Print ISSN: 2461-1409
Journal Sensi Online ISSN: 2655-5298

Research Article Open Access (CC–BY-SA)

Deep Learning Pada Detektor Jerawat: Model YOLOv5

Hendra Kusumah[1], Muhammad Suzaki Zahran[*,2], Kadek Naufal Rifqi[3], Devi Alawiyah Putri[4], Ety Meina Wakti Hapsari[5]

[1]Program Studi Magister Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Raharja Tangerang
[2]Program Studi Sistem Komputer, Fakultas Sains dan Teknologi, Universitas Raharja Tangerang
[3]Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Raharja Tangerang
[4,5]Program Studi Manajemen Informatika, Fakultas Sains dan Teknologi, Universitas Raharja Tangerang
e-mail: [1]hendra.kusumah@raharja.info , [*,2] m.suzaki@raharja.info , [3]kadek.naufal@raharja.info , [4]devi.alawiyah@raharja.info , [5]ety.meina@raharja.info

Abstrak

Jerawat (Acne Vulgaris) merupakan masalah utama yang sulit untuk dihindari pada masyarakat daerah perkotaan, seperti Jakarta dan sekitarnya. Penyebab utama dari jerawat yaitu tingginya polusi udara yang disebabkan oleh hasil pembakaran transportasi dan sektor industri. Sisa pembakaran ini umumnya mengandung PM (Particulate Matter) dengan ukuran yang cukup kecil (PM2.5 dan PM10) yang mampu masuk ke dalam kulit melalui pori-pori dan bereaksi dengan beberapa senyawa diudara sehingga menyebabkan banyak permasalahan kulit lainnya. Penelitian ini berfokus pada pendeteksian jerawat dengan menggunakan model Deep Learning, yaitu YOLOv5. YOLOv5 dilatih dengan menggunakan tiga optimizer berbeda (SGD, Adam, dan AdamW) sebanyak 100 epochs. Setelah dilakukan pelatihan, didapatkan hasil F1-score dengan optimizer SGD sebesar 43%, Adam 39%, dan AdamW sebesar 40%. Pada penelitian ini, optimizer SGD memiliki nilai F1 tertinggi sehingga dijadikan sebagai optimizer teroptimum yang dapat digunakan pada permasalahan di penelitian ini.

Kata kunci—Jerawat, *Acne Vulgaris*, PM2.5, YOLOv5, *Deep Learning*

Abstract

Acne (Acne Vulgaris) is a major problem that is difficult to avoid in urban areas, such as Jakarta and its surroundings. The main cause of acne is the high air pollution caused by the combustion of the transportation and industrial sectors. This combustion residue generally contains PM (Particulate Matter) with a fairly small size (PM2.5 and PM10) which is able to enter the skin through the pores and react with several compounds in the air, causing many other skin problems. This study focuses on detecting acne using the Deep Learning model, namely YOLOv5. YOLOv5 was trained using three different optimizers (SGD, Adam, and AdamW) for 100 epochs. After the training, the F1-score with the optimizer SGD was 43%, Adam was 39%, and AdamW was 40%. In this study, the SGD optimizer has the highest F1 value so that it is used as the optimal optimizer that can be used in the problems in this study.

Keywords—*Acne, Acne Vulgaris*, PM2.5, *Deep Learning*

1. PENDAHULUAN

Tingginya tingkat produktivitas industri Indonesia di berbagai sektor menimbulkan berbagai produk dengan pengembangan dan inovasi baru bermunculan. Selaras dengan peningkatan produktivitas ini, akumulasi limbah yang dibuang ke lingkungan juga terus bertambah. Polutan udara merupakan satu dari beberapa limbah yang dihasilkan oleh peningkatan produktivitas industri ini, satu diantaranya yaitu polutan PM2.5 dan PM10. PM (Particulate Matter) merupakan partikel-partikel kecil yang ada di udara yang berasal dari sisa pembuangan dan pembakaran. Sebanyak kurang lebih 43%-46% PM2.5 dan PM10 berasal dari buangan transportasi dan pembakaran industri [1]. Rata-rata konsentrasi polutan PM2.5 Indonesia pada tahun 2021 bernilai sebesar 6.9 kali lebih tinggi dari panduan kualitas udara WHO sehingga Indonesia menjadi negara dengan kualitas udara terburuk ke-17 di dunia [2]. Selain itu, Indonesia mencapai kasus konsentrasi PM2.5 tertinggi dengan konsentrasi PM2.5 sebesar 34.3 mikron/m3 [3]. Dengan ukurannya yang cukup kecil, yaitu kurang dari 2.5 mikrometer, PM2.5 memungkinkan untuk masuk ke dalam kulit melalui pori-pori atau celah dari rambut halus di kulit karena kulit merupakan organ pertama yangn akan menjadi kontak langsung dengan lingkungan luar [4]. Selain itu, PM2.5 mampu bereaksi dengan beberapa senyawa, seperti Nitrogen Oksida (NOx), Sulfur Dioksida (SO2), dan Karbon Monoksida (CO), sehingga membentuk *Aryl Hydrocarbon Receptor* (AhR) yang menjadi penyebab utama permasalahan kulit, seperti penuaan, kerutan, perubahan pigmentasi kulit, jerawat, hingga kanker. Diketahui tiap kenaikan 10 µg/m3 PM2.5 akan menambah sekitar 1,71% pasien berjerawat dalam kurun waktu 0-7 hari dengan rasio pasien berumur di atas 25 tahun lebih tinggi dibanding di bawah 25 tahun, baik pria maupun wanita. Hal ini menjadikan jerawat masalah utama yang sering dialami oleh banyak orang [5]. Pada penelitian kali ini, akan membahas mengenai deteksi jerawat dengan menggunakan model Deep Learning [6]–[9].

2. METODE PENELITIAN

2.1 Dataset

Dataset yang kami gunakan berasal dari website resmi Dermnetnz [10], yaitu sebanyak 243 gambar berisikan jerawat pada area facial. Selanjutnya gambar-gambar ini kami proses menggunakan platform Roboflow [11] sehingga kami dapat menandai bagian-bagian spesifik, yaitu bagian yang mengandung jerawat, dengan menggunakan Bounding-Boxes [12]. Dengan menggunakan Bounding-Boxes ini kami dapat melatih model yang kami buat agar dapat lebih mengefisiensikan proses pelatihan.

Gambar 1 Dataset original

Gambar 2 Dataset setelah Bounding-Box

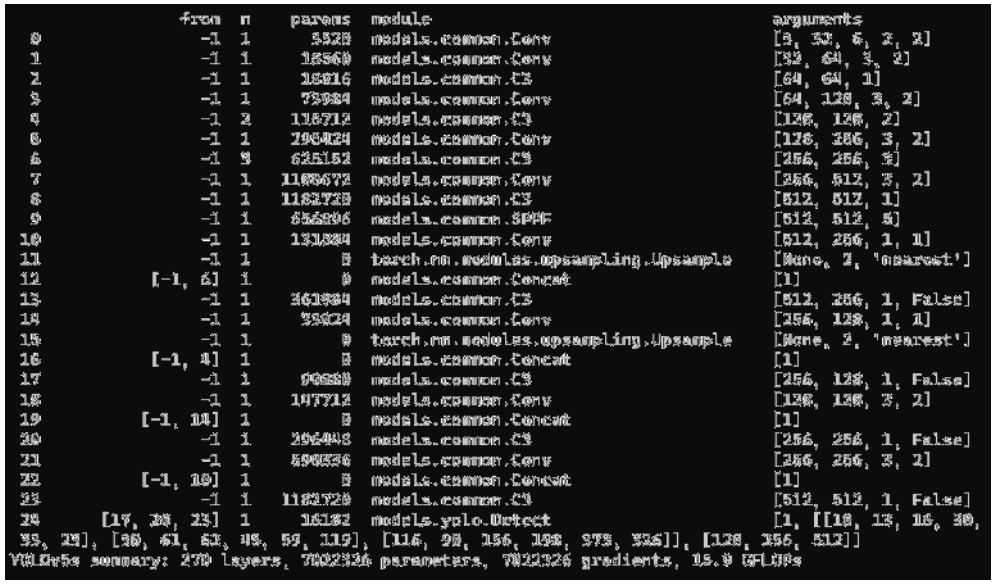
2. 2 Yolo

Yolo merupakan satu dari beberapa arsitektur Deep Learning yang berfokus pada Object Recognition (pengenalan obyek) dengan memanfaatkan metode bounding boxes. Yolo pertama kali diperkenalkan oleh Joseph Redmon, dkk., pada tahun 2015 dalam papernya "you only look once: unified, real-time object detection" [13]. Pada dasarnya YOLO menggunakan arsitektur R-CNN (Regional Convolutional Neural Network) dengan memanfaatkan metode bounding boxes. Metode ini akan melokalisasi serta mengklasifikasi bagian-bagian gambar yang dipilih secara acak pada gambar. Bagian-bagian yang memiliki nilai-nilai tertinggi pada saat lokalisasi akan disimpan sebagai hasil deteksi.

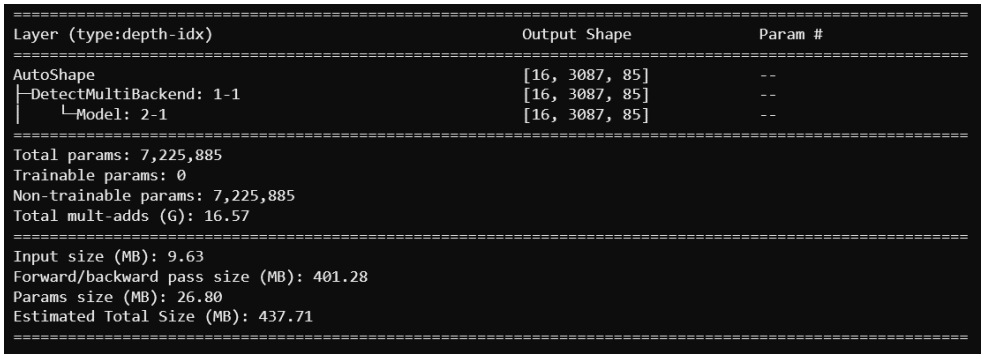
Pada dasarnya YOLO memanfaatkan single neural network dalam arsitekturnya yang digunakan untuk menggabungkan proses klasifikasi yang pada model-model lainnya dilakukan secara terpisah. Pendekatan ini menghasilkan algoritma pengenalan obyek lebih cepat dari model-model lainnya sehingga YOLO mampu mengalahkan model-model sebelumnya [8], [14], [15]. Redmon, dkk., berhasil mengembangkan YOLO hingga ke versi YOLOv3 [7]dengan kemampuan pengenalan obyek realtime yang sangat cepat dan juga ringan untuk dijalankan. Namun, karena diketahui modelnya digunakan oleh pihak-pihak yang kurang bertanggung jawab hingga ke ranah militer, Redmon, dkk., memutuskan untuk tidak mengembangkan YOLO serta memberhentikan riset yang berhubungan dengan Computer Vision guna mencegah disalahgunakannya hasil penelitian tersebut [16].

Beberapa peneliti tetap mengembangkan model YOLO hingga saat ini banyak variasi-variasi baru YOLO, seperti YOLOv4 [9] yang dikembangkan oleh Alexey Bochkovskiy, dkk., yang berfokus pada pengembangan fitur-fitur baru yang dicoba untuk diterapkan pada YOLO. Selain itu, YOLOv5 [17] juga berhasil dikembangkan oleh Glenn Jocher pada tahun 2020 dengan menggunakan framework yang berbeda, yaitu framework pytorch.

Penelitian ini berfokus pada penggunaan YOLOv5 dengan jenis model YOLOv5S (YOLOv5 – Small) yang ditujukan secara khusus untuk kategori jerawat.Secara umum, YOLOv5 memiliki arstektur seperti gambar di atas. Namun dikarenakan sumber daya yang digunakan untuk penelitian memiliki keterbatasan, peneliti memilih untuk menggunakan YOLOv5s karena memiliki struktur yang lebih sederhana. Pada penelitian ini, peneliti menggunakan Laptop dengan konfigurasi prosesor AMD Ryzen 9 5900HS, RAM 16 GB, dan pemrosesan grafis NVIDIA GeForce RTX 3050 dengan dukungan CUDA. Selain itu, konfigurasi yang digunakan pada model secara umum, yaitu image-size 618x618, batch-size 16, 100 epochs, patience 100.



Gambar 3 Layer YOLOv5s



Gambar 4 Arsitektur YOLOv5s

2. 3 Algoritma Optimisasi

Tujuan utama dari pembuatan suatu model machine learning yaitu untuk mendapatkan suatu model yang teroptimisasi dengan baik sehingga mampu menghadapi permasalahan yang dihadapi, baik pernah diselesaikan maupun permasalahan-permasalahan baru yang belum terselesaikan. Hal ini mampu dicapai dengan memanfaatkan algoritma optimisasi. Setidaknya ada dua cara yang dapat digunakan untuk mencapai model optimum, yaitu dengan menggunakan pendekatan empirical risk minimization [18], [19] dan metode gradient descent [20], [21]. Pada penelitian ini, peneliti menggunakan pendekatan gradient descent untuk mengoptimisasi model yang dilatih. Metode gradient descent bertindak sebagai barometer pada proses pelatihan model machine learning dengan tujuan untuk meningkatkan akurasi dengan membarui parameter yang digunakan setiap kali melakukan iterasi [22]. Ada dua komponen utama penyusun gradient descent, yaitu learning rate [23], [24] dan cost function [25]–[28]. Learning rate (Gradient Descent Step) merupakan langkah yang digunakan untuk mencapai titik minimum (local minimum maupun global minimun). Umumnya learning rate bernilai cukup kecil, namun nilainya harus disesuaikan agar mampu mencapai tujuan yang diinginkan. Jika nilai learning rate terlalu besar, maka model akan kesulitan untuk mencapai nilai minimum

karena titik yang dituju bisa saja terlewatkan, sebaliknya jika terlalu kecil, kemungkinan model kesulitan dalam mencari titik terendah tersebut karena hanya mengalami sedikit perubahan serta memori yang dibutuhkan untuk melakukan pelatihan akan menjadi sangat besar.

Selain itu, cost function (dikenal juga loss function) digunakan untuk mengukur difference atau error antara nilai sebenarnya dari y dengan nilai prediksi model. Pengukuran ini dilakukan untuk memberikan feedback kepada model sehingga parameter yang digunakan dapat disesuaikan dengan cara meminimalisir nilai error tersebut. Pada penelitian ini, peneliti menggunakan 3 metode gradient descent yang digunakan untuk mengoptimisasi model yang dilatih.

2. 3. 1 Stochastic Gradient Descent (SGD)

ada gradient descent [29], formula yang digunakan menggunakan metode sum untuk menghitung jumlah residual. Residual merupakan selisih antara label sebenarnya dengan label prediksi. Tujuan perhitungan nilai residual ini yaitu untuk tau seberapa besar error yang didapatkan, sehingga tingkat error ini dapat diminimalisir dengan optimum.

$$Sum\ of\ squared\ residuals\ = \sum_{n=1}^{m} (y - \hat{y})^2 \tag{1}$$

Nilai error ini dapat diminimisasi dengan menggunakan gradiennya. Oleh sebab itu diperlukan derivasi dari formula sum of squared residuals ini. Namun, untuk permasalahan optimisasi yang datanya sangat besar (big data), formula ini dapat dikatakan sebagai high cost karena memerlukan banyak memori untuk menyimpan data dan juga menghitung jumlah semua data sehingga menjadi kurang efisien. Permasalahan ini dapat ditanggulangi dengan menggunakan nilai acak (stokastik) pada data yang dimiliki.

$$w^{(r+1)} = w^r - \alpha \nabla f_{i_r}(w^r) \tag{2}$$

Pemilihan nilai acak [30], [31] ini dapat mengurangi kebutuhan memori dari proses penjumlahan sebelumnya. Pada (2), proses dilakukan untuk mengupdate kemiringan baru. Proses ini dilakukan berulangkali secara iteratif seperti gradient descent pada umumnya sehingga mampu ditemukan titik optimum dari model yang dilatih tersebut[32]. Walaupun dapat dikatakan lebih cepat dan minim kebutuhan memori, SGD tetap memiliki konsekuansi dari pengurangan kompleksitas ini, yaitu adanya non-zero gradient noise[29].

2. 3. 2 Adaptive Moment Estimation (Adam)

Adam [33], [34] merupakan algoritma gabungan dari algoritma RMSProp dan Momentum dengan menyimpan learning rate RMSProp dan weight dari rerata momentum [35], [36].

$$m_i = \beta_1 m_i + (1 - \beta_1) \frac{\partial L}{\partial \theta_i} \tag{3}$$

$$v_i = \beta_2 v_i + (1 - \beta_2) \left(\frac{\partial L}{\partial \theta_i}\right)^2 \tag{4}$$

Parameter-parameter (3) dan (4) akan mengalami bias saat mendekati 1, terlebih jika inisialisasi time steps dan decay rates yang sangat kecil. Oleh karena itu, dibutuhkan bias-

corrected dan moment estimate dengan membagi parameter (3) dan (4) dengan selisih antara 1 dengan decay factor.

$$\hat{m} = \frac{m_i}{1 - \beta_1} \quad (5)$$

$$\hat{v} = \frac{v_i}{1 - \beta_2} \quad (6)$$

Author dari Adam itu sendiri menyarankan nilai untuk beta-1 adalah 0.9, beta-2 0.999, dan untuk epsilon yaitu 10^-8. Setelah di dapatkan nilai optimum dari parameter (3) dan (4), formula dari Adam dapat dikalkulasi sebagai berikut:

$$\theta_i = \theta_i - \frac{\alpha}{\sqrt{\hat{v} + \epsilon}} \hat{m} \quad (7)$$

Berdasarkan formula dari Adam tersebut, diketahui Adam menggunakan basis RMSProp tetapi dengan pengestimasian gradien melalui metode momentum. Tujuan dari kombinasi ini yaitu untuk meningkatkan kecepatan pelatihan. Dengan metode ini Adam berhasil mengalahkan algoritma-algoritma optimisasi sebelumnya pada tahap pelatihan ataupun pada eksperimen-eksperimen lainnya. Namun, Adam menggunakan hyperparameter baru yang dapat mempersulit hyperparameter tuning ketika permasalahan yang dihadapi semakin kompleks.

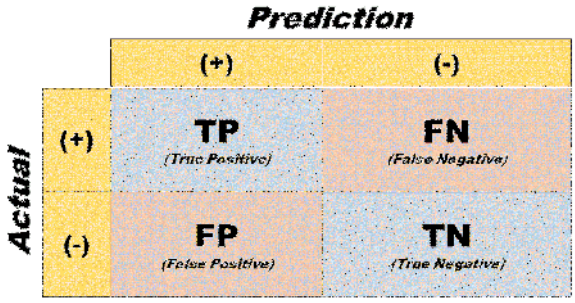
2. 3. 3 Adaptive Moment Estimation Weight Decay (AdamW)

AdamW [37], [38] merupakan metode optimisasi stokastik yang memodifikasi fungsi spesifik pada Adam dengan cara memisahkan weight decay (regularization) dari gradient update-nya. Weight decay (regularization) merupakan teknik regularisasi yang diterapkan kepada weight neural network untuk mencegah adanya overfitting.

$$\theta_{t+1,i} = \theta_{t,i} - \eta \left(\frac{1}{\sqrt{\hat{v}_t + \epsilon}} . \hat{m}_t + w_{t,i} \theta_{t,i} \right), \forall t \quad (8)$$

2. 4. Evaluasi

Evaluasi pada penelitian ini menggunakan confusion matrix[39]. Confusion matrix merupakan matriks yang ditujukan untuk mengukur seberapa baik suatu model klasifikasi yang telah dilatih. Komponen utama dari confusion matrix yaitu true positif (TP), false positif (FP), true negative (TN), dan false negative (FN).



Gambar 5 Confusion Matrix

Pada confusion matrix, dapat dilihat beberapa nilai yang dapat mengukur seberapa baik suatu model yang dihasilkan, yaitu dengan menggunakan accuracy, recall, precision, dan F1-Score [40]–[42].

$$Accuracy = \frac{TP + TN}{Total\ Prediksi} \quad (9)$$

Nilai accuracy mengukur seberapa benar model memprediksikan klasifikasi suatu label terhadap jumlah dari data yang digunakan saat prediksi.

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

Recall mengukur seberapa baik model dengan melihat berapa banyak klasifikasi yang benar dan yang benar dianggap salah.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

Sedangkan precision mengukur model dengan membandingkan jumlah klasifikasi yang benar sesuai dengan ground truth dengan jumlah prediksi benar.

$$F1 - Score = 2\left(\frac{Precision * Recall}{Precision + Recall}\right) \quad (12)$$

Setelah didapatkan nilai recall dan precision, perbandingan antar keduanya dapat dibandingkan dengan menggunakan F1-score yang merata-ratakan harmoniknya.

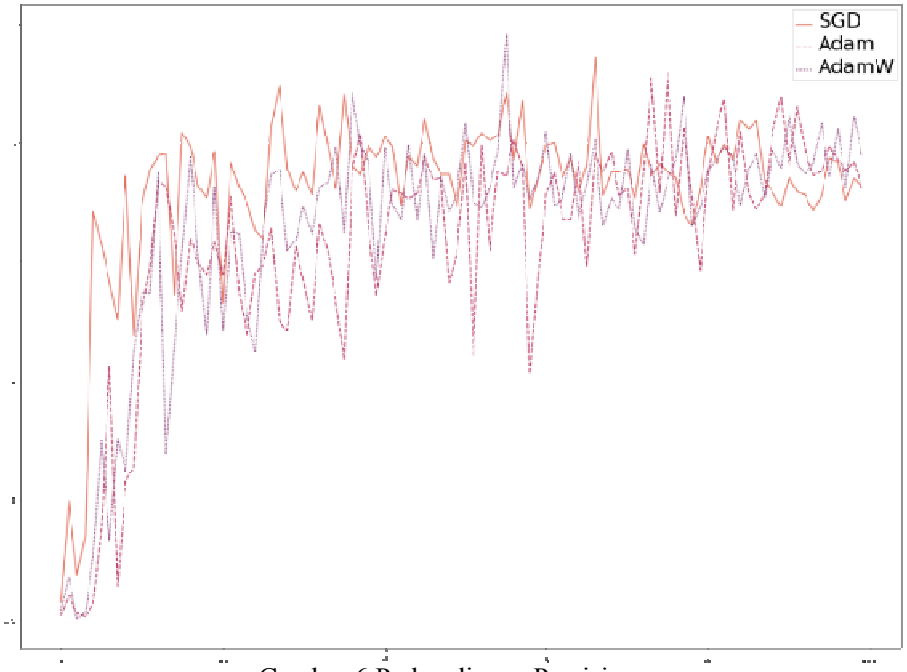
3. HASIL DAN PEMBAHASAN

Pada penelitian ini, model YOLOv5 yang digunakan merupakan model YOLOv5s, yaitu model YOLOv5 versi small. Penggunaan YOLOv5s ini dikarenakan adanya keterbatasan sumber daya komputer peneliti. Namun, penggunaan YOLOv5s ini tidak mengganggu aktivitas penelitian, hanya saja arsitektur model yang digunakan lebih sederhana daripada model YOLOv5 versi lainnya. Ukuran gambar yang kami gunakan memiliki ukuran 618x618 pixel, dengan batch size sebesar 16. Iterasi (epochs) yang diterapkan sebanyak 100 epochs, dengan menggunakan model CUDA sehingga memanfaatkan kemampuan pemrosesan grafis. Setelah dilakukan pelatihan sebanyak 100 epochs pada model dengan menggunakan 3 jenis optimizer berbeda (SGD, Adam, dan AdamW), didapatkan bahwa rerata precision dan recall:

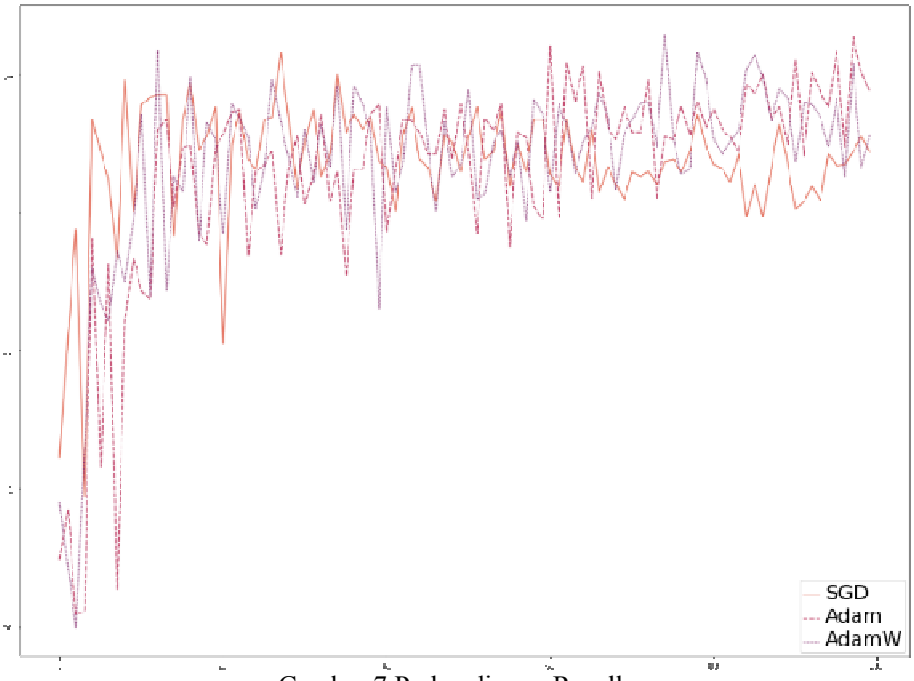
Tabel 1 Perbandingan Hasil Training YOLOv5s

Model	Optimizer	Precision	Recall
YOLOv5s	SGD	36,078 %	33,566 %
	Adam	31,452 %	32,839 %
	AdamW	32,802 %	33,573 %

Rerata tertinggi precision diraih oleh model dengan optimizer SGD, namun jika dilihat dari rerata recall, AdamW memiliki nilai tertinggi dibandingkan dengan optimizer lainnya. Nilai tertinggi dari precision dan recall yang didapatkan yaitu sebesar 49,079% dan 42,883% yang didapatkan dengan menggunakan optimizer AdamW.

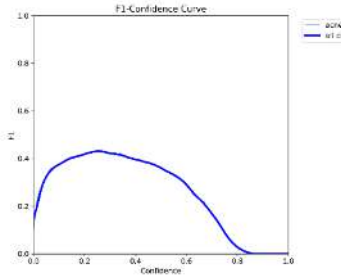


Gambar 6 Perbandingan Precision

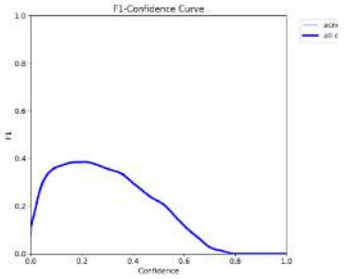


Gambar 7 Perbandingan Recall

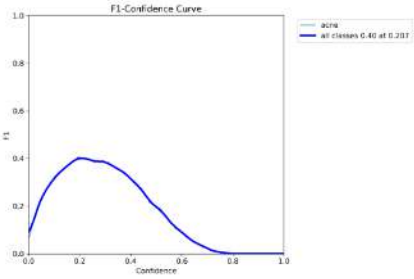
Untuk dapat membandingkan tiap optimizer, dilakukan perhitungan dengan menggunakan F1-Score pada tiap model dengan optimizer tersebut. Hasil dari perhitungan F1-Score yaitu:



Gambar 8 Grafik F1-Confidence SGD



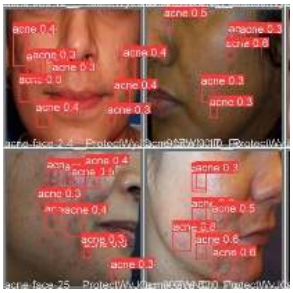
Gambar 9 Grafik F1-Confidence Adam



Gambar 10 Grafik F1-Confidence AdamW

Terlihat dari masing-masing grafik, SGD memiliki F1-Score tertinggi dengan nilai 43%. Dapat dipastikan bahwa optimizer SGD lebih baik digunakan dibandingkan dengan optimizer Adam dan AdamW. Selain itu, AdamW lebih baik untuk digunakan dibandingkan dengan Adam walaupun nilai keduanya tidak berbeda jauh secara signifikan. Beberapa contoh hasil rekognisi dari pelatihan dapat dilihat pada gambar di bawah ini.

Gambar 11 Hasil deteksi dengan optimizer SGD



Gambar 12 Hasil deteksi dengan optimizer Adam

Gambar 13 Hasil deteksi dengan optimizer AdamW

4. KESIMPULAN

Pemanfaatan teknik bounding-box dirasa membantu dalam proses pelatihan model yang digunakan. Model YOLOv5s yang dilatih dengan menggunakan iterasi sebanyak 100 epochs dan memanfaatkan teknologi CUDA menghasilkan nilai yang menjanjikan. Terlihat dalam analisis, dengan menggunakan tiga optimizer yang berbeda, model tersebut mampu mendeteksi jerawat yang ada pada wajah di gambar yang diberikan. Dari ketiga optimizer yang digunakan, terlihat SGD (stochastic gradient descent) memiliki nilai F1 tertinggi, yaitu 43%, dibandingkan dengan Adam dan AdamW yang hanya mendapatkan nilai 39% dan 40%. Dari penilaian F1 ini, SGD dapat dikatakan sebagai optimizer teroptimum yang memungkinkan untuk digunakan pada permasalahan kali ini. Kedepannya, penelitian ini diharapkan mampu mengenali jerawat dengan lebih optimum. Selain itu, mampu dilatih dan diterapkan pada mobile maupun edge devices.

DAFTAR PUSTAKA

[1] P. Lestari, M. K. Arrohman, S. Damayanti, and Z. Klimont, “Emissions and spatial distribution of air pollutants from anthropogenic sources in Jakarta,” *Atmos Pollut Res*, vol. 13, no. 9, p. 101521, Sep. 2022, doi: 10.1016/J.APR.2022.101521.

[2] “Informasi Indeks Kualitas Udara (AQI) dan Polusi Udara di Indonesia | IQAir.” https://www.iqair.com/id/indonesia (accessed Sep. 01, 2022).

[3] "Konsentrasi PM2,5 di Jakarta Membahayakan - Kompas.id." https://www.kompas.id/baca/humaniora/2022/06/18/konsentrasi-pm25-di-jakarta-membahayakan (accessed Sep. 01, 2022).
[4] R. D. Pamela, "Jakarta's Air Pollution, and the New Emerging Cause for Skin Aging," *Journal of Dermatology and Skin Science*, vol. 2, no. 1, 2020.
[5] X. Li *et al.*, "The relationship between short-term PM2.5 exposure and outpatient visits for acne vulgaris in Chongqing, China: a time-series study," *Environmental Science and Pollution Research 2022 29:40*, vol. 29, no. 40, pp. 61502–61511, Apr. 2022, doi: 10.1007/S11356-022-20236-8.
[6] Q. Aini, N. Lutfiani, H. Kusumah, and M. S. Zahran, "Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo," *CESS (Journal of Computer Engineering, System and Science)*, vol. 6, no. 2, p. 192, 2021, doi: 10.24114/cess.v6i2.25840.
[7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767
[8] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, [Online]. Available: http://arxiv.org/abs/2004.10934
[10] "Acne (face) images | DermNet." https://dermnetnz.org/topics/acne-face-images (accessed Sep. 03, 2022).
[11] "Roboflow: Give your software the power to see objects in images and video." https://roboflow.com/ (accessed Sep. 03, 2022).
[12] Y. Xu *et al.*, "Gliding Vertex on the Horizontal Bounding Box for Multi-Oriented Object Detection," *IEEE Trans Pattern Anal Mach Intell*, vol. 43, no. 4, pp. 1452–1459, Apr. 2021, doi: 10.1109/TPAMI.2020.2974745.
[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
[14] "Q1 - Developing smart covid19 social distancing surveillance drone using yolo.pdf."
[15] M. Simon *et al.*, "Complexer-YOLO: Real-time 3D object detection and tracking on semantic point clouds," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2019-June, pp. 1190–1199, 2019, doi: 10.1109/CVPRW.2019.00158.
[16] "Joseph Redmon on Twitter: '"We shouldn't have to think about the societal impact of our work because it's hard and other people can do it for us" is a really bad argument.' / Twitter." https://twitter.com/pjreddie/status/1230523827446091776?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1230524770350817280%7Ctwgr%5E33cf8189af3836f97c440610106fc8fcec348dd4%7Ctwcon%5Es2_&ref_url=https%3A%2F%2Fsyncedreview.com%2F2020%2F02%2F24%2Fyolo-creator-says-he-stopped-cv-research-due-to-ethical-concerns%2F (accessed Sep. 03, 2022).
[17] "YOLOv5 Documentation." https://docs.ultralytics.com/ (accessed Sep. 03, 2022).
[18] Y. Kusunoki, J. Błaszczyński, M. Inuiguchi, and R. Słowiński, "Empirical risk minimization for dominance-based rough set approaches," *Inf Sci (N Y)*, vol. 567, pp. 395–417, Aug. 2021, doi: 10.1016/J.INS.2021.02.043.
[19] A. Montanari, B. Saeed, B. Edu, P.-L. Loh, and M. Raginsky, "Universality of empirical risk minimization," *Proceedings of Machine Learning Research*, vol. 178. PMLR, pp. 4310–4312, Jun. 28, 2022. Accessed: Sep. 03, 2022. [Online]. Available: https://proceedings.mlr.press/v178/montanari22a.html
[20] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu, "Conflict-Averse Gradient Descent for Multi-task learning," *Adv Neural Inf Process Syst*, vol. 34, pp. 18878–18890, Dec. 2021, Accessed: Sep. 03, 2022. [Online]. Available: https://github.com/Cranial-XIX/CAGrad.
[21] S. H. Haji and A. M. Abdulazeez, "COMPARISON OF OPTIMIZATION TECHNIQUES BASED ON GRADIENT DESCENT ALGORITHM: A REVIEW," *PalArch's Journal of Archaeology of Egypt / Egyptology*, vol. 18, no. 4, pp. 2715–2743, Feb. 2021, Accessed: Sep. 03, 2022. [Online]. Available: https://www.archives.palarch.nl/index.php/jae/article/view/6705

[22] "What is Gradient Descent? | IBM." https://www.ibm.com/cloud/learn/gradient-descent (accessed Sep. 03, 2022).
[23] S. Tang, Y. Zhu, and S. Yuan, "An improved convolutional neural network with an adaptable learning rate towards multi-signal fault diagnosis of hydraulic piston pump," *Advanced Engineering Informatics*, vol. 50, p. 101406, Oct. 2021, doi: 10.1016/J.AEI.2021.101406.
[24] N. Loizou, S. Vaswani, I. Laradji, S. Lacoste-Julien, and M. Diro, "Stochastic Polyak Step-size for SGD: An Adaptive Learning Rate for Fast Convergence," vol. 130. PMLR, pp. 1306–1314, Mar. 18, 2021. Accessed: Sep. 03, 2022. [Online]. Available: https://proceedings.mlr.press/v130/loizou21a.html
[25] J. Wang, P. Chen, N. Zheng, B. Chen, J. C. Principe, and F. Y. Wang, "Associations between MSE and SSIM as cost functions in linear decomposition with application to bit allocation for sparse coding," *Neurocomputing*, vol. 422, pp. 139–149, Jan. 2021, doi: 10.1016/J.NEUCOM.2020.10.018.
[26] N. Loka, I. Couckuyt, F. Garbuglia, D. Spina, I. van Nieuwenhuyse, and T. Dhaene, "Bi-objective Bayesian optimization of engineering problems with cheap and expensive cost functions," *Engineering with Computers 2022*, pp. 1–11, Jan. 2022, doi: 10.1007/S00366-021-01573-7.
[27] X. Luo, "Novel iterative ensemble smoothers derived from a class of generalized cost functions," *Comput Geosci*, vol. 25, no. 3, pp. 1159–1189, Jun. 2021, doi: 10.1007/S10596-021-10046-1/FIGURES/12.
[28] P. Wilmott, "Machine Learning," *Machine Learning and the City*, pp. 217–248, May 2022, doi: 10.1002/9781119815075.CH19.
[29] A. Jung, "Machine Learning: Foundations, Methodologies, and Applications," vol. LXX, no. 1, pp. 105–122, 2022, Accessed: Sep. 03, 2022. [Online]. Available: https://link.springer.com/bookseries/16715
[30] Y. Lei and T. Hu, "Generalization Performance of Multi-pass Stochastic Gradient Descent with Convex Loss Functions," *Journal of Machine Learning Research*, vol. 22, pp. 1–41, 2021, Accessed: Sep. 03, 2022. [Online]. Available: http://jmlr.org/papers/v22/19-716.html.
[31] S. L. Smith, B. Dherin, D. G. T. Barrett, and S. De, "On the Origin of Implicit Regularization in Stochastic Gradient Descent," Jan. 2021, doi: 10.48550/arxiv.2101.12176.
[32] S. Wojtowytsch, "Stochastic gradient descent with noise of machine learning type. Part II: Continuous time analysis," Jun. 2021, doi: 10.48550/arxiv.2106.02588.
[33] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
[34] "A 2021 Guide to improving CNNs-Optimizers: Adam vs SGD | by Sieun Park | Geek Culture | Medium." https://medium.com/geekculture/a-2021-guide-to-improving-cnns-optimizers-adam-vs-sgd-495848ac6008 (accessed Sep. 03, 2022).
[35] N. Attrapadung *et al.*, "Adam in Private: Secure and Fast Training of Deep Neural Networks with Adaptive Moment Estimation," Jun. 2021, doi: 10.48550/arxiv.2106.02203.
[36] Y. Arouri and M. Sayyafzadeh, "An adaptive moment estimation framework for well placement optimization," *Computational Geosciences 2022 26:4*, vol. 26, no. 4, pp. 957–973, Feb. 2022, doi: 10.1007/S10596-022-10135-9.
[37] X. Shen, D. Li, R. Fang, Y. Zhou, and X. Wu, "Distributed adaptive online learning for convex optimization with weight decay," *Asian J Control*, vol. 24, no. 2, pp. 562–575, Mar. 2022, doi: 10.1002/ASJC.2489.
[38] R. Llugsi, S. el Yacoubi, A. Fontaine, and P. Lupera, "Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito," *ETCM 2021 - 5th Ecuador Technical Chapters Meeting*, Oct. 2021, doi: 10.1109/ETCM53643.2021.9590681.
[39] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies 2021, Vol. 9, Page 81*, vol. 9, no. 4, p. 81, Nov. 2021, doi: 10.3390/TECHNOLOGIES9040081.
[40] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F1 measure," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial*

Intelligence and Lecture Notes in Bioinformatics), vol. 8725 LNAI, no. PART 2, pp. 225–239, 2014, doi: 10.1007/978-3-662-44851-9_15.

[41] D. Fourure, M. U. Javaid, N. Posocco, and S. Tihon, "Anomaly Detection: How to Artificially Increase Your F1-Score with a Biased Evaluation Protocol," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12978 LNAI, pp. 3–18, 2021, doi: 10.1007/978-3-030-86514-6_1/COVER.

[42] J. Miao and W. Zhu, "Precision–recall curve (PRC) classification trees," *Evolutionary Intelligence 2021 15:3*, vol. 15, no. 3, pp. 1545–1569, Apr. 2021, doi: 10.1007/S12065-021-00565-2.