

IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE ON THE DEVELOPMENT OF AUTOMATIC LEATHER PRODUCT INDUSTRIAL WASTE DETECTION SYSTEM

IMPLEMENTASI ARTIFICIAL INTELLIGENCE PADA PENGEMBANGAN SISTEM DETEKSI LIMBAH INDUSTRI PRODUK KULIT OTOMATIS

Eka Legya Frannita^{1,*}, dan Mochammad Charis Hidayatullah¹, Naimah Putri²

¹Department of Leather Product Processing Technology, Politeknik ATK Yogyakarta, Yogyakarta, Indonesia

²Department of Leather Processing Technology, Politeknik ATK Yogyakarta, Yogyakarta, Indonesia

* Corresponding author: *eka.legya@atk.ac.id

Abstract:

Rapid globalization has led to the growth of a significant amount of waste, including in the industry. Due to the considerable amount of growth of waste every year, effective and efficient waste management is needed to protect our environment. In the leather product industry, waste management is strongly important since it may have a significant impact to the employee and production process. Regarding those issues, waste management technology is considered proposed in order to solve those problems. Current research reported the outstanding work of implementing artificial intelligence for detecting and recognizing industrial waste. Artificial intelligence was proven to be a highly recommended approach that is able to classify several waste types with outstanding performance. Regardless of those facts, artificial intelligence still remains several hurdles, such as the high computational demands, especially for deep learning networks. Regarding the mentioned issue, we proposed a more proper deep learning network for recognizing industrial waste. In this research work, we use Single Shot Detector (SSD) to recognize and classify industrial waste. Our proposed solution was performed in the TrashNet dataset and Waste Picture dataset. Our proposed solution successfully achieve mAP of 0.8813, accuracy of 0,9795, precision of 0,9985 and recall of 0,9693 in the training process. Whereas, in the testing process we achieve average accuracy of 0,8254. According to those results, we can conclude that our proposed solution is suitable for industrial waste detection and has potential to be implemented as an embedded system for recognizing industrial waste automatically.

Keywords: artificial intelligence, deep learning, automatic industrial waste detection.

Intisari:

Pertumbuhan globalisasi yang cepat telah menyebabkan pertumbuhan limbah dalam jumlah yang signifikan, termasuk di industri. Karena pertumbuhan sampah yang cukup besar setiap tahun, pengelolaan sampah yang efektif dan efisien diperlukan untuk melindungi lingkungan kita. Dalam industri produk kulit, pengelolaan limbah sangat penting karena dapat berdampak

signifikan terhadap karyawan dan proses produksi. Berkaitan dengan permasalahan tersebut, teknologi pengelolaan sampah dipertimbangkan untuk dapat mengatasi permasalahan tersebut. Penelitian terbaru menunjukkan hasil yang luar biasa dalam menerapkan kecerdasan buatan untuk mendeteksi dan mengenali limbah industri. Kecerdasan buatan terbukti menjadi pendekatan yang sangat direkomendasikan yang mampu mengklasifikasikan beberapa jenis limbah dengan kinerja yang luar biasa. Namun, terlepas dari fakta tersebut, kecerdasan buatan masih memiliki beberapa kendala, seperti tuntutan komputasi yang tinggi, terutama untuk implementasi *deep learning*. Sehubungan dengan masalah tersebut, penulis mengusulkan pengembangan arsitektur *deep learning* yang lebih tepat untuk mengenali limbah industri. Dalam penelitian ini, penulis menggunakan *Single Shot Detector* (SSD) untuk mengenali dan mengklasifikasikan limbah industri. Solusi yang penulis usulkan dilakukan di dataset TrashNet dan dataset Waste Picture. Solusi yang penulis usulkan berhasil mencapai mAP sebesar 0,8813, akurasi sebesar 0,9795, presisi sebesar 0,9985 dan *recall* sebesar 0,9693 dalam proses pelatihan. Sedangkan pada proses pengujian penulis memperoleh akurasi rata-rata sebesar 0,8254. Berdasarkan hasil tersebut, dapat disimpulkan bahwa solusi yang diusulkan cocok untuk deteksi limbah industri dan berpotensi untuk diimplementasikan sebagai sistem tertanam untuk mengenali limbah industri secara otomatis.

Kata kunci: *artificial intelligence, deep learning*, deteksi sampah industri otomatis.

Pendahuluan

Sampah merupakan permasalahan global yang sangat mengganggu aktivitas dan keberlangsungan hidup masyarakat. Menurut *The World Counts* setiap tahunnya 2,12 miliar ton sampah dibuang. Sampah yang semakin banyak ini tentunya memberikan dampak antara lain polusi tanah, polusi udara, polusi lautan, dan polusi air tanah. Bahan kimia berbahaya dari sampah dapat mencemarkan tanah yang kemudian tercampur ke sumber bahan makanan. Proses pembakaran sampah di tempat pembuangan sampah dapat melepaskan bahan beracun seperti dioxin ke udara. Pada tahun 2050, diprediksi bahwa lautan akan dipenuhi oleh sampah plastik yang jumlahnya dapat melebihi populasi ikan. Selain itu, sebanyak 280 miliar ton air tanah tercemar tiap tahunnya [1].

Kulit adalah produk hasil daur ulang dari limbah padat industri daging. Sektor kulit merupakan salah satu industri terkemuka yang memegang nilai perdagangan global sebesar USD 394,12 miliar pada tahun 2020 dan diperkirakan akan tumbuh pada tingkat pertumbuhan tahunan gabungan (CAGR) sebesar 5,9% dari tahun 2021 hingga 2028 dengan perkiraan nilai perdagangan sebesar USD 624,08 miliar pada 2028 [2]. Produksi kulit global tahunan diperkirakan mencapai 23 miliar persegi *feet* di mana sektor alas kaki kulit mendominasi pasar dengan pangsa 65% dan jenis produk kulit lainnya sebesar 35% [3]. Proses pengolahan kulit menjadi produk ini tentunya membawa dampak tersendiri. Dampak positif yang diberikan adalah terpenuhinya kebutuhan masyarakat akan produk-produk kulit seperti alas kaki, tas hingga pakaian. Namun disisi lain, pengolahan produk kulit yang melimpah juga memberikan dampak negatif yakni beban polusi yang tidak terkendali. Berbagai jenis pemrosesan kulit yang dilakukan tentunya berpengaruh pada kuantitas limbah (dalam bentuk cair dan padat) yang dihasilkan. Hal ini terjadi karena pada dasarnya dari 1000 kg kulit hewan hanya menghasilkan 200 kg bahan kulit untuk membuat suatu produk. Akibatnya, proses pengolahan produk kulit meninggalkan hampir 70–80% limbah padat yang menyebabkan beban lingkungan yang sangat besar [2].

Dalam mendukung perkembangan industri untuk masa depan yang lebih baik, keseimbangan antara proses produksi produk kulit dan pencemaran akibat timbulan sampah harus diimbangi dengan teknologi pengelolaan sampah yang lebih baik [4]. Mengintegrasikan pencegahan polusi ke dalam sistem pengelolaan limbah industri sangat penting untuk mencegah terjadinya polusi udara dan tanah [5]. Pada dasarnya, pengelolaan limbah industri memerlukan pengomposan lahan, daur ulang, penimbunan, dan pemisahan limbah. Pemisahan adalah pemisahan sampah berdasarkan jenisnya untuk pembuangan yang efisien. Pengomposan proses memperbaiki tanah dengan penambahan bahan organik setelah sampah terurai. Daur ulang adalah proses menggunakan kembali atau menggunakan kembali bahan limbah untuk mengurangi jumlah sampah yang dapat dihasilkan. Proses yang dijelaskan di atas juga menggunakan sejumlah teknologi pengelolaan sampah yang sudah ada di fasilitas pengelolaan sampah [6]–[8]. Dari beberapa proses di atas, proses pemisahan sampah merupakan proses awal yang sangat penting untuk dilakukan karena dapat berpengaruh pada proses-proses yang lain [9].

Artificial intelligence atau yang sering kita kenal dengan istilah AI merupakan teknologi yang sangat bermanfaat dalam membantu berbagai macam aktivitas manusia. Dalam industri 4.0, pemanfaatan AI telah memberikan dampak yang signifikan dalam proses manufaktur. Penerapan AI dalam proses produksi telah terbukti memberikan efisiensi dan efektivitas yang sangat baik sehingga dapat menunjang proses produksi [10]. Dalam pengelolaan limbah produksi, pemanfaatan AI juga memiliki potensi yang sangat besar. Berdasarkan beberapa penelitian terdahulu, AI, khususnya *deep learning*, memiliki peran yang signifikan pada proses pemisahan sampah [11]–[13]. Sebagai contoh, penelitian yang dilakukan oleh Dang dkk yang menciptakan arsitektur untuk mengenali jenis limbah logam. Dalam arsitektur tersebut, empat model DNN yang berbeda, yaitu ResNet, AlexNet, GoogleNet, dan VGGNet, dibandingkan dan dianalisis guna menentukan model yang paling sesuai untuk memisahkan jenis-jenis limbah logam [14]. Bobulski dkk mengembangkan metode berbasis CNN untuk mengenali jenis-jenis sampah plastik [15]. Metode *object detection* juga telah banyak digunakan untuk mendeteksi jenis-jenis sampah [16]. Seperti penelitian yang dilakukan oleh Ku dkk [17] dan Xiao dkk [18] tentang penggunaan RCNN untuk mendeteksi limbah konstruksi. Selain itu juga terdapat penelitian yang dilakukan oleh Watanabe dkk [19] tentang penerapan Yolo-v3 untuk mendeteksi sampah laut bawah air. Hasil penelitian tersebut kemudian diteliti lebih lanjut oleh Kong dkk [20] dan telah diimplementasi untuk mendukung proses pengumpulan sampah plastik di dalam air secara otomatis.

Beberapa penelitian terdahulu di atas menunjukkan bahwa penggunaan *deep learning* untuk mendeteksi sampah sangat membantu proses pemisahan sampah. Namun, meskipun menghasilkan performa yang sangat baik, penggunaan *deep learning* juga memiliki kekurangan yakni memiliki waktu komputasi yang cenderung sangat lama. Untuk mengatasi hal tersebut, penelitian ini memberikan kontribusi sebagai berikut:

1. Mengembangkan *object detection* menggunakan *Single Shot Detector* (SSD) FPN Lite sebagai metode *deep learning* dengan komputasi yang ringan untuk mendeteksi limbah industri.
2. Melakukan studi komparasi antara metode yang diusulkan dengan penelitian-penelitian sebelumnya untuk mengevaluasi reliabilitas performa dari metode yang diusulkan.

Metode Penelitian

Alat dan Bahan

Penelitian ini menggunakan dataset publik untuk pelatihan model. Terdapat dua dataset public yang digunakan dalam penelitian ini. Dataset pertama adalah dataset TrashNet [21]. Dataset TrashNet diperoleh dari github dengan jumlah data sebanyak lebih dari 2100 gambar. Gambar diambil dengan kamera ponsel menggunakan sinar matahari dan/atau pencahayaan ruangan. Objek yang difoto diletakkan di atas latar belakang putih atau memenuhi seluruh tampilan. Semua gambar berukuran 512x384 piksel. Selain menggunakan dataset Trashnet, penelitian ini juga menggunakan dataset Waste Picture [22]. Dataset Waste Pictures terdiri atas hampir 24.000 gambar sampah yang diambil dari pencarian Google. Seluruh data tersebut dikelompokkan ke dalam 34 kelas. Dataset ini berisi data yang sangat beragam. Ukurannya juga berbeda secara signifikan. Sebagian besar foto berukuran di bawah 2000x2000 piksel.

Perangkat lunak yang digunakan dalam penelitian ini terdiri dari *Code Editor*, *web browser*, dan *labeling software*. Perangkat lunak *Code Editor* yang digunakan adalah *Visual Studio Code* dan *Jupyter Notebook*. Kedua *Code Editor* tersebut digunakan untuk menulis kode dengan bahasa *Python 3.8* dan *framework Tensorflow*. Selanjutnya *web browser* dan *Google Chrome* digunakan untuk membuka *Google Colaboratory* saat training dan evaluasi secara *online*. Selanjutnya software *Labellmg* digunakan untuk proses labeling data. Selain itu penelitian ini juga memanfaatkan *Roboflow* untuk melakukan augmentasi data.

Metode

Secara umum, pengembangan metode yang diusulkan dilakukan dalam lima tahapan. Kelima tahapan tersebut berkaitan satu sama lain. Berikut adalah rincian dari masing-masing tahapan yang dilakukan dalam penelitian yang diusulkan:

1. Tahap akuisisi data

Tahap akuisisi data merupakan tahap yang bertujuan untuk menyiapkan data dan memastikan data yang digunakan dalam penelitian sudah tepat dan sesuai dengan metode yang dipilih. Dalam tahap ini ada beberapa yang dilakukan. Tahap awal dilakukan proses

labelling dan pembuatan koordinat objek sampah. Proses pembuatan ground truth ini dilakukan dengan menggunakan tools *Labellmg*. *Labellmg* merupakan sebuah tools yang berfungsi untuk membantu proses anotasi objek. Hasil dari proses ini adalah file xml yang didalamnya terdapat nama file citra, koordinat objek dan label (kelas) dari objek.

2. Tahap perbaikan citra

Tahap perbaikan citra bertujuan untuk memperbaiki kontras serta pencahayaan citra agar menjadi lebih jelas. Penelitian ini menggunakan metode *contrast strecting* untuk melakukan perbaikan citra. Metode ini bekerja dengan cara meregangkan *range* intensitas piksel pada gambar. Metode perbaikan citra dengan menggunakan *contrast strecting* dilakukan dengan menerapkan formula berikut [23]:

$$s = (r - r_{min}) \frac{(I_{max} - I_{min})}{(r_{max} - r_{min})} + I_{min} \quad (1)$$

dimana,

r = nilai intensitas piksel

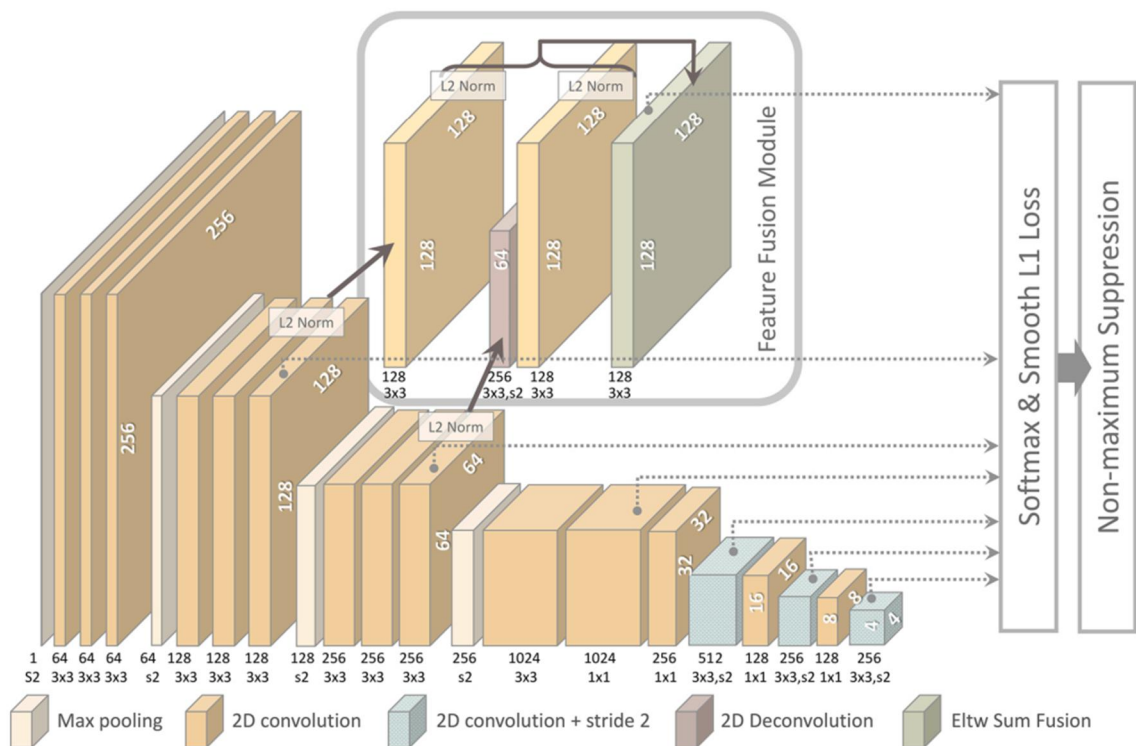
I_{max} = nilai intensitas piksel tertinggi (255)

I_{min} = nilai intensitas piksel tertinggi (0)

Dalam penelitian ini, proses perbaikan citra dengan menggunakan metode contrast stretching diawali dengan melakukan proses normalisasi. Proses normalisasi ini dilakukan dengan cara membagi pixel value dalam gambar dengan angka 255 (sebagai intensitas tertinggi pada citra RGB). Selanjutnya akan diperoleh gambar dengan intensitas yang bernilai antara 0 hingga 1.

3. Tahap pelatihan model

Setelah dilakukan perbaikan citra menggunakan contrast stretching, proses selanjutnya ada pelatihan model *object detection* menggunakan arsitektur *single-shot detector* (SSD). SSD bekerja seperti YOLO yakni membagi citra ke dalam beberapa bagian, akan tetapi dalam setiap bagian, *anchor* dengan beragam skala dan rasio aspek dibentuk untuk memperkirakan luas area *bounding box* [24]. Model arsitektur SSD yang digunakan dalam penelitian ini diilustrasikan pada Gambar 1 [25].



Gambar 1. Skema model *feature-fused single-shot detector* (SSD) yang dibangun dan dikembangkan [25].

Penelitian ini membangun SSD dasar dengan 17 lapisan konvolusi dengan dan tanpa modul *feature-fused* tambahan menggunakan metode *Element-sum* [26]. Penelitian ini dilakukan dengan mengimplementasikan Keras SSD yang dimodifikasi dan disesuaikan dengan Keras 2.3.1 serta ditambahkan backend Tensorflow 2.1. Prediksi dibuat pada enam skala resolusi, dengan normalisasi L2 digunakan untuk dua lapisan deteksi pertama.

Pada proses pelatihan SSD, penelitian ini menetapkan label kelas biner ke setiap *anchor* atau *default box*. *Anchor* positif dan negatif dipilih berdasarkan besarnya *intersection over union* (IoU) dengan kotak pembatas *ground-truth*. *Anchor* yang memiliki IoU lebih besar dari 0,2 diberi label sebagai kelas positif, sedangkan *anchor* dengan IoU kurang dari 0,1 diberi label sebagai kelas negatif. *Anchor* dengan IoU antara dua ambang batas dikeluarkan dari pelatihan model. Dengan parameter ini, penulis meminimalkan perhitungan kesalahan dalam melakukan klasifikasi serta kesalahan dalam proses lokalisasi. Berikut adalah formula yang digunakan [27][28].:

$$L(p, u, t^u, v) = L_{cls} + \lambda u L_{loc}(t^u, v) \quad (1)$$

di mana p adalah probabilitas prediksi *anchor* adalah objek sampah, u adalah kelas *ground-truth*, t adalah *bounding box* hasil dari regresi, dan v adalah *bounding-box* dari *ground-truth*. $L_{cls}(p, u)$ adalah loss dari proses klasifikasi dimana log loss untuk *true class* u didefinisikan dengan $-\log p_u$. L_{loc} didefinisikan sebagai loss dari proses lokalisasi yang dilakukan dengan menggunakan regresi. Bobot loss pada proses lokalisasi diatur pada nilai 1 sesuai dengan penelitian yang dilakukan oleh Liu dkk [27][28].

4. Evaluation metrics

Evaluation metrics merupakan *metrics* yang digunakan untuk mengevaluasi dan mengukur performa dari model yang telah dikembangkan. Dalam penelitian ini, proses evaluasi dilakukan dalam dua cara yakni evaluasi deteksi dan evaluasi klasifikasi. Evaluasi deteksi dilakukan dengan mengukur *mean average precision* (mAP). Penggunaan mAP disesuaikan dengan metode deteksi yang digunakan dalam penelitian ini. Penelitian ini menggunakan *object detection method* dengan prinsip kerja RPN dimana dalam mendeteksi suatu objek dilakukan dengan 3 rasio (1:1, 1:2 dan 2:1) serta 3 *scale* atau skala (128, 256 dan 512). Sehingga untuk satu objek kemungkinan paling banyak bisa terdeteksi kandidat sebanyak 9 kandidat. Menanggapi hal tersebut maka diperlukan pengukuran yang presisi terhadap kandidat mana yang paling mirip dengan *ground truth*. Sehingga penelitian ini juga memilih mAP agar kandidat yang terdeteksi merupakan kandidat terbaik dari beberapa kandidat yang telah berhasil dideteksi oleh model. mAP dapat dihitung dengan menggunakan persamaan berikut [24]:

$$mAP = \frac{1}{N} \sum_{i=1} AP_i \quad (2)$$

dimana N merupakan jumlah kelas dan AP adalah nilai *average precision* pada setiap kelas. AP dihitung dengan menggunakan persamaan berikut [24]:

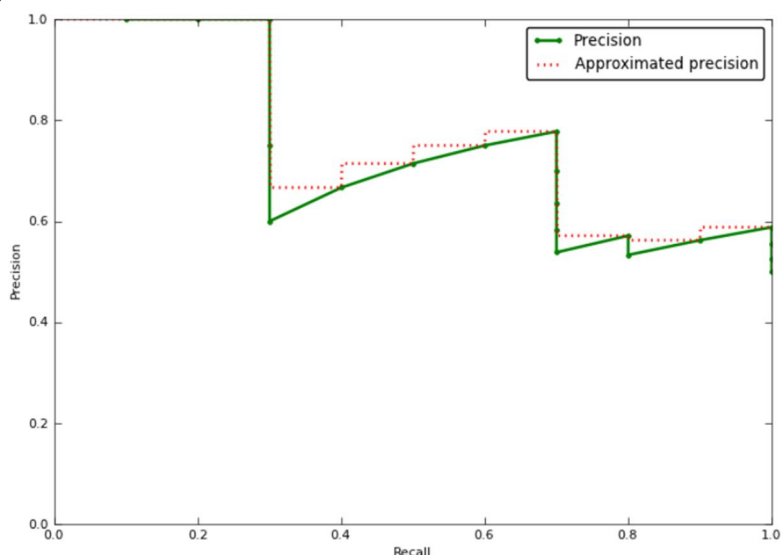
$$AP = \sum_n (R_n - R_{n-1}) P_n \quad (3)$$

dimana P_n dan R_n adalah nilai *precision* dan *recall* pada threshold ke-n. n merupakan *cut-off threshold*. Perhitungan *average precision* pada dasarnya menggunakan *interpolation* dengan 11 poin yang disebut dengan *threshold*. Berikut adalah ilustrasi dari ke sebelas poin tersebut.

Tabel 1. *Threshold proses interpolation* [24]

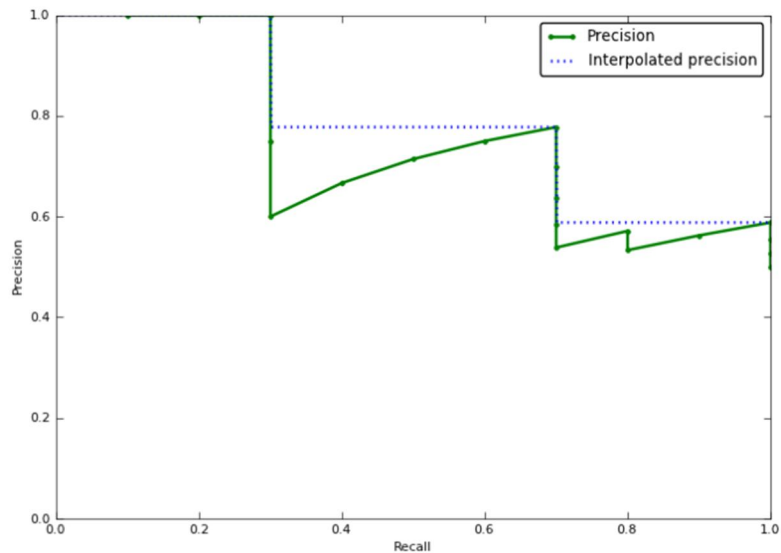
Retrieval cutoff	Precision	Recall
Threshold 1	100%	20%
Threshold 2	100%	40%
Threshold 3	66%	40%
Threshold 4	75%	60%
Threshold 5	60%	60%
Threshold 6	66%	80%
Threshold 7	57%	80%
Threshold 8	50%	80%
Threshold 9	44%	80%
Threshold 10	50%	100%

Dari tabel di atas selanjutnya dibuat grafik untuk menunjukkan posisi *precision* terhadap *recall* seperti gambar berikut:



Gambar 2. Grafik plotting point untuk mengukur AP [28]

Setelah mendapatkan grafik di atas, selanjutnya dilakukan *smoothing* sehingga didapatkan grafik sebagai berikut:



Gambar 3. Grafik interpolasi [28]

Grafik interpolasi atau hasil smoothing ditunjukkan oleh garis putus-putus warna biru. Setelah mendapatkan poin-poin utama hasil smoothing, selanjutnya dilakukan pengukuran AP dengan menggunakan rumus pada persamaan (3). Sebelum dihitung menggunakan persamaan (3), terlebih dahulu dibuat tabel agar lebih memudahkan dalam mengilustrasikan nilai *precision* dan recall untuk setiap *threshold*.

Tabel 2. Hasil interpolasi [28]

Retrieval cutoff	Precision	Recall
Threshold 1	100%	20%
Threshold 2	100%	40%
Threshold 3	66%	40%
Threshold 4	75%	60%
Threshold 5	60%	60%
Threshold 6	66%	80%

Dari hasil di atas selanjutnya dihitung AP menggunakan persamaan (4) sebagai berikut:

$$AP = (1*0,3)+(0,78*0)+(0,78*0,4)+(0,58*0)+(0,58*0,3)+(0,5*0) = \mathbf{0,786}$$

Sedangkan untuk evaluasi pada proses klasifikasi dilakukan dengan mengukur akurasi, recall dan presisi. Berikut adalah formula yang digunakan [28]:

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$presisi = \frac{TP}{TP + FP} \quad (5)$$

$$recall = \frac{TP}{TP + FN} \quad (6)$$

Hasil dan Pembahasan

Sebagaimana dijelaskan pada bagian sebelumnya, proses penelitian diawali dengan proses akuisisi data dan perbaikan citra. Selanjutnya dibagi dalam proporsi 80:20. 80% proporsi data digunakan sebagai data untuk pelatihan model. Selanjutnya, 20% proporsi data digunakan sebagai data testing atau data uji. Dalam proses pelatihan data menggunakan metode SSD, penelitian ini menerapkan berbagai macam skenario guna menghasilkan struktur deep learning terbaik dengan performa yang unggul. Berikut adalah rangkuman dari beberapa skenario pelatihan model.

Tabel 3. Rangkuman hasil pelatihan model dengan beberapa skenario

Skenario	Batch size	Step	mAP	Akurasi	Presisi	Recall
Skenario 1	8	20k	0,7418	0,6014	0,9610	0,6103
Skenario 2	16	20k	0,7691	0,7791	0,9593	0,7795
Skenario 3	32	10k	0,7611	0,8292	0,9628	0,8288
Skenario 4	32	20k	0,8716	0,9211	0,9832	0,9201
Skenario 5	64	10k	0,8813	0,9795	0,9985	0,9693
Skenario 6	64	15k	0,6598	0,7195	0,8502	0,7195
Skenario 7	100	15k	0,7114	0,7501	0,9698	0,7338
Skenario 8	100	20k	0,7013	0,6514	0,9301	0,6421

Sumber: hasil analisis

Berdasarkan hasil di atas, terlihat bahwa skenario 5 memiliki performa terbaik dibandingkan dengan skenario-skenario yang lain. Model hasil training dengan menggunakan struktur deep learning pada skenario 5 selanjutnya digunakan untuk tahap testing. Berikut adalah hasil testing dari beberapa kali percobaan.

Tabel 4. Hasil testing pada beberapa kali percobaan

No.	Pengujian model	Akurasi
1.	Testing - 1	0,8271
2.	Testing - 2	0,8266
3.	Testing - 3	0,5128
4.	Testing - 4	0,8103
5.	Testing - 5	0,8501
Rata-rata		0,8254

Sumber: hasil analisis

Rata-rata hasil pengujian menunjukkan bahwa performa deteksi sangat bagus. Dari analisis beban komputasi, SSD dengan struktur arsitektur pada skenario 5 menghasilkan estimated count of arithmetic sebesar 1.710 G. Selanjutnya hasil pelatihan menggunakan SSD dibandingkan dengan metode lain yakni YOLO V5 baik dari sisi akurasi serta dari sisi beban komputasi untuk dianalisis model deep learning mana yang paling unggul. Dalam percobaan pelatihan model menggunakan YOLO V5 dengan *batch size* 16, jumlah *epoch* 100 memperoleh akurasi sebesar

0,8197; presisi sebesar 0,9925; dan *recall* sebesar 0,8078. Pada saat jumlah *epoch* ditambah menjadi 150 terjadi peningkatan yakni akurasi mencapai 0,8981; presisi sebesar 0,9851; dan *recall* sebesar 0.8947. Sedangkan beban komputasi dari arsitektur ini adalah 5.393 G

Berdasarkan perbandingan antara SSD dengan YOLO V5, terlihat bahwa dari sisi performa pengujian tidak terdapat perbedaan yang signifikan. Namun dari ini beban komputasi terlihat bahwa YOLO V5 memiliki beban komputasi lebih dari 3 kali lipat beban komputasi SSD. Hal ini tentunya akan berdampak besar ketika model hasil pelatihan ditanamkan pada suatu sistem yang mana beban komputasi akan menyulitkan pengguna untuk mendapatkan informasi secara real-time. Oleh karena itu, SSD lebih dipertimbangkan untuk digunakan sebagai model untuk sistem deteksi sampah agar proses pemilahan sampah dapat diproses lebih cepat.

Kesimpulan

Penelitian ini bertujuan untuk mengembangkan model deteksi menggunakan arsitektur deep learning sebagai langkah awal pengembangan sistem deteksi sampah otomatis yang mana sistem tersebut nantinya sangat berguna pada proses pengelolaan sampah di industri. Ruang lingkup penelitian dibatasi pada deteksi sampah untuk industri produk kulit. Pengembangan model deteksi dilakukan dalam empat tahapan yakni tahap akuisisi data, tahap perbaikan citra, tahap pelatihan dan pengujian model, serta tahap evaluasi. Berdasarkan hasil penelitian dan analisis yang telah dilakukan, pengembangan model deteksi menggunakan arsitektur SSD memiliki performa yang sangat baik untuk mendeteksi sampah dengan tepat. Hal ini terlihat dari mAP yang diperoleh yakni sebesar 0,8813; akurasi sebesar 0,9795; presisi sebesar 0,9985; dan *recall* sebesar 0,9693. Dari sisi beban komputasi, arsitektur SSD juga memiliki beban komputasi yang relatif rendah sehingga sangat memungkinkan untuk dikembangkan menjadi *embedded system*. Kedepannya, penelitian ini dapat menjadi langkah awal dalam pengembangan sistem deteksi sampah otomatis dalam bidang industry untuk mempermudah industry dalam mengelola sampah yang muncul selama proses produksi berlangsung. Sehingga penanganan sampah menjadi lebih efektif dan efisien.

Daftar Pustaka

- [1] "Global Waste Problem," *Global Challenges*, 2022. <https://www.theworldcounts.com/challenges/planet-earth/waste/global-waste-problem> (accessed Sep. 22, 2022).
- [2] V. Muralidharan, S. Palanivel, and M. Balaraman, "Turning problem into possibility: A comprehensive review on leather solid waste intra-valorization attempts for leather processing," *J. Clean. Prod.*, vol. 367, p. 133021, 2022, doi: <https://doi.org/10.1016/j.jclepro.2022.133021>.
- [3] C. R. China, M. M. Maguta, S. S. Nyandoro, A. Hilonga, S. V. Kanth, and K. N. Njau, "Alternative tanning technologies and their suitability in curbing environmental pollution from the leather industry: A comprehensive review," *Chemosphere*, vol. 254, p. 126804, 2020, doi: <https://doi.org/10.1016/j.chemosphere.2020.126804>.
- [4] O. Omoloso, K. Mortimer, W. R. Wise, and L. Jraisat, "Sustainability research in the leather industry: A critical review of progress and opportunities for future research," *J. Clean.*

- Prod.*, vol. 285, p. 125441, 2021, doi: <https://doi.org/10.1016/j.jclepro.2020.125441>.
- [5] C. Bircanoğlu, M. Atay, F. Beşer, Ö. Genç, and M. A. Kızrak, "RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*, 2018, pp. 1–7. doi: [10.1109/INISTA.2018.8466276](https://doi.org/10.1109/INISTA.2018.8466276).
- [6] F. Shaikh, N. Kazi, F. Khan, and Z. Thakur, "Waste Profiling and Analysis using Machine Learning," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 488–492. doi: [10.1109/ICIRCA48905.2020.9183035](https://doi.org/10.1109/ICIRCA48905.2020.9183035).
- [7] A. Trubetskaya, W. Horan, P. Conheady, K. Stockil, S. Merritt, and S. Moore, "A methodology for assessing and monitoring risk in the industrial wastewater sector," *Water Resour. Ind.*, vol. 25, p. 100146, 2021, doi: <https://doi.org/10.1016/j.wri.2021.100146>.
- [8] M. Erkinay Ozdemir, Z. Ali, B. Subeshan, and E. Asmatulu, "Applying machine learning approach in recycling," *J. Mater. Cycles Waste Manag.*, vol. 23, no. 3, pp. 855–871, 2021, doi: [10.1007/s10163-021-01182-y](https://doi.org/10.1007/s10163-021-01182-y).
- [9] S. Neelakandan *et al.*, "Metaheuristics with Deep Transfer Learning Enabled Detection and classification model for industrial waste management," *Chemosphere*, vol. 308, p. 136046, 2022, doi: <https://doi.org/10.1016/j.chemosphere.2022.136046>.
- [10] N. Intalar, K. Chumnumporn, C. Jeenanunta, and A. Tunpan, "Towards Industry 4.0: Digital transformation of traditional safety shoes manufacturer in Thailand with a development of production tracking system," *Eng. Manag. Prod. Serv.*, vol. 13, no. 4, pp. 79–94, 2021, doi: [10.2478/emj-2021-0033](https://doi.org/10.2478/emj-2021-0033).
- [11] M. W. Rahman, R. Islam, A. Hasan, N. I. Bithi, M. M. Hasan, and M. M. Rahman, "Intelligent waste management system using deep learning with IoT," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2072–2087, 2022, doi: <https://doi.org/10.1016/j.jksuci.2020.08.016>.
- [12] S. Majchrowska *et al.*, "Deep learning-based waste detection in natural and urban environments," *Waste Manag.*, vol. 138, pp. 274–284, 2022, doi: <https://doi.org/10.1016/j.wasman.2021.12.001>.
- [13] Q. Zhang *et al.*, "A multi-label waste detection model based on transfer learning," *Resour. Conserv. Recycl.*, vol. 181, p. 106235, 2022, doi: <https://doi.org/10.1016/j.resconrec.2022.106235>.
- [14] L. Dang, T. Cao, and Y. Hoshino, "Classification of Metal Objects Using Deep Neural Networks in Waste Processing Line," *Int. J. Innov. Comput. Inf. Control IJICIC*, vol. 15, pp. 1901–1912, Oct. 2019, doi: [10.24507/ijicic.15.05.1901](https://doi.org/10.24507/ijicic.15.05.1901).
- [15] M. Yurtsever and U. Yurtsever, "Use of a convolutional neural network for the classification of microbeads in urban wastewater," *Chemosphere*, vol. 216, pp. 271–280, 2019, doi: <https://doi.org/10.1016/j.chemosphere.2018.10.084>.
- [16] W.-L. Mao, W.-C. Chen, H. I. K. Fathurrahman, and Y.-H. Lin, "Deep learning networks for real-time regional domestic waste detection," *J. Clean. Prod.*, vol. 344, p. 131096, 2022, doi: <https://doi.org/10.1016/j.jclepro.2022.131096>.
- [17] Y. Ku, J. Yang, H. Fang, W. Xiao, and J. Zhuang, "Deep learning of grasping detection for a robot used in sorting construction and demolition waste," *J. Mater. Cycles Waste Manag.*, vol. 23, no. 1, pp. 84–95, 2021, doi: [10.1007/s10163-020-01098-z](https://doi.org/10.1007/s10163-020-01098-z).
- [18] W. Xiao, J. Yang, H. Fang, J. Zhuang, Y. Ku, and X. Zhang, "Development of an automatic

- sorting robot for construction and demolition waste," *Clean Technol. Environ. Policy*, vol. 22, no. 9, pp. 1829–1841, 2020, doi: 10.1007/s10098-020-01922-y.
- [19] J.-I. Watanabe, Y. Shao, and N. Miura, "Underwater and airborne monitoring of marine ecosystems and debris," *J. Appl. Remote Sens.*, vol. 13, no. 4, p. 44509, Oct. 2019, doi: 10.1117/1.JRS.13.044509.
- [20] S. Kong, M. Tian, C. Qiu, Z. Wu, and J. Yu, "IWSCR: An Intelligent Water Surface Cleaner Robot for Collecting Floating Garbage," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 10, pp. 6358–6368, 2021, doi: 10.1109/TSMC.2019.2961687.
- [21] "TrashNet," *Waste Dataset*, 2016. <https://github.com/garythung/trashnet> (accessed Aug. 30, 2022).
- [22] W. Ziang, "Waste Picture," *Dataset*, 2018. <https://www.kaggle.com/datasets/wangziang/waste-pictures> (accessed Aug. 30, 2022).
- [23] X. LEI, H. WANG, J. I. E. SHEN, Z. H. E. CHEN, and W. ZHANG, "A novel intelligent underwater image enhancement method via color correction and contrast stretching," *Microprocess. Microsyst.*, p. 104040, 2021, doi: <https://doi.org/10.1016/j.micpro.2021.104040>.
- [24] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector BT - Computer Vision – ECCV 2016," in *Lecture Notes in Computer Science*, 2016, pp. 21–37.
- [25] S. Amemiya, H. Takao, S. Kato, H. Yamashita, N. Sakamoto, and O. Abe, "Automatic detection of brain metastases on contrast-enhanced CT with deep-learning feature-fused single-shot detectors," *Eur. J. Radiol.*, vol. 136, p. 109577, 2021, doi: <https://doi.org/10.1016/j.ejrad.2021.109577>.
- [26] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, and J. Wu, "Feature-fused SSD: fast detection for small objects," in *Proc.SPIE*, Apr. 2018, vol. 10615, p. 106151E. doi: 10.1117/12.2304811.
- [27] Y. Liu *et al.*, "A deep convolutional neural network-based automatic delineation strategy for multiple brain metastases stereotactic radiosurgery," *PLoS One*, vol. 12, no. 10, pp. 1–17, 2017, doi: 10.1371/journal.pone.0185844.
- [28] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *Lect. Notes Comput. Sci.*, 2016.