ISSN 2527-3671 (*Online*)
ISSN 2087-8893 (*Print*)

Teknologi: Jurnal Ilmiah Sistem Informasi 12 (1) Januari 2022 1-8

Tersedia *online* di www.journal.unipdu.ac.id

Halaman jurnal di www.journal.unipdu.ac.id/index.php/teknologi

Research article

Perbaikan klasifikasi sampah menggunakan *pretrained Convolutional Neural Network*

Muhammad H. Zayd [a,], Muhammad W. Oktavian [b], Dewa G. T. Meranggi [c], Javier A. Figo [d], Novanto Yudistira [e]*

[a,b,c,d] *Program Studi Teknik Informatika, Universitas Brawijaya, Malang, Indonesia*
[e] *Program Studi Teknologi Informasi, Universitas Brawijaya, Malang, Indonesia*

email: [a,*] hisyamzayd@student.ub.ac.id

*Korespondensi

Dikirim 25 April 2021; **Direvisi** 19 Mei 2021; **Diterima** 25 Agustus 2021; **Diterbitkan** 13 Mei 2022

Abstrak

Daur ulang sampah diperlukan demi keberlangsungan kehidupan bermasyarakat yang baik. Proses daur ulang sampah pada umumnya dilakukan secara manual dan dengan kategori sampah yang banyak jenisnya. Agar proses pemilahan sampah dapat berjalan lebih efektif dan efisien, diperlukan pemilahan sampah secara otomatis. Salah satu contohnya adalah dengan menggunakan sistem klasifikasi jenis sampah berdasarkan gambar menggunakan algoritma *deep learning* atau *machine learning*. Pada penelitian sebelumnya, AlexNet dan SVM digunakan sebagai algoritma klasifikasi jenis sampah pada enam kategori sampah yang berbeda. Hasil penelitian tersebut menunjukkan performa SVM lebih baik dengan akurasi sebesar 63% dibandingkan performa AlexNet dengan akurasi sebesar 20%. Padahal pada umumnya seharusnya performa algoritma CNN dapat menghasilkan akurasi yang lebih baik dibandingkan dengan SVM. Berdasarkan hal tersebut kami mengusulkan penelitian kembali terhadap klasifikasi jenis sampah pada *dataset* yang sama dengan penelitian sebelumnya tetapi dengan menggunakan arsitektur algoritma CNN yang terbaru yaitu ResNet. Lebih lengkapnya arsitektur yang digunakan adalah AlexNet, ResNet-18, dan ResNet-50 masing masing dengan dan tanpa pretrained, sehingga total terdapat enam jenis arsitektur algoritma CNN yang digunakan dalam *training*. Pada penelitian ini digunakan arsitektur AlexNet dengan konfigurasi yang berbeda dengan penelitian sebelumnya. Parameter *training* adalah jumlah *epoch* sebanyak 50 *epoch* dengan memperhatikan pada *epoch* ke berapa hasil *training* sudah menunjukkan konvergen. Dari hasil *training* didapatkan akurasi tertinggi diperoleh oleh arsitektur ResNet-50 dengan pretrained yakni sebesar 91,16% dan menunjukkan hasil yang konvergen sejak *epoch* ke-14. Kemudian untuk akurasi terendah diperoleh oleh arsitektur AlexNet tanpa *pretrained* yaitu sebesar 58% dan menunjukkan hasil yang konvergen sejak *epoch* ke-21.

Kata Kunci: AlexNet, *Convolutional Neural Network*, klasifikasi sampah, *pretrained network*, ResNet.

Improvement of garbage classification using pretrained Convolutional Neural Network

Abstract

Recycling waste is necessary for the sustainability of a good social life. The recycling process for waste is generally carried out manually and with many types of waste categories. For the waste sorting process to run more effectively and efficiently, automatic waste sorting is required. One example is by using an image-based waste type classification system using deep learning or machine learning algorithms. In previous research, AlexNet and SVM were used as classification algorithms for waste types in 6 different waste categories. The results of this study show that SVM performance is better with an accuracy of 63% compared to AlexNet's performance with an accuracy of 20%. Whereas in general, the CNN algorithm's performance should produce better accuracy than SVM. Based on this, we propose to re-examine the classification of waste types in the same dataset as previous research using the latest CNN algorithm architecture, namely ResNet. More completely, the architectures used are AlexNet, ResNet-18, and ResNet-50, respectively with and without pretrained, so that a total of 6 types of CNN algorithm architecture are used in the training. In this study, the AlexNet architecture is used with a different configuration from previous studies. The test parameter is the number of epochs of 50 epochs with attention to how many epochs the training results have shown convergent. From the training results, the highest accuracy obtained by the ResNet-50 architecture with pretrained is 91.16% and shows convergent results since the 14th epoch. Then for the lowest accuracy obtained by the AlexNet architecture without pretrained, which is 58% and shows convergent results since the 21st epoch.

Keywords: AlexNet, *Convolutional Neural Network, garbage classification, pretrained network, ResNet.*

Untuk mengutip artikel ini dengan *APA Style*:
Zayd, M. H., Oktavian, M. W., Meranggi, D. G. T., Figo, J. A., & Yudistira, N. (2022). Perbaikan klasifikasi sampah menggunakan pretrained Convolutional Neural Network. *Teknologi: Jurnal Ilmiah Sistem Informasi*, 12(1), 1–8. https://doi.org/10.26594/TEKNOLOGI.V0I0.2403

M. H. Zayd dkk./Teknologi 12 (1) Januari 2022 1-8
ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

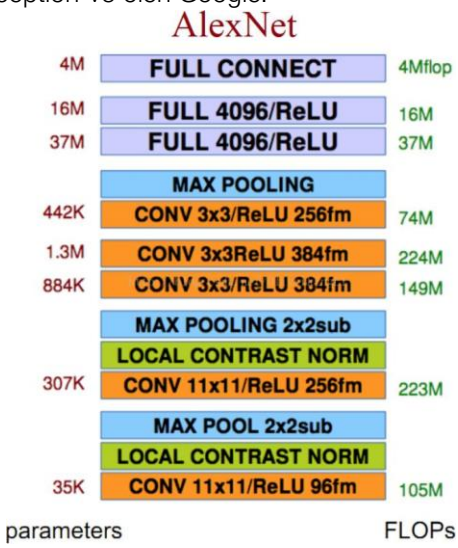
1. Pendahuluan

Peningkatan jumlah penduduk menghasilkan dampak yang serius terhadap semua aspek dalam suatu wilayah atau kota, salah satunya adalah meluapnya kuantitas sampah. Menurut Chandra (2006), sampah adalah segala sesuatu yang tidak digunakan, tidak dipakai, tidak disenangi, atau sesuatu yang dibuang yang berasal dari kegiatan manusia dan tidak terjadi dengan sendirinya. Berdasarkan data yang dikeluarkan oleh Kementrian Lingkungan Hidup dan Kehutanan pada tahun 2020, memperkirakan total berat sampah nasional mencapai 67,8 ton dengan asumsi, setiap Warga Negara Indonesia (WNI) menghasilkan sampah sebanyak 0,7 kg (Setiawan, 2021).

Proses daur ulang adalah salah satu solusinya. Dari sekian banyak proses daur ulang sampah, menurut Lestari et al. (2020) proses pemilahan sampah adalah yang paling penting. Apabila dilakukan pemilahan jenis sampah berdasarkan kategori tertentu terlebih dahulu, maka proses daur ulang akan berlangsung lebih cepat. Namun menurut Yang & Thung (2016), pada prakteknya para pembuang sampah terkadang bingung mengenai jenis kategori sampah yang dimilikinya dengan alasan banyaknya material penyusun sampah tersebut. Berdasarkan permasalahan tersebut, dilakukan penelitian terhadap klasifikasi kategori sampah untuk mengetahui apakah kategori tersebut dapat dilakukan proses daur ulang ataukah tidak (Yang & Thung, 2016).

Pada penelitian tersebut digunakan *Convolutional Neural Network* (CNN) AlexNet dan *Support Vector Machine* (SVM) sebagai algoritma pemilah sampah otomatis pada 6 kategori sampah berbeda (kardus, gelas (pecah belah), besi, kertas, plastik, dan sampah lainnya). Hasil penelitian menunjukkan SVM lebih baik dengan akurasi sebesar 63% di atas akurasi CNN yang hanya 20%. Pada umumnya seharusnya penggunaan CNN dapat menghasilkan akurasi yang lebih baik dibandingkan dengan SVM (Meng & Chu, 2020). Selain itu, beberapa penelitian yang memanfaatkan CNN dan variannya telah dilakukan pada beberapa topik diantaranya adalah pada bidang medis dan biologi dan menunjukkan performa dengan akurasi yang tinggi (Jiang et al., 2019; Jin et al., 2019; Yudistira et al., 2020). Bahkan apabila kita memanfaatkan teknik regularizer maka CNN berpotensi memiliki performa yang lebih tinggi lagi dari baseline (Kofler et al., 2020). Pada bidang kearifan lokal, CNN dan variannya mulai dimanfaatkan untuk klasifikasi berbagai varian batik di Indonesia (Meranggi et al., 2022). Tidak hanya data berbasiskan citra, CNN Nampak menunjukkan performa terbaik apabila dilatih pada data video untuk mengenali aksi manusia (Yudistira & Kurita, 2020).

Berdasarkan hal tersebut kami mengusulkan penelitian kembali terhadap klasifikasi sampah pada *dataset* yang sama dengan menggunakan arsitektur CNN yang lebih modern yaitu ResNet. ResNet adalah model CNN yang menjuarai ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) pada tahun 2015, unggul dari model CNN Inception-v3 oleh Google.



Gambar 1. Arsitektur dari AlexNet, mulai dari layer pertama (bawah) sampai dengan layer terakhir (atas) (ProgrammerSought, 2018)

2. State of the Art

2.1. AlexNet

AlexNet memperkenalkan konsep paralel dalam proses pelatihan CNN. Proses pelatihan pada CNN umumnya membutuhkan jumlah layer yang besar diikuti dengan jumlah data pelatihan yang juga besar. Konsep paralel ini terbagi menjadi 2:

ISSN 2502-3357 (*Online*)
M. H. Zayd dkk./Teknologi 12 (1) Januari 2022 1-8 ISSN 2503-0477 (*Print*)

- Pendekatan paralel pada model: Masing-masing *worker* dari CNN melakukan pelatihan CNN pada setiap bagian dari model yang berbeda-beda.
- Pendekatan paralel pada dimensi: Masing-masing *worker* dari CNN melakukan pelatihan CNN pada setiap bagian dari data yang berbeda-beda.

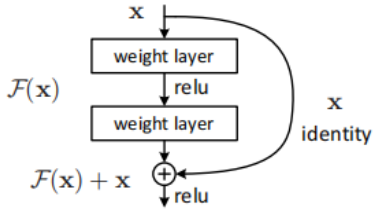
Pada Gambar 1 menunjukkan bahwa jumlah keseluruhan layer dari AlexNet adalah sebanyak 11 layer. Masing-masing layer memiliki dimensi filter atau *kernel* dan nilai *Stride* yang berbeda-beda. Berikut adalah deskripsi dimensi *kernel*, nilai *Stride*, jumlah *Input channel* dan *output channel*, fungsi aktivasi urutan layer dan jumlah parameter keseluruhan dari setiap layer pada AlexNet:

- Layer ke-1, jenis layer = Konvolusi. Dimensi *kernel* = 11×11. *Stride* = 4. *Input channel* = 3 (RGB, *red, green, blue*). *Output channel* = 96. Fungsi aktivasi = ReLU (Rectified Linear Unit). Jumlah parameter = 34.944.
- Layer ke-2, jenis layer = *Max pooling*. Dimensi *kernel* = 3×3. *Stride* = 2.
- Layer ke- 3, jenis layer = Konvolusi. Dimensi *kernel* = 5×5. *Stride* = 1. *Input channel* = 96. *output channel* = 256. Fungsi aktivasi = ReLU. Jumlah parameter = 614.656.
- Layer ke- 4, jenis layer = *Max pooling*. Dimensi *kernel* = 3×3. *Stride* = 2.
- Layer ke-5, jenis layer = Konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 256. *Output channel* = 384. Fungsi aktivasi = ReLU. Jumlah parameter = 885.120.
- Layer ke-6, jenis layer = Konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 384. *Output channel* = 384. Fungsi aktivasi = ReLU. Jumlah parameter = 1.327.488.
- Layer ke-7, jenis layer = Konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 384. *Output channel* = 256. Fungsi aktivasi = ReLU. Jumlah parameter = 884.922.
- Layer ke-8, jenis layer = *Max pooling*. Dimensi *kernel* = 3×3. *Stride* = 2.
- Layer ke-9, jenis layer = *Fully connected*. *Input channel* = 9216. *Output channel* = 4096. Fungsi aktivasi = ReLU. Jumlah parameter = 37.752.832.
- Layer ke-10, jenis layer = *fully connected*. *Input channel* = 4096. *Output channel* = 4096. Fungsi aktivasi = ReLU. Jumlah parameter = 16.781.312.
- Layer ke-11, jenis layer = *Fully connected*. *Input channel* = 4096. *Output channel* = 1000. Fungsi aktivasi = *Softmax*. Jumlah parameter = 4.097.000.

Jumlah parameter dari keseluruhan layer pada AlexNet adalah sebesar 62.378.344.

2.2. ResNet

Deep CNN telah menjadi bukti metode yang paling fenomenal untuk klasifikasi citra. *Deep* CNN secara tersirat mengintegrasikan fitur level yang rendah, sedang, dan tinggi pada gambar. Level fitur dapat diperkaya dengan semakin banyaknya jumlah layer pada *neural network*. Hal ini dibuktikan dengan penelitian klasifikasi ImageNet yang menunjukkan pada umumnya setiap peneliti menggunakan jaringan yang sangat dalam atau layer yang sangat banyak (He et al., 2016).



Gambar 2. *Identity Mapping* (He et al., 2016)

Namun yang masih menjadi pertanyaan adalah apakah betul dengan meningkatkan jumlah layer akan meningkatkan akurasi atau tingkat pembelajaran *neural network*? He et al. (2016) membuktikan bahwa bisa jadi semakin banyak layer semakin meningkatkan nilai *error* dan menurunnya tingkat akurasi. Hal ini bukanlah kasus *overfitting*, karena hasil klasifikasi pada data latih dan data tes menunjukkan hasil yang serupa. Meningkatnya nilai *error* disebabkan semakin dalam layer pada *neural network*, maka semakin kecil nilai kembali hasil perhitungan *gradient* untuk *update* bobot (He et al., 2016). He et al. (2016) mengajukan solusi dari permasalahan ini yaitu dengan menggunakan layer tambahan berupa *identity mapping*. *Identity mapping* ditunjukkan pada Gambar 2.

Penggunaan *identity mapping* yakni nilai *x* akan membantu *neural network* dalam klasifikasi dikarenakan memberikan penekanan nilai asli *x* terhadap hasil klasifikasi pendekatan terhadap *x* atau *f(x)*, sehingga dapat menurunkan nilai *error*, karena semakin dekatnya hasil *f(x)+x* dengan *x* (He et al., 2016).

ResNet mendapatkan penghargaan utama dalam kompetisi ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) pada tahun 2015 dengan tingkat *error* yang paling rendah yaitu 3,57% lebih baik dari VGG (*Visual Geometry Group*) (7,32%) dan GoogLeNet (6,66%). Penggunaan ResNet juga menurunkan tingkat *error* dengan semakin banyaknya layer yang ditambahkan pada *neural network*. Hal ini memecahkan permasalahan yang didefinisikan sebelumnya.

M. H. Zayd dkk./Teknologi 12 (1) Januari 2022 1-8
ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

layer name	output size	18-layer	34-layer	50-layer	
conv1	112×112	7×7, 64, stride 2			
conv2_x	56×56	3×3 max pool, stride 2			
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix}$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix}$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix}$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix}$
	1×1	average pool, 1000-d fc, softm			
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	

Gambar 3. Arsitektur ResNet-18, ResNet-34, dan ResNet-50 (He et al., 2016)

Jenis arsitektur dari ResNet dibedakan berdasarkan jumlah dari layer pada jaringan tersebut. Diantara arsitektur yang digunakan dalam pengujian untuk kompetisi ILSVRC adalah ResNet-18, ResNet-34, ResNet-50, ResNet-101, dan ResNet-152. Pada penelitian ini nantinya akan digunakan jenis ResNet-18 dan ResNet-50. Arsitektur ResNet-18, ResNet-34, dan ResNet-50 ditunjukkan pada Gambar 3.

Berikut adalah deskripsi dimensi *kernel*, nilai *stride*, jumlah *input channel* dan *output channel*, fungsi aktivasi urutan layer dan jumlah parameter keseluruhan dari setiap layer pada ResNet-18:

- Layer ke-1, jenis layer = Konvolusi. Dimensi *kernel* = 7×7. *Stride* = 2. *Input channel* = 3 (RGB). *Output channel* = 64. Fungsi aktivasi = ReLU.
- Layer ke-2, jenis layer = *Max pooling*. Dimensi *kernel* = 3×3. *Stride* = 2.
- Layer ke 3 sd. 6, jenis layer konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 64. *Output channel* = 64. Fungsi aktivasi = ReLU.
- Layer ke 7 sd. 10, jenis layer konvolusi. Dimensi *kernel* = 3x3. *Stride* = 1. *Input channel* = 64 pada layer ke 7 dan 128 pada layer lainnya, *Output channel* = 128. Fungsi aktivasi = ReLU.
- Layer ke 11 sd. 14, jenis layer konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 128 pada layer ke11 dan 256 pada layer lainnya. *Output channel* = 256. Fungsi aktivasi = ReLU.
- Layer ke 15 sd. 18, jenis layer konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 256 pada layer ke 15 dan 512 pada layer lainnya. *Output channel* = 512. Fungsi aktivasi = ReLU.
- Lalu diakhiri dengan *fully connected* layer dengan *average pooling* dan fungsi aktifasi = *Softmax*.

Berikut adalah deskripsi dimensi *kernel*, nilai *stride*, jumlah *input channel* dan *output channel*, fungsi aktivasi urutan layer dan jumlah parameter keseluruhan dari setiap layer pada ResNet-50:

- Layer ke-1, jenis layer = Konvolusi. Dimensi *kernel* = 7×7. *Stride* = 2. *Input channel* = 3 (RGB). *Output channel* = 64. Fungsi aktivasi = ReLU.
- Layer ke-2, jenis layer = *Max pooling*. Dimensi *kernel* = 3×3. *Stride* = 2.
- Layer ke 3, 6, dan 9 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 64 pada layer ke 3 dan 256 pada layer lainnya. *Output channel* = 64. Fungsi aktivasi = ReLU.
- Layer ke 4, 7, dan 10 jenis layer konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 64. *Output channel* = 256. Fungsi aktivasi = ReLU.
- Layer ke 5, 8, dan 11 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 64. *Output channel* = 256. Fungsi aktivasi = ReLU.
- Layer ke 12, 15, 18, dan 21 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 256 pada layer ke 12 dan 512 pada layer lainnya. *Output channel* = 128. Fungsi aktivasi = ReLU.
- Layer ke 13, 16, 19, dan 22 jenis layer konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 128. *Output channel* = 128. Fungsi aktivasi = ReLU.
- Layer ke 14, 17, 20, 23 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 128. *Output channel* = 512. Fungsi aktivasi = ReLU.
- Layer ke 24, 27, 30, 33, 36, dan 39 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 512 pada layer ke 24 dan 1.024 pada layer lainnya. *Output channel* = 256. Fungsi aktivasi = ReLU.
- Layer ke 25, 28, 31, 34, 37, dan 40 jenis layer konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 256. *Output channel* = 256. Fungsi aktivasi = ReLU.
- Layer ke 26, 29, 32, 35, 38 dan 41 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 256. *Output channel* = 1024. Fungsi aktivasi = ReLU.

ISSN 2502-3357 (*Online*)
M. H. Zayd dkk./Teknologi 12 (1) Januari 2022 1-8 ISSN 2503-0477 (*Print*)

- Layer ke 42, 45, dan 48 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 1.024 pada layer ke 42 dan 2.048 pada layer lainnya. *Output channel* = 512. Fungsi aktivasi = ReLU.
- Layer ke 43, 46, dan 49 jenis layer konvolusi. Dimensi *kernel* = 3×3. *Stride* = 1. *Input channel* = 512. *Output channel* = 512. Fungsi aktivasi = ReLU.
- Layer ke 44, 47, dan 50 jenis layer konvolusi. Dimensi *kernel* = 1×1. *Stride* = 1. *Input channel* = 512. *Output channel* = 2.048. Fungsi aktivasi = ReLU.
- Lalu diakhiri dengan *fully connected* layer dengan *average pooling* dan fungsi aktifasi = *Softmax*.

Jumlah parameter dari keseluruhan layer pada ResNet-18 dan ResNet-50 masing-masing sebesar 11.174.000 dan 23.521.000 (Ruiz, 2018). Jumlah masing-masing parameter dari ResNet-18 dan ResNet-50 jauh lebih sedikit jika dibandingkan dengan AlexNet dengan jumlah parameter sebesar 62.378.344.

2.3. Cross entropy loss

Cross entropy adalah ukuran dari bidang teori informasi, yang dibangun berdasarkan entropi dan biasanya menghitung perbedaan antara dua distribusi probabilitas. *Cross entropy* juga terkait dengan dan sering disalahartikan sebagai *logistic loss*, yang biasa disebut *log loss*. Meskipun kedua ukuran tersebut berasal dari sumber yang berbeda. Namun, ketika digunakan sebagai fungsi *loss* untuk model klasifikasi. Kedua ukuran tersebut menghitung kuantitas yang sama dan dapat digunakan secara bergantian (Brownlee, 2019).

Pada klasifikasi biner, Persamaan (1) digunakan untuk mencari nilai binary *cross entropy (BCE) loss,*

$$BCE = -(y \log(p) + (1-y) \log(1-p)) \quad (1)$$

di mana *p* adalah probabilitas *output* dari CNN (skala 0-1) dan *y* adalah label binari (0 atau 1). Kemudian untuk *cross entropy (CE) loss* klasifikasi *multiclass* dengan jumlah *class* lebih dari 2 menggunakan Persamaan (2),

$$CE = -\sum_{c=1}^{M} y_{o,c} log(\, p_{o,c}) \quad (2)$$

di mana *M* adalah banyaknya kelas, $y_{0,c}$ adalah label *output o* indeks (kelas) ke-*c* (0 atau 1). Sedangkan $p_{0,c}$ adalah probabilitas *output o* indeks (kelas) ke-*c* (skala 0-1).

3. Metode Penelitian

3.1. Lingkungan eksperimen

Penelitian dilakukan dengan menggunakan aplikasi program IDE Google Colab dengan *running time* GPU. Google Colab dipilih karena penggunaannya yang praktis dan tersedia fitur seperti penggunaan *resource computing* GPU yang cepat.

3.2. Perbedaan algoritma CNN yang digunakan dengan penelitian sebelumnya

Pada penelitian ini menggunakan 3 arsitektur CNN utama yaitu AlexNet, ResNet-18, dan ResNet-50 masing-masing dengan jenis CNN dengan *pretrained* dan tanpa *pretrained*, sehingga total menggunakan 6 arsitektur CNN.

Pada penelitian sebelumnya oleh Yang & Thung (2016), algoritma yang digunakan dalam klasifikasi jenis sampah adalah SVM dengan SIFT (*Scale Invariant Feature Transform*) *feature* dan AlexNet tetapi hanya ¾ bagian dari AlexNet, tidak keseluruhan. Konfigurasi SVM menggunakan radial basis *kernel*, karena *kernel* yang lain seperti *linear kernel* dan *polynomial kernel* tidak berjalan dengan baik (Yang & Thung, 2016). Kemudian nilai untuk parameter *C* adalah 1.000 dan nilai *Gamma* untuk SIFT adalah 0,5.

3.3. Perbedaan parameter training dengan penelitian sebelumnya

Meskipun dalam penelitian ini, salah satunya menggunakan arsitektur model yang sama yaitu AlexNet dan *dataset* yang sama dengan penelitian sebelumnya oleh *Yang & Thung (2016)*, terdapat perbedaan dalam penentuan parameter *training* dan proses penentuan bobot awal *training* pada kedua penelitian.

Parameter pelatihan yang digunakan dalam penelitian ini diantaranya adalah *learning rate*, jumlah *epoch*, fungsi untuk perhitungan *loss* dan fungsi *optimizer*. *Learning rate* adalah nilai koefisien yang menentukan besar kecil pengaruh gradien terhadap perubahan atau *update* pada bobot pada proses *training*. Nilai *learning rate* yang digunakan pada *training* adalah sebesar 0,001.

Epoch adalah jumlah keseluruhan iterasi dari proses *training*. Nilai *epoch* yang digunakan pada pelatihan sebanyak 50 *epoch* untuk semua model. Fungsi *loss* yang digunakan pada proses *training* adalah fungsi *cross entropy loss* dan *optimizer* yang digunakan adalah *Stochastic Gradient Descent* (SGD).

Kemudian pada penelitian sebelumnya oleh Yang & Thung (2016) dalam klasifikasi jenis sampah, parameter penelitian yang digunakan adalah dibagi menjadi 2 bagian yaitu parameter untuk algoritma SVM dan AlexNet. Pada SVM, selain menggunakan SIFT sebagai bagian *preprocessing*, dilakukan pembagian data untuk *training* dan *testing* sebesar 70:30. Pada AlexNet jumlah *epoch* adalah sebanyak 60 *epoch*, dengan *batchsize* sebesar 32, nilai *learning rate* sama dengan 5^{e-8}, menggunakan konsep *weight decay*

M. H. Zayd dkk./Teknologi 12 (1) Januari 2022 1-8
ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

setiap 5 *epoch*, penggunaan *L2 regularization* dengan nilai $7{,}5^{e-2}$, serta penggunaan *Kaiming weight initialization* sebagai metode untuk inisiasi nilai awal pada *neural network*.

3.4. Dataset

Dataset yang digunakan untuk proses *training* ini bersumber dari *website* penyedia data Kaggle yang bisa diakses melalui laman https://www.kaggle.com/asdasdasasdas/garbage-classification. *Dataset* ini terdiri atas 2.532 gambar sampah yang terbagi menjadi 6 jenis dengan rincian:

- Sampah kardus sejumlah 393 gambar.
- Sampah gelas sejumlah 491 gambar.
- Sampah besi sejumlah 400 gambar.
- Sampah kertas sejumlah 584 gambar.
- Sampah plastik sejumlah 472 gambar.
- Sampah lainnya sejumlah 127 gambar.

3.5. Preprocessing

Pada penelitian ini, sebelum dilakukan proses *training* atau pelatihan, citra masukan akan diolah ke dalam *preprocessing* terlebih dahulu, di mana akan terdapat beberapa proses yakni proses *resize*, *crop*, dan normalisasi. Pada proses *resize*, citra *input* akan diubah ukurannya dari 512×384 *pixel* menjadi 256×256 *pixel*, kemudian dilakukan proses *crop* atau pemotongan citra dengan ukuran 224 *pixel* pada bagian tengah (*center crop*), dan yang terakhir adalah normalisasi menggunakan standar deviasi dan rata-rata pada citra masukkan yang telah dilakukan proses *resize* dan *crop*.

Pada penelitian sebelumnya oleh Yang & Thung (2016), terdapat dua konfigurasi yang berbeda untuk *preprocessing*. Konfigurasi yang pertama adalah untuk metode SVM, yakni penggunaan metode SIFT untuk penyesuaian nilai warna setiap *pixel* dari setiap gambar. Kemudian konfigurasi yang kedua adalah ukuran gambar ditetapkan pada 256×256. Dari kedua konfigurasi tersebut tidak terdapat proses seperti *resize*, *crop*, normalisasi, ataupun rotasi.

3.6. Training

Setelah *preprocessing* selesai, tahapan berikutnya adalah *training*. Proses *training* merupakan tahapan pelatihan jaringan berdasarkan *dataset* yang ada untuk memperoleh akurasi yang tinggi dari klasifikasi yang dilakukan. Tahapan ini terdiri dari proses *feedforward* dan proses *backpropagation*:

- *Feedforward*
  Tahapan pada proses *feedforward* adalah pertama citra vektor akan melalui proses konvolusi dan *max pooling* untuk mereduksi ukuran citra dan memperbanyak neuron sehingga terbentuk banyak jaringan yang mana menambah *variant* data untuk dipelajari. Hasil dari proses *feedforward* berupa bobot yang akan digunakan untuk evaluasi.
- *Backpropagation*
  Proses *backpropagation* merupakan tahap kedua dari proses *training*. Pada tahap ini merupakan hasil proses dari *feedforward* yang dilakukan *trace error* atau kesalahannya dari lapisan *output* sampai lapisan pertama, kemudian dihitung nilai gradiennya. *Training* dilakukan terhadap seluruh 6 model arsitektur yang telah ditentukan sebelumnya.

4. Hasil dan Pembahasan

Dari proses *training* yang telah dilakukan didapatkan hasil klasifikasi sebagai berikut sebagaimana pada Tabel 1.

Tabel 1. Hasil akurasi, nilai *loss*, dan titik *epoch* konvergen pada setiap model setelah dilakukan proses *training*

Model	Akurasi	*Loss*	*Epoch*
AlexNet	0,5858	1,1763	21
ResNet18	0,7005	0,8262	20
ResNet50	0,6306	1,0069	26
Alexnet (*Pretrained*)	0,8166	0,5787	18
ResNet18 (*Pretrained*)	0,8799	0,3666	18
ResNet50 (*Pretrained*)	0,9116	0,3038	14

Sebagaimana pada Tabel 1, AlexNet mancapai akurasi dengan nilai yang paling rendah yaitu 58,58%. Apabila menggunakan *pretrained*, Alexnet dapat mencapai akurasi lebih baik yaitu sebesar 81,66% dengan konvergensi yang membutuhkan lebih sedikit *epoch* yaitu 18 *epoch*. Hal ini membuktikan bahwa dengan pretrained AlexNet membutuhkan *epoch* yang lebih sedikit untuk mencapai konvergen dan dapat menghasilkan akurasi yang lebih baik.

Memperhatikan hasil penelitian sebelumnya oleh Yang & Thung (2016) dalam klasifikasi sampah menggunakan SVM dan AlexNet, pada penelitian ini hasil akurasi AlexNet jauh lebih baik yaitu 58,58% jika dibandingkan dengan hasil akurasi AlexNet pada penelitian sebelumnya yaitu sebesar 20%. Hal ini

ISSN 2502-3357 (*Online*)
M. H. Zayd dkk./Teknologi 12 (1) Januari 2022 1-8 ISSN 2503-0477 (*Print*)

disebabkan pada penelitian sebelumnya hanya menggunakan ¾ layer dari AlexNet. Kemudian hal lain yang mempengaruhi peningkatan akurasi AlexNet pada penelitian ini adalah penggunaan *preprocessing* seperti *resize*, *crop*, dan juga normalisasi.

Kemudian pada penelitian sebelumnya, hasil akurasi daripada SVM adalah sebesar 63% yang menunjukkan masih lebih unggul di atas hasil akurasi AlexNet tanpa *pretrained* meskipun sudah menggunakan 100% layer daripada AlexNet dan dengan beberapa konfigurasi pada tahapan *preprocessing*.

Hasil daripada ResNet-18 dan ResNet-50 mencapai akurasi yang lebih baik daripada AlexNet baik dengan *pretrained* maupun tanpa *pretrained*. Hasil yang lebih baik ini dapat dicapai dikarenakan adanya *skip connection* yang menyebabkan ResNet terhindar dari fenomena yang dinamakan *vanishing gradient* yaitu gradien yang menjadi sangat kecil karena jumlah layer yang banyak. Kemudian penggunaan *pretrained* juga meningkatkan kedua jaringan ResNet.

Akurasi tertinggi dicapai oleh model arsitektur ResNet50 *pretrained* dengan nilai akurasi sebesar 91,16% dan akurasi terendah dicapai oleh model arsitektur AlexNet tanpa pretrained dengan nilai akurasi sebesar 58,58%.

5. Kesimpulan

Dari penelitian ini dapat disimpulkan bahwa 1) akurasi SVM terhadap klasifikasi sampah tetap lebih baik di atas AlexNet yakni sebesar 63% dibandingkan 58,58%; 2) Penggunaan 100% layer pada jaringan AlexNet menghasilkan akurasi yang lebih baik dibandingkan hanya ¾ bagian saja; 3) Penggunaan konfigurasi untuk data pada tahapan *preprocessing* seperti *resize*, *crop*, dan normalisasi dapat meningkatkan akurasi AlexNet; 4) Akurasi tertinggi dicapai oleh model ResNet50 *pretrained* yaitu dengan hasil akurasi sebesar 91,16%; 5) Akurasi terendah dicapai oleh model AlexNet tanpa pretrain dengan hasil akurasi sebesar 58,58%; 6) ResNet menghasilkan nilai akurasi yang lebih baik daripada AlexNet; 7) Model dengan *pretrained* terbukti dapat meningkatkan kecepatan waktu untuk mencapai konvergen dan meningkatkan akurasi.

Penelitian selanjutnya adalah bagaimana meningkatkan akurasi dari prediksi dengan melakukan pelatihan model menggunakan beberapa teknik augmentasi data sepeti CutMix dan MixUp, teknik regularisasi seperti sparse regularization, dan domain adaptation. Model-model baru deep learning yang terlebih dahulu dilatih pada dataset citra yang besar seperti ImageNet dapat digunakan seperti DenseNet (Huang et al., 2017) atau *Vision Transformer* (Zhou et al., 2021). Apabila kita menginginkan suatu arsitektur yang lebih efisien dan dapat diimplementasikan dalam perangkat-perangkat keras sederhana maka MobileNet (Howard et al., 2017), EfficientNet (Tan & Le, 2019), atau ShufffleNet (Zhang et al., 2018) dapat dimanfaatkan sebagai model *backbone* CNN-nya.

6. Kontribusi Penulis

M. H. Zayd: *Data curation, Formal Analysis, Investigation, Methodology, Software, Visualization, dan Writing – original draft*. M. W. Oktavian: *Investigation, Methodology, Software, dan Writing – original draft*. D. G. T. Meranggi: *Data curation, Methodology, Visualization, dan Writing – original draft*. J. A. Figo: *Software dan Visualization*. N. Yudistira: *Conceptualization, Funding acquisition, Supervision, Validation, dan Writing – review & editing*.

7. Declaration of Competing Interest

Penulis menyatakan tidak ada konflik kepentingan.

8. Referensi

Brownlee, J. (2019). *A Gentle Introduction to Cross-Entropy for Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/cross-entropy-for-machine-learning/

Chandra, B. (2016). Pengantar Kesehatan Lingkungan. In *EGC*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. http://image-net.org/challenges/LSVRC/2015/

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. https://doi.org/10.48550/arxiv.1704.04861

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Jiang, J., Hu, Y. C., Liu, C. J., Halpenny, D., Hellmann, M. D., Deasy, J. O., Mageras, G., & Veeraraghavan, H. (2019). Multiple resolution residually connected feature streams for automatic lung tumor

ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

segmentation from CT images. *IEEE Transactions on Medical Imaging*, *38*(1), 134–144. https://doi.org/10.1109/TMI.2018.2857800

Jin, Q., Meng, Z., Pham, T. D., Chen, Q., Wei, L., & Su, R. (2019). DUNet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems*, *178*, 149–162. https://doi.org/10.1016/J.KNOSYS.2019.04.025

Kofler, A., Haltmeier, M., Schaeffter, T., Kachelrieß, M., Dewey, M., Wald, C., & Kolbitsch, C. (2020). Neural networks-based regularization for large-scale medical image reconstruction. *Physics in Medicine & Biology*, *65*(13), 135003. https://doi.org/10.1088/1361-6560/AB990E

Lestari, N. E., Purnama, A., Safitri, A., & Koto, Y. (2020). Peningkatan Pengetahuan dan Sikap Pemilahan Sampah Pada Anak Usia Sekolah Melalui Metode Simulasi. *Jurnal Pengabdian Masyarakat Indonesia Maju*, *1*(2). https://journals.stikim.ac.id/index.php/JLS1/article/view/668

Meng, S., & Chu, W.-T. (2020). A Study of Garbage Classification with Convolutional Neural Networks. *2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, 152–157. https://doi.org/10.1109/Indo-TaiwanICAN48429.2020.9181311

Meranggi, D. G. T., Yudistira, N., & Sari, Y. A. (2022). Batik Classification Using Convolutional Neural Network with Data Improvements. *JOIV : International Journal on Informatics Visualization*, *6*(1), 6–11. https://doi.org/10.30630/JOIV.6.1.716

ProgrammerSought. (2018). *Lenet-5, Alexnet detailed explanation and tensorflow code implementation*. Programmer Sought. https://www.programmersought.com/article/54943696781/

Ruiz, P. (2018). *ResNets for CIFAR-10*. Towards Data Science. https://towardsdatascience.com/resnets-for-cifar-10-e63e900524e0

Setiawan, A. (2021). *Membenahi Tata Kelola Sampah Nasional*. Indonesia.Go.Id. https://indonesia.go.id/kategori/indonesia-dalam-angka/2533/membenahi-tata-kelola-sampah-nasional

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of Machine Learning Research*, 6105–6114. http://proceedings.mlr.press/v97/tan19a.html

Yang, M., & Thung, G. (2016). Classification of Trash for Recyclability Status. *CS 229 Machine Learning*, 940–945.

Yudistira, N., Kavitha, M., Itabashi, T., Iwane, A. H., & Kurita, T. (2020). Prediction of Sequential Organelles Localization under Imbalance using A Balanced Deep U-Net. *Scientific Reports*, *10*(1). https://doi.org/10.1038/s41598-020-59285-9

Yudistira, N., & Kurita, T. (2020). Correlation Net: Spatiotemporal multimodal deep learning for action recognition. *Signal Processing: Image Communication*, *82*, 115731. https://doi.org/10.1016/J.IMAGE.2019.115731

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6848–6856. https://doi.org/10.1109/CVPR.2018.00716

Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., & Feng, J. (2021). *DeepViT: Towards Deeper Vision Transformer*. https://doi.org/10.48550/arxiv.2103.11886