediksi. Proses prediksi bertujuan untuk bisa memprediksi data baru diluar data *training* dan data validasi. Pertama kita mendefinisikan preproses yang berfungsi mengubah citra menjadi *array* untuk masuk ke dalam model yang telah dilatih. Proses kerjanya image diubah ke RGB dan dilakukan *resize* dengan metode interpolasi. Proses selanjutnya adalah *reshape* yaitu mengubah *array* 3 dimensi menjadi 4 dimensi. Setelah proses diatas selanjutnya *load* model yang telah dilatih, setelah berhasil memanggil model maka selanjutnya memprediksi sebuah citra. Untuk proses prediksi citra ditunjukkan pada gambar 7 untuk proses reshape dan load model dan gambar 8. proses prediksi

```
# load the input image and then clone it so we can draw on it later
waste_types = ['cardboard', 'glass', 'metal', 'paper', 'plastic', 'trash']
test_d = 'data/test/' + waste_types[0] + '/'
# test_d = 'mytest/'
test_files = os.listdir(test_d)[0:16]
for i, fn in enumerate(test_files):

    # Set up subplot; subplot indices start at 1
    sp = plt.subplot(nrows, ncols, i + 1, facecolor='red')
    sp.axis('Off') # Don't show axes (or gridlines)
    # predicting images
    path = test_d + fn
    image = cv2.imread(path)

    img = load_img(path, target_size=TARGET_SIZE)

    output = image.copy()
    output = imutils.resize(output, width=400)
```

Gambar 7. Reshape dan Load Model

```
img = img_to_array(img)

# our model was trained on RGB ordered images but OpenCV represents
# images in BGR order, so swap the channels, and then resize to
# 224x224 (the input dimensions for VGG16)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image = cv2.resize(image, TARGET_SIZE)
# print(image.shape)

# convert the image to a floating point data type and perform mean
# subtraction
image = image.astype("float32") / 255.
img = np.expand_dims(img, axis=0)
img = preprocess_input(img)

# pass the image through the network to obtain our predictions
# preds = model.predict(np.expand_dims(image / 255., axis=0))[0]
preds = model.predict(img)[0]
i = np.argmax(preds)
label = waste_types[i]
# print(label)

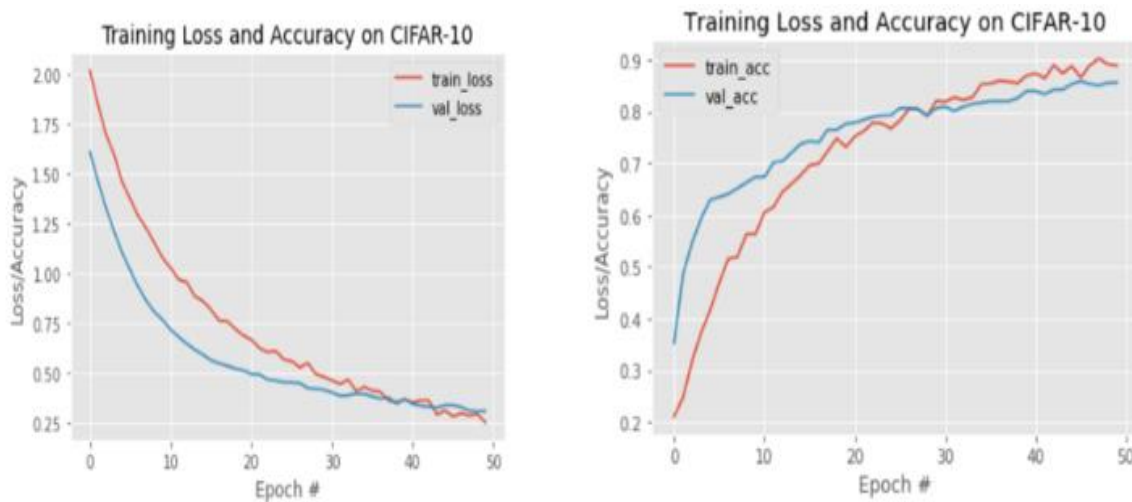
# draw the prediction on the output image
text = "{}: {:.2f}%".format(label, preds[i] * 100)
cv2.putText(output, text, (3, 20), cv2.FONT_HERSHEY_SIMPLEX, 1.05,
    (0, 128, 0), 2)
```

Gambar 8. Proses Prediksi

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Data Training

Terdapat 6 kategori dataset yang digunakan pada penelitian ini yang diambil dari data public kaggle. Dataset terdiri dari kardus sebanyak 393, kaca sebanyak 491, logam sebanyak 400, kertas sebanyak 584, plastik sebanyak 472, dan sampah sebanyak 127. Total dataset yang digunakan adalah 2527 data. Data tersebut dibagi menjadi dua data yaitu data training dan data validasi. Pada *image prosessing* menggunakan image Augmentasi dengan rotasi range 40, *fill mode* bernilai *nearest*, *horizontal flip* bernilai *true*, *shear range* bernilai 0.2, *width shift range* bernilai 0.2, *height shift range* bernilai 0.2, dan *zoom range* sebesar 0.2. Hasil dari proses augmentasi dapat dilihat pada gambar 9. Gambar 9 menunjukkan hasil 50 kali epoch dengan akurasi sebesar 0.91 dan 0.25 untuk nilai loss.



Gambar 9. Grafik akurasi dan Hasil Loss

#### 3.2 Hasil Perhitungan

Dua metode yang digunakan dalam penelitian untuk mendapatkan nilai akurasi adalah confosius matrik dan epoch 50 kali. Gambar 10 menunjukkan hasil *confosius matrik* dan *epoch* sebanyak 50 kali dapat dilihat pada tabel 2.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| cardboard    | 1.00      | 0.95   | 0.97     | 61      |
| glass        | 0.94      | 0.89   | 0.92     | 76      |
| metal        | 0.91      | 0.97   | 0.94     | 62      |
| paper        | 0.91      | 0.97   | 0.94     | 90      |
| plastic      | 0.93      | 0.93   | 0.93     | 73      |
| trash        | 0.83      | 0.71   | 0.77     | 21      |
| accuracy     |           |        | 0.93     | 383     |
| macro avg    | 0.92      | 0.90   | 0.91     | 383     |
| weighted avg | 0.93      | 0.93   | 0.93     | 383     |

Gambar 10. *Confosius* Matrik

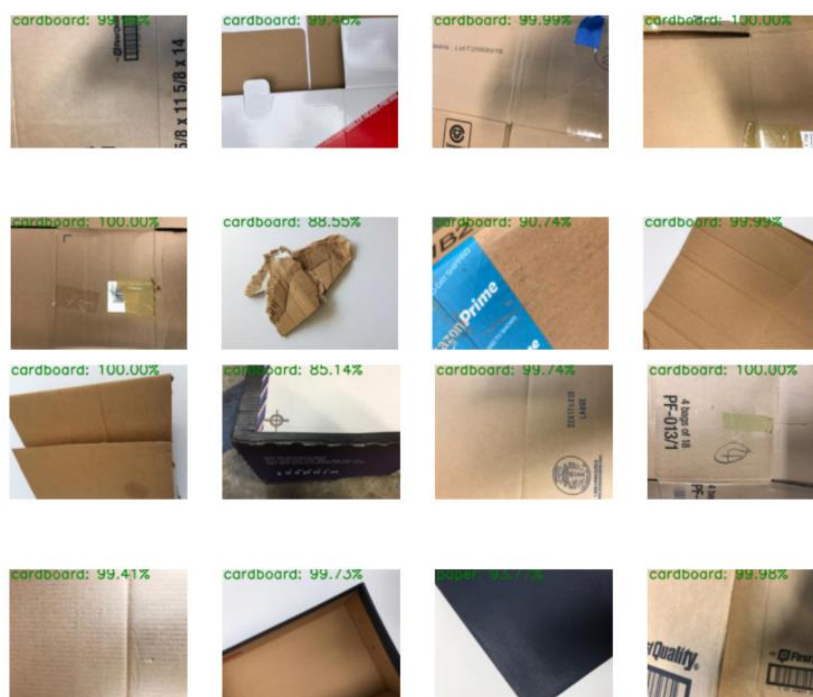
Dari hasil *confosius matrik* dapat dilihat nilai akurasi sebesar 0.93. Nilai f1 score masing – masing katagori adalah: Kardus sebesar 0.97, kaca sebesar 0.92, logam sebesar 0.94, kertas sebesar 0.94, plastic sebesar 0.93 dan trash sebnyak 0.77. Dari hasil epoch 50 kali didapat hasil akurasi terbesar pada nilai 0.91, nilai loss 0.25, val loss 0.3, dan nilai val akurasi sebesar 0.89.

Tabel 2. Epoch Accuracy dan Loss

| Epoch | Accuracy | Loss |
|-------|----------|------|
| 1     | 0.19     | 2.01 |
| 2     | 0.24     | 1.8  |
| 3     | 0.33     | 1.6  |
| 4     | 0.36     | 1.69 |
| 5     | 0.43     | 1.45 |
| 6     | 0.47     | 1.37 |
| 7     | 0.49     | 1.28 |
| 8     | 0.53     | 1.22 |
| 9     | 0.55     | 1.15 |
| 10    | 0.59     | 1.07 |
| 11    | 0.62     | 1.02 |
| 12    | 0.64     | 0.96 |
| 13    | 0.65     | 0.95 |
| 14    | 0.67     | 0.88 |
| 15    | 0.68     | 0.86 |
| 16    | 0.7      | 0.81 |
| 17    | 0.72     | 0.76 |
| 18    | 0.73     | 0.75 |
| 19    | 0.735    | 0.72 |
| 20    | 0.734    | 0.68 |
| 21    | 0.75     | 0.66 |
| 22    | 0.77     | 0.62 |
| 23    | 0.78     | 0.6  |
| 24    | 0.77     | 0.61 |
| 25    | 0.78     | 0.56 |
| 26    | 0.8      | 0.55 |
| 27    | 0.81     | 0.52 |
| 28    | 0.81     | 0.54 |
| 29    | 0.83     | 0.49 |
| 30    | 0.83     | 0.47 |
| 31    | 0.83     | 0.46 |
| 32    | 0.84     | 0.44 |
| 33    | 0.83     | 0.46 |
| ...   | ...      | ...  |
| ...   | ...      | ...  |
| ...   | ...      | ...  |
| 41    | 0.86     | 0.35 |
| 42    | 0.88     | 0.36 |
| 43    | 0.87     | 0.36 |
| 44    | 0.89     | 0.29 |
| 45    | 0.89     | 0.31 |
| 46    | 0.9      | 0.28 |
| 47    | 0.9      | 0.29 |
| 48    | 0.9      | 0.28 |
| 49    | 0.9      | 0.29 |
| 50    | 0.91     | 0.25 |

### 3.3 Hasil Klasifikasi

Dari hasil load citra yang sudah ditraining maka dapat dilakukan klasifikasi citra limbah tersebut kemudian akan memprossing citra menjadi array kemudian direshape menjadi tensor baru kemudian bisa untuk prediksi. Gambar 11 merupakan hasil dari prediksi citra limbah yang didapat dengan data diambil secara acak.



Gambar 11. Hasil Prediksi

Dari hasil prediksi yang didapat seperti pada gambar 11, dapat disimpulkan dengan tabel 3.

Tabel 3. Hasil Prediksi

| Citra | Katagori | Akurasi |
|-------|----------|---------|
| 1     | Kardus   | 93.93   |
| 2     | Kardus   | 99.4    |
| 3     | Kardus   | 99.9    |
| 4     | Kardus   | 100     |
| 5     | Kardus   | 100     |
| 6     | Kardus   | 85.5    |
| 7     | Kardus   | 90.7    |
| 8     | Kardus   | 99.9    |
| 9     | Kardus   | 100     |
| 10    | Kardus   | 85.14   |
| 11    | Kardus   | 99.74   |
| 12    | Kardus   | 100     |
| 13    | Kardus   | 99.41   |
| 14    | Kardus   | 99.75   |
| 15    | Kertas   | 93.77   |
| 16    | Kardus   | 99.98   |



#### 4. KESIMPULAN

Dari hasil percobaan dari penelitian klasifikasi citra limbah dengan algoritma *convolutional neural network* dengan arsitektur mobilenet menggunakan dataset sebanyak 2527 data dengan dibagi 6 katagori telah berhasil mengklasifikasikan citra limbah diperoleh akurasi sebesar 91% dengan epoch sebanyak 50 kali dan 25% untuk nilai *loss*. Untuk akurasi data testing diperoleh sebesar 93% dan nilai f1 score masing-masing class adalah Kardus sebesar 0.97, kaca sebesar 0.92, logam sebesar 0.94, kertas sebesar 0.94, plastik sebesar 0.93 dan trash sebanyak 0.77.

#### Daftar Pustaka

- [1] H. Peng, N. Shen, H. Ying, and Q. Wang, "Factor analysis and policy simulation of domestic waste classification behavior based on a multiagent study—Taking Shanghai's garbage classification as an example," *Environ. Impact Assess. Rev.*, vol. 89, no. October 2020, p. 106598, 2021, doi: 10.1016/j.eiar.2021.106598.
- [2] J. Qin, C. Wang, X. Ran, S. Yang, and B. Chen, "A robust framework combined saliency detection and image recognition for garbage classification," *Waste Manag.*, vol. 140, no. May 2021, pp. 193–203, 2022, doi: 10.1016/j.wasman.2021.11.027.
- [3] S. Meng and W. T. Chu, "A Study of Garbage Classification with Convolutional Neural Networks," *Indo - Taiwan 2nd Int. Conf. Comput. Anal. Networks, Indo-Taiwan ICAN 2020 - Proc.*, pp. 152–157, 2020, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181311.
- [4] S. Majchrowska *et al.*, "Deep learning-based waste detection in natural and urban environments," *Waste Manag.*, vol. 138, no. June 2021, pp. 274–284, 2022, doi: 10.1016/j.wasman.2021.12.001.
- [5] Y. Nan, J. Ju, Q. Hua, H. Zhang, and B. Wang, "A-MobileNet: An approach of facial expression recognition," *Alexandria Eng. J.*, vol. 61, no. 6, pp. 4435–4444, 2022, doi: 10.1016/j.aej.2021.09.066.
- [6] H. Oktafiandi, W. Winarnie, and ..., "Perbandingan Algoritma untuk Analisis Sentimen Terhadap Google Play Store Menggunakan Machine Learning," *J. Ekon. dan ...*, vol. 11, no. 2, pp. 16–21, 2023, [Online]. Available: <http://e-journal.polsa.ac.id/index.php/jnet/article/view/234%0Ahttp://e-journal.polsa.ac.id/index.php/jnet/article/download/234/159>
- [7] W. Winarnie, K. Kusriani, and A. D. Hartanto, "Pengurangan Dimensi dengan Metode Linear Discriminant Analist ( LDA ) Pendahuluan paru-paru . Deteksi dini pencegahan kanker payudara merupakan solusi dari masalah tersebut . Kanker payudara penyebab 22 % kematian oleh kanker pada wanita . [ 1 ]. Algoritm," *Infotek J. Inform. dan Teknol.*, vol. 6, no. 2, pp. 228–237, 2023.
- [8] B. Xu *et al.*, "CattleFaceNet: A cattle face identification approach based on RetinaFace and ArcFace loss," *Comput. Electron. Agric.*, vol. 193, p. 106675, 2022, doi: 10.1016/j.compag.2021.106675.
- [9] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, N. Suciati, and C. Fatichah, "Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for classifying the freshness of fish eyes," *Inf. Process. Agric.*, no. xxxx, pp. 1–12, 2022, doi: 10.1016/j.inpa.2022.01.002.
- [10] S. Zhang, Y. Chen, Z. Yang, and H. Gong, "Computer Vision Based Two-stage Waste Recognition-Retrieval Algorithm for Waste Classification," *Resour. Conserv. Recycl.*, vol. 169, no. February, p. 105543, 2021, doi: 10.1016/j.resconrec.2021.105543.
- [11] S. Liang and Y. Gu, "A deep convolutional neural network to simultaneously localize and recognize waste types in images," *Waste Manag.*, vol. 126, pp. 247–257, 2021, doi: 10.1016/j.wasman.2021.03.017.
- [12] C. Wang, J. Qin, C. Qu, X. Ran, C. Liu, and B. Chen, "A smart municipal waste management system based on deep-learning and Internet of Things," *Waste Manag.*, vol. 135, no. January, pp. 20–29, 2021, doi: 10.1016/j.wasman.2021.08.028.
- [13] Z. Kang, J. Yang, G. Li, and Z. Zhang, "An Automatic Garbage Classification System Based on Deep Learning," *IEEE Access*, vol. 8, pp. 140019–140029, 2020, doi: 10.1109/ACCESS.2020.3010496.
- [14] U. Kulkarni, S. M. Meena, S. V. Gurlahosur, and G. Bhogar, "Quantization Friendly MobileNet (QF-MobileNet) Architecture for Vision Based Applications on Embedded Platforms," *Neural Networks*, vol. 136, pp. 28–39, 2021, doi: 10.1016/j.neunet.2020.12.022.
- [15] Z. Chen, J. Yang, L. Chen, and H. Jiao, "Garbage classification system based on improved ShuffleNet v2," *Resour. Conserv. Recycl.*, vol. 178, no. November 2021, p. 106090, 2022, doi:

10.1016/j.resconrec.2021.106090.

- [16] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *Arxiv, Comput. Sci. > Comput. Vis. Pattern Recognit.*, Apr. 2017, doi: <https://doi.org/10.48550/arXiv.1704.04861>.