

SEGMENTASI ANOMALI PRODUK DENGAN SSIM AUTOENCODER

Ahmad Sabri

Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424
sabri@staff.gunadarma.ac.id

ABSTRAK

Dalam industri manufaktur, menghasilkan produk yang bebas anomali (cacat produk) merupakan suatu keharusan. Di sisi lain, inspeksi produk secara manual cukup memakan waktu dan biaya. Penggunaan kecerdasan buatan merupakan pilihan yang tepat mengatasi kendala tersebut. Penelitian yang dilakukan Bergmann pada tahun 2018 mengusulkan model SSIM autoencoder untuk melakukan segmentasi anomali dengan citra produk sebagai inputnya. Penggunaan SSIM sebagai loss function berdasarkan pertimbangan bahwa piksel pada citra memiliki korelasi dengan piksel-piksel di sekitarnya. Dimotivasi oleh penelitian tersebut, penelitian ini menginvestigasi bagaimana pengaruh perbedaan ukuran bottleneck layer dan evaluation window size terhadap akurasi model yang didefinisikan Bergmann tersebut. Pada penelitian ini didefinisikan 3 model autoencoder, yang masing-masing memiliki ukuran bottleneck 100, 300 dan 500. Setiap model dilatih untuk mendeteksi dan melakukan segmentasi anomali pada citra produk grid, screw dan zipper berdasarkan dataset MVTec AD. Segmentasi dilakukan dengan mengacu pada threshold yang diberikan oleh indeks Youden J. Pelatihan dan evaluasi dilakukan sebanyak 100 epoch dengan menggunakan DSSIM loss function, yang merupakan fungsi dari SSIM. Evaluasi menerapkan 3 ukuran window persegi yaitu 7, 11, dan 15. Hasil pengujian memberikan maksimum AUC untuk grid, zipper, dan screw masing-masing adalah 0,9397 (ukuran bottleneck 100, window size 15), 0,7844 (ukuran bottleneck 300, window size 15), dan 0,8837 (ukuran bottleneck 500, window size 11). Disimpulkan bahwa selain konstruksi model, kompleksitas objek juga memberi pengaruh pada akurasi.

Kata Kunci: autoencoder, MVTec AD dataset, segmentasi anomali, SSIM, unsupervised learning

PENDAHULUAN

Model deteksi anomali adalah model kecerdasan buatan yang dilatih untuk mendeteksi dan melakukan segmentasi anomali pada objek dalam sebuah citra. Umumnya anomali ini terjadi pada daerah tunggal tertentu pada citra.

Dalam konteks industri manufaktur, model segmentasi anomali dapat diterapkan pada tahap *quality control* untuk mendeteksi lokasi dan jenis anomali yang terjadi pada produk. Inspeksi produk dilakukan dengan mengambil citra produk dengan kamera resolusi tinggi, dan menginput citra yang dihasilkan ke dalam model deteksi anomali. Jika terdapat anomali pada produk, maka model memberikan output berupa citra produk tersebut dengan anomali yang tersegmentasi. Metode ini dapat membuat proses inspeksi produk berlangsung lebih cepat dan dengan biaya yang lebih rendah (dalam jangka panjang) dibandingkan dengan metode inspeksi manual.

Secara teknis, deteksi anomali pada sebuah citra adalah bersifat regional, dengan proporsi luas yang kecil. Hal ini berbeda dengan *novelty detection*, di mana distribusi *inlier* dan *outlier* dalam level citra berbeda secara signifikan (oleh karena itu juga disebut sebagai klasifikasi level citra). *Novelty detection* berkembang sangat pesat disebabkan antara lain oleh ketersediaan dataset berskala besar dan melimpah, yang dipelopori oleh MNIST [1], CIFAR10 [2], dan ImageNet [3], diikuti oleh banyak sumber-sumber lainnya dalam dekade terakhir ini.

Dibandingkan dengan model klasifikasi level citra yang berkembang pesat, perkembangan model segmentasi anomali terkendala dengan sedikitnya ketersediaan dataset yang mendukung pengembangan model ini. Minimnya dataset citra produk beranomali disebabkan proporsi produk beranomali adalah sangat kecil dibanding produk normal (tanpa anomali). Selain itu,

jenis anomali seringkali tidak diketahui sebelum anomali tersebut terjadi.

Model *unsupervised learning* adalah pilihan yang tepat untuk mengatasi kelangkaan sampel produk dengan anomali. Model dilatih dengan dataset citra produk normal, sehingga mampu mengenali dan merekonstruksi citra produk normal. Citra hasil rekonstruksi kemudian dibandingkan dengan citra original. Jika pada citra original terdapat anomali, maka hal ini akan terdeteksi dari residual antara kedua citra tersebut. Autoencoder adalah model *unsupervised learning* yang tepat untuk melakukan pemodelan ini.

Bergman et al. [4,5] membangun dataset *MVTec Anomaly Detection* (MVTec AD) untuk pengembangan model segmentasi anomali. Dilanjutkan dengan penelitian [6], mereka mengusulkan dan menunjukkan bahwa penggunaan *structural similarity* (SSIM) [7] sebagai *loss function* dalam proses pelatihan maupun evaluasi model untuk segmentasi anomali pada dua jenis *fabric texture*, mampu meningkatkan metrik *area under curve* (AUC) secara signifikan, dibandingkan dengan menggunakan *loss function* ℓ^1 -distance [8], ℓ^2 -distance [9,10].

Dimotivasi oleh penelitian [6], penelitian ini menginvestigasi pengaruh perbedaan ukuran *bottleneck layer* (dimensi laten) dan *evaluation window size* terhadap akurasi model. Untuk tujuan tersebut, pada penelitian ini didefinisikan 3 jenis autoencoder yang masing-masing memiliki ukuran *bottleneck* sebesar 100, 300, dan 500.

Permasalahan dibatasi untuk segmentasi anomali pada citra *grayscale* dari objek *grid*, *zipper*, dan *screw* pada dataset MVTEC AD. Evaluasi dilakukan dengan menggunakan DSSIM *loss function* dengan 3 ukuran *window* yaitu 7, 11, dan 15. Penggunaan DSSIM dengan pertimbangan bahwa piksel pada citra memiliki korelasi dengan piksel-piksel di sekitarnya.

METODE PENELITIAN

Dataset

Dataset yang digunakan adalah citra *grayscale* dari objek *grid*, *screw*, dan *zipper* yang tersedia pada MVTEC AD [11]. Dataset terdiri dari kategori *train*, *test* (baik),

test (anomali), *groundtruth* (baik), *groundtruth* (anomali). Resume dari ketiga dataset tersebut disajikan pada Tabel 1.

Tabel 1. Resume dataset yang digunakan.

	Objek		
	Grid	Screw	Zipper
Train	264	320	240
Test (baik)	21	41	32
Test (anomali)	57	119	119
Groundtruth (baik)	21	41	32
Groundtruth (anomali)	57	119	119
Kelompok anomali	5	5	7
Jenis anomali	bent broken glue metal contamination thread	manipulated front scratch head scratch neck thread side thread top	broken teeth combined fabric border fabric interior rough split teeth squeezed teeth

Dataset untuk training hanya mencakup citra objek tanpa anomali, sedangkan dataset untuk testing mencakup citra objek dengan maupun tanpa anomali, yang dipilah berdasarkan jenis anomali. *Groundtruth* digunakan sebagai acuan untuk menghitung AUC.

Loss function

Indeks *structural similarity* (SSIM) diperkenalkan oleh Wang pada 2004 [7]. SSIM ukuran numerik dari perbedaan citra \mathbf{p} dan \mathbf{q} berukuran $K \times K$, dengan mempertimbangkan aspek *luminance* (pencahayaan) $l(\mathbf{p}, \mathbf{q})$, kontras $c(\mathbf{p}, \mathbf{q})$, dan struktur $s(\mathbf{p}, \mathbf{q})$. Indeks SSIM antara citra \mathbf{p} dan \mathbf{q} didefinisikan sebagai:

$$SSIM(\mathbf{p}, \mathbf{q}) = l(\mathbf{p}, \mathbf{q})^\alpha c(\mathbf{p}, \mathbf{q})^\beta s(\mathbf{p}, \mathbf{q})^\gamma \quad (1)$$

di mana $\alpha, \beta, \gamma \in R$ merupakan konstanta *user-defined* yang berfungsi sebagai bobot dari masing-masing aspek.

Besaran *luminance* diperoleh dengan membandingkan mean intensitas citra $\mu_{\mathbf{p}}$ dan $\mu_{\mathbf{q}}$:

$$l(\mathbf{p}, \mathbf{q}) = \frac{2\mu_{\mathbf{p}}\mu_{\mathbf{q}} + c_1}{\mu_{\mathbf{p}}^2 + \mu_{\mathbf{q}}^2 + c_1} \quad (2)$$

Besaran kontras diperoleh dari fungsi variansi masing-masing citra $\sigma_{\mathbf{p}}^2$ dan $\sigma_{\mathbf{q}}^2$:

$$c(\mathbf{p}, \mathbf{q}) = \frac{2\sigma_{\mathbf{p}}\sigma_{\mathbf{q}} + c_2}{\sigma_{\mathbf{p}}^2 + \sigma_{\mathbf{q}}^2 + c_2} \quad (3)$$

Besaran struktur diperoleh dari fungsi kovariansi σ_{pq} dari kedua citra:

$$s(\mathbf{p}, \mathbf{q}) = \frac{2\sigma_{pq} + c_2}{2\sigma_p\sigma_q + c_2} \quad (4)$$

Konstanta c_1 dan c_2 menjamin stabilitas numerik dan umumnya ditetapkan $c_1 = 0,01$ dan $c_2 = 0,03$. Dengan mensubsitusi Persamaan 2, 3, 4 ke Persamaan 1 diperoleh:

$$SSIM(\mathbf{p}, \mathbf{q}) = \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)}. \quad (5)$$

Nilai SSIM berada pada range $[-1,1]$.

SSIM = 1 jika kedua adalah citra identik.

Untuk *loss function* digunakan indeks *structural dissimilarity* (DSSIM), yang merupakan fungsi dari SSIM, dan didefinisikan sebagai:

$$DSSIM = \frac{1 - SSIM(\mathbf{p}, \mathbf{q})}{2}. \quad (6)$$

Penggunaan DSSIM sebagai *training loss* digagas oleh Bergman et al. [6] untuk mengakomodir dependensi antar piksel yang berdekatan. Hal ini tidak terakomodir dengan loss ℓ^2 yang digunakan dalam penelitian-penelitian sebelumnya [8,9,10].

Bergman menunjukkan bahwa penggunaan DSSIM mampu meningkatkan metrik *area under curve* (AUC) dibandingkan dengan ℓ^2 .

Autoencoder

Model yang digunakan adalah autoencoder dengan struktur sebagaimana yang didefinisikan pada [6]. Gambar 1 menunjukkan susunan layer pada bagian *encoder*. Layer *decoder* didefinisikan dengan urutan sebaliknya, dengan menggunakan *deconvolutional* layer.

Layer	Output Size	Parameters		
		Kernel	Stride	Padding
Input	128x128x1			
Conv1	64x64x32	4x4	2	1
Conv2	32x32x32	4x4	2	1
Conv3	32x32x32	3x3	1	1
Conv4	16x16x64	4x4	2	1
Conv5	16x16x64	3x3	1	1
Conv6	8x8x128	4x4	2	1
Conv7	8x8x64	3x3	1	1
Conv8	8x8x32	3x3	1	1
Conv9	1x1xd	8x8	1	0

Gambar 1. Struktur bagian encoder.

Citra original pada MVTEC AD berukuran 1024×1024 (untuk *grid*, *zipper* dan *screw*). Sebelum diinput ke model, dilakukan *cropping* dengan ukuran 256×256 . Setiap *patch* dilakukan *resizing* menjadi berukuran 128×128 . *Patching* dilakukan karena idealnya SSIM tidak diterapkan dalam level citra, melainkan dalam cakupan area yang lebih kecil. Proses ini mengakibatkan setiap citra pada dataset training menghasilkan 16 *patch* berukuran 128×128 .

Layer Conv9 adalah *bottleneck layer* dengan dimensi laten $1 \times 1 \times d$. Pada penelitian ini dilakukan investigasi model untuk $d \in \{100,300,500\}$, untuk melihat bagaimana pengaruh variasi ukuran bottleneck terhadap AUC.

Setelah itu dilakukan *train-test split*, dengan proporsi 80 : 20 pada dataset train.

Segmentasi anomali

Untuk memperoleh segmentasi anomali, dilakukan penghitungan *loss* antara citra original dengan citra rekonstruksi dengan fungsi berikut, yang pada dasarnya merupakan Persamaan 6:

$$L(\mathbf{x}, \theta) = \frac{1 - SSIM(\mathbf{x}, \hat{\mathbf{x}})}{2} \quad (7)$$

di mana \mathbf{x} dan $\hat{\mathbf{x}}$ berturut-turut adalah matriks citra original dan citra rekonstruksi, $\theta = (w_E, b_E, w_D, b_D)$ adalah bobot dan bias dari *encoder* dan *decoder layer*. Fungsi $L(\mathbf{x}, \theta)$ menghasilkan citra residual.

Threshold untuk segmentasi anomali mengacu pada indeks Youden J yang didefinisikan sebagai:

$$J = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1. \quad (8)$$

dimana TP, TN, FN, FP masing-masing adalah *true positive*, *true negative*, *false negative* dan *false positive*.

Segmentasi diperoleh dengan memetakan $L(\mathbf{x}, \theta)$ ke matriks citra biner \mathbf{M} dengan fungsi:

$$m_{ij} = \begin{cases} 1, & \text{jika } L_{ij} > J \\ 0, & \text{dalam hal lain} \end{cases} \quad (9)$$

di mana m_{ij} dan L_{ij} masing-masing adalah entri baris i kolom j dari \mathbf{M} dan $\mathbf{L}(\mathbf{x}, \boldsymbol{\theta})$.

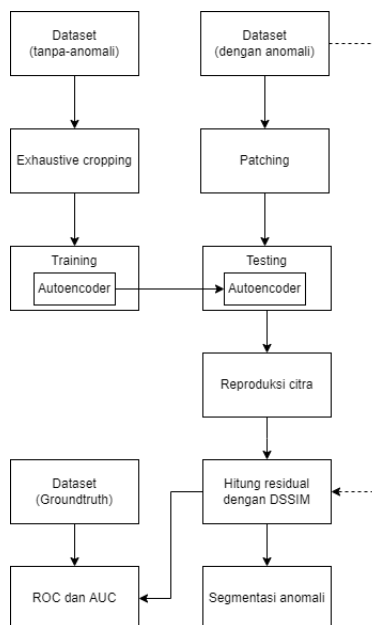
Prosedur

Pada dataset tanpa anomali dilakukan *exhaustive cropping* dan *resizing*, kemudian autoencoder dilatih dengan *cropped-resized* dataset tersebut.

Autoencoder terlatih kemudian diuji dengan dataset anomali. Citra yang direkonstruksi oleh model terlatih tersebut kemudian dibandingkan dengan citra original, yang menghasilkan citra residual berdasarkan DSSIM.

Segmentasi anomali terjadi pada piksel dengan *loss* yang lebih besar dari *threshold* yang ditetapkan. Metrik AUC dihitung dengan membandingkan piksel-piksel anomali yang *overlap* dengan piksel anomali pada *groundtruth*. Secara grafis, AUC divisualisasikan sebagai kurva *Receiver Operating Characteristics* (ROC).

Gambar 2 menampilkan alur skematik dari prosedur yang telah dijelaskan. Pelatihan dan evaluasi model dilakukan dengan bahasa pemrograman Python dengan referensi pada [12,13]



Gambar 2. Prosedur pelatihan dan evaluasi model.

HASIL DAN PEMBAHASAN

Tiga model autoencoder dengan struktur yang telah dijelaskan pada bagian Metode Penelitian, dengan perbedaan ukuran *bottleneck* yaitu 100, 300, dan 500, dilatih dengan menggunakan dataset *grid*, *zipper*, dan *screw* dari MVTEC AD.

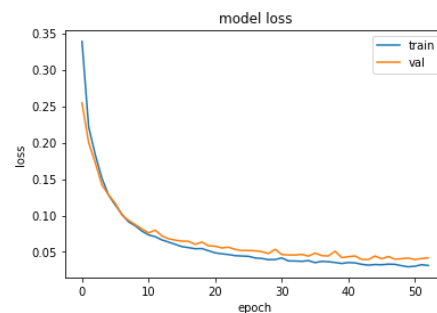
Model terlatih dievaluasi menggunakan *SSIM loss function* yang dihitung untuk *window* persegi berukuran 7 dan 15.

Untuk menjelaskan implementasi prosedur yang dijelaskan pada bagian Metode Penelitian, berikut diberikan tampilan visual sebuah sampel dari masing-masing objek untuk citra beranomali original, citra yang direkonstruksi model, citra residual (dihasilkan dari pengurangan antara citra original dengan citra rekonstruksi), citra hasil *thresholding* yang menampilkan segmentasi anomali, serta grafik ROC dari sampel.

Di bagian akhir ditampilkan resume hasil dari seluruh skenario.

Pelatihan dengan dataset *Grid*

Proses training dengan dataset *grid*, dalam contoh ini untuk autoencoder berdimensi laten 500, diberikan oleh grafik *loss vs epoch* pada Gambar 3.



dataset=grid, latent dim=500, batch size=16, loss=ssim, random crop=0, #epoch=100

Gambar 3. Training/validation loss history untuk dataset *grid*, dimensi laten 500.

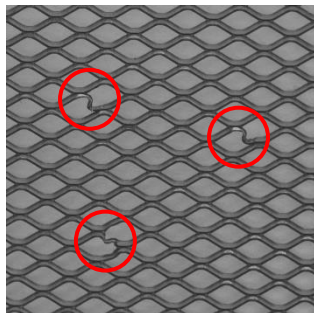
Gambar 4 menampilkan sampel citra original untuk *grid* yang memiliki anomali berupa tekukan (*bent*). Citra ini diinput ke dalam model autoencoder yang sudah dilatih, dan model ini memberikan output berupa rekonstruksi citra sebagaimana ditunjukkan pada Gambar 5. Tampak bahwa citra rekonstruksi merupakan citra tanpa anomali.

Selanjutnya dilakukan perhitungan residual berdasarkan Persamaan 7, di mana x dan \hat{x} berturut-turut adalah matriks citra original dan citra rekonstruksi. Citra residual yang dihasilkan ditampilkan pada Gambar 6. Dari citra tersebut tampak bagian anomali *bent* yang terdapat pada produk.

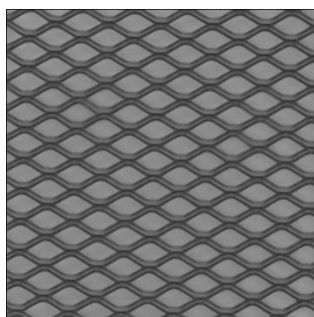
Selanjutnya dilakukan *thresholding* dengan *threshold* mengacu pada Indeks Youden (Persamaan 8). Untuk contoh ini diperoleh Indeks Youden J sebesar 0,7573, Mengacu pada Persamaan 9 dan nilai J tersebut, maka fungsi *thresholding* didefinisikan sebagai:

$$m_{ij} = \begin{cases} 1, & \text{jika } L_{ij} > 0,7573 \\ 0, & \text{dalam hal lain} \end{cases}$$

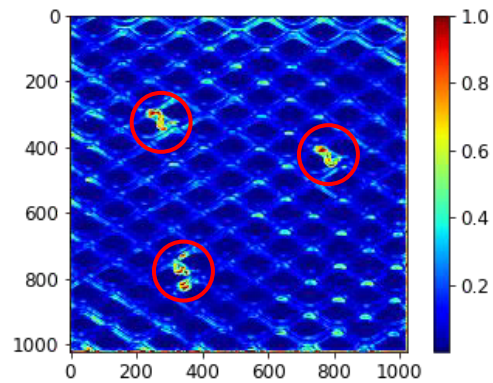
Citra yang dihasilkan oleh *thresholding* ini berupa citra biner, di mana piksel yang terang merupakan lokasi anomali, sebagaimana ditampilkan pada Gambar 7.



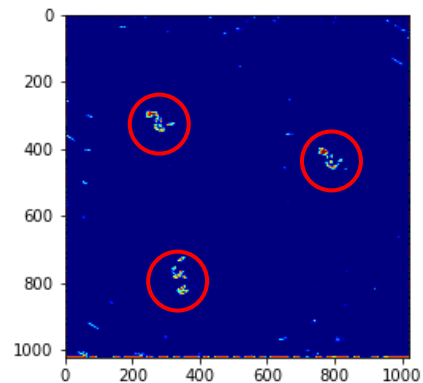
Gambar 4. *Bent grid (original)*



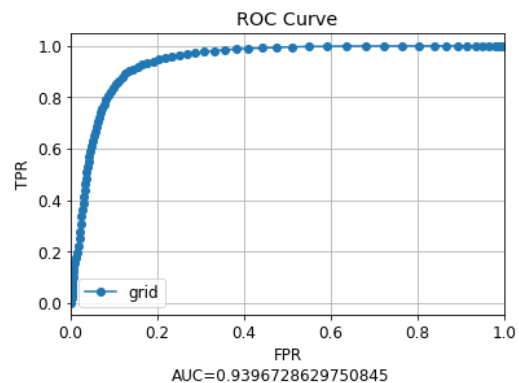
Gambar 5. *Bent grid (rekonstruksi)*



Gambar 6. *Citra residual untuk citra bent grid original dan rekonstruksinya.*



Gambar 7. *Citra segmentasi hasil thresholding citra residual.*



Gambar 8. *Kurva ROC untuk grid autoencoder dengan dimensi laten 500, ukuran window 15.*

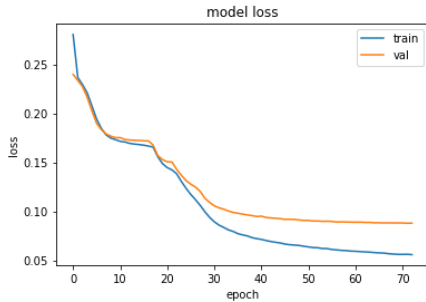
Pengujian model dengan dimensi laten 500 dan ukuran window 7 menghasilkan kurva ROC pada Gambar 8 dengan $AUC=0,9397$.

Selanjutnya diberikan penjelasan implementasi prosedur pelatihan dan evaluasi model berdasarkan dataset *zipper* dan *screw*. Penjelasan diberikan dengan

lebih ringkas karena prosedurnya serupa dengan yang telah dijelaskan sebelumnya.

Pelatihan dengan dataset *Zipper*

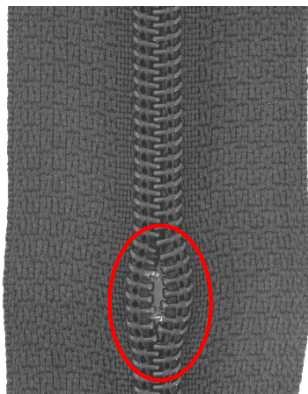
Gambar 9 menampilkan grafik loss vs epoch selama training dengan dataset zipper, menggunakan dimensi laten 500.



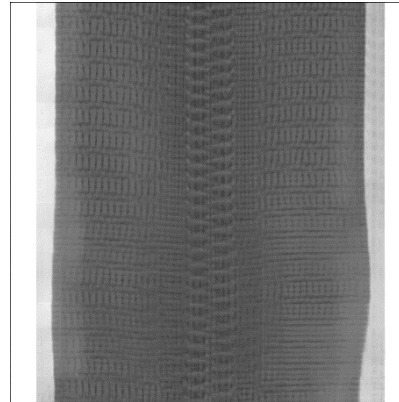
dataset=zipper, latent dim=300, batch size=16, loss=ssim, random crop=0, #epoch=100

Gambar 9. Training/validation loss history untuk dataset zipper, dimensi laten 300.

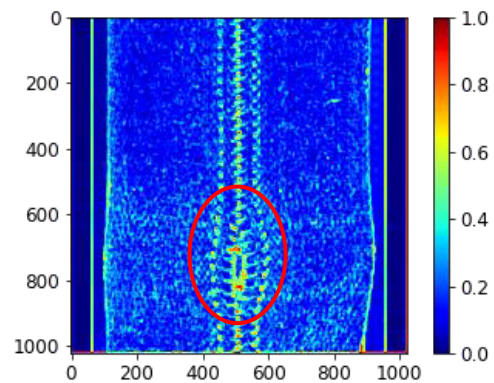
Sampel citra original untuk zipper dengan anomali berupa gerigi terlepas (*split teeth*) diberikan pada Gambar 10. Rekonstruksi citra oleh model memberikan hasil sebagaimana ditunjukkan pada Gambar 11. Anomali tidak dihasilkan pada citra rekonstruksi tersebut. Citra residual ditampilkan pada Gambar 12, menunjukkan lokasi anomali.



Gambar 10. Split-teeth zipper (original).

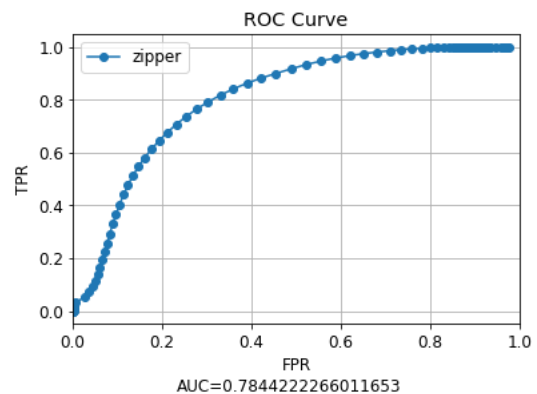


Gambar 11. Split-teeth zipper (rekonstruksi)



Gambar 12. Citra residual untuk citra split-teeth zipper original dan rekonstruksinya

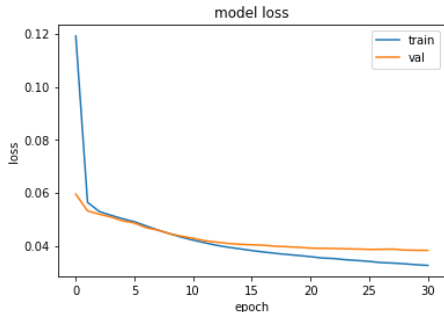
Pengujian model dengan dimensi laten 300 dan ukuran window 15 menghasilkan kurva ROC pada Gambar 13 dengan AUC 0,7844.



Gambar 13. Kurva ROC untuk zipper autoencoder dengan dimensi laten 300, ukuran window 15, menghasilkan AUC 0,7844.

Pelatihan dengan dataset *Screw*

Gambar 14 menampilkan grafik loss vs epoch selama training dengan dataset grid, model memiliki dimensi laten 500.



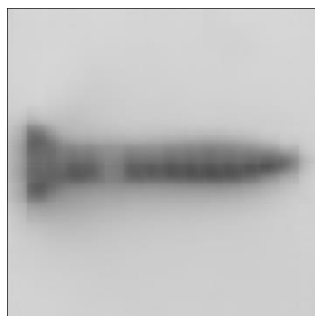
dataset=screw, latent dim=500, batch size=16, loss=ssim, random crop=0, #epoch=100

Gambar 14. Training/validation loss history untuk dataset zipper, dimensi laten 300.

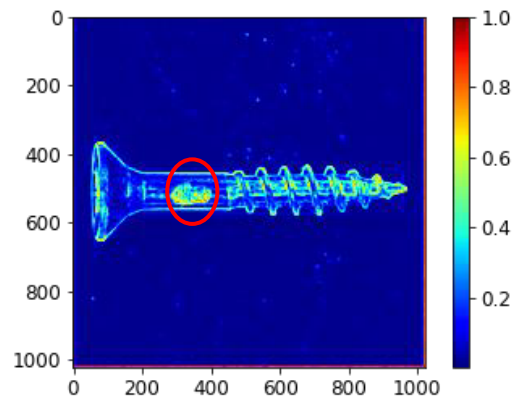
Gambar 15 menampilkan sampel citra original untuk *screw* dengan anomali berupa goresan pada leher (*scratch neck*). Rekonstruksi citra memberikan hasil sebagaimana ditunjukkan pada Gambar 16. Tampak bahwa anomali dihasilkan dengan redup pada citra rekonstruksi. Citra residual ditampilkan pada Gambar 17. Lokasi anomali tampak dengan jelas pada bagian yang dilingkari.



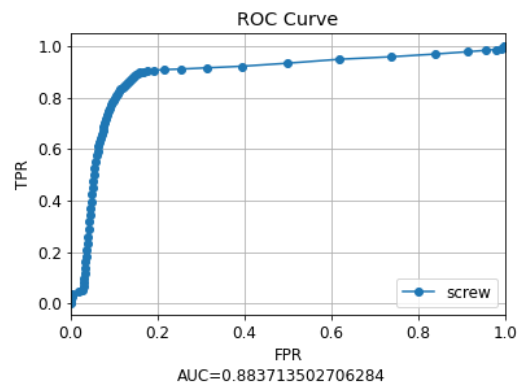
Gambar 15. *Scratch-neck screw* (original).



Gambar 16. *Scratch-neck screw* (rekonstruksi).



Gambar 17. Citra residual untuk citra *scratch-neck screw* original dan rekonstruksinya



Gambar 18. Kurva ROC untuk *screw* autoencoder dengan dimensi laten 500, ukuran window 11, menghasilkan AUC 0,8837.

Pengujian model dengan dimensi laten 500 dan ukuran window 11 menghasilkan kurva ROC pada Gambar 18 dengan AUC 0,8837.

Resume hasil

Resume hasil secara lengkap diberikan oleh Tabel 2.

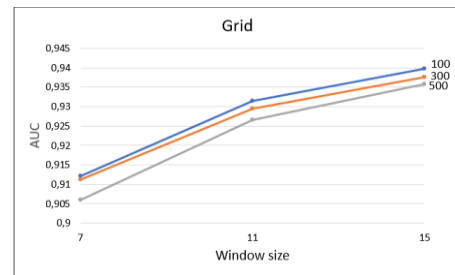
Tabel 2. Resume hasil dari semua skenario.

Objek	Latent dim	Ukuran window	AUC	Max	Min	Range
Grid	100	7	0,9121	0,9397	0,9060	0,0337
		11	0,9315			
		15	0,9397			
	300	7	0,9113			
		11	0,9295			
		15	0,9375			
	500	7	0,9060			
		11	0,9265			
		15	0,9358			
Zipper	100	7	0,7037	0,7844	0,7037	0,0807
		11	0,7495			
		15	0,7732			
	300	7	0,7051			
		11	0,7578			
		15	0,7844			
	500	7	0,7097			
		11	0,7570			
		15	0,7813			
Screw	100	7	0,8818	0,8837	0,8746	0,0091
		11	0,8832			
		15	0,8781			
	300	7	0,8768			
		11	0,8779			
		15	0,8746			
	500	7	0,8828			
		11	0,8837			
		15	0,8805			

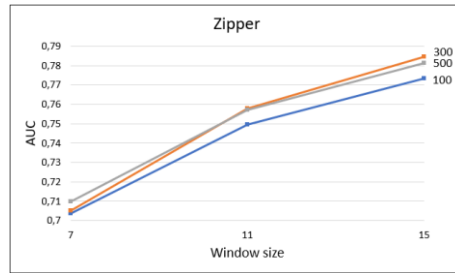
Dari 9 skenario pengujian model untuk setiap jenis objeknya, diperoleh hasil berikut:

1. AUC tertinggi untuk *grid* adalah 0,9397, diberikan oleh model dengan dimensi laten 100 dan ukuran window SSIM 15.
2. AUC tertinggi untuk *zipper* adalah 0,7844, diberikan oleh model dengan dimensi laten 300 dan ukuran window SSIM 15.
3. AUC tertinggi untuk *screw* adalah 0,8837, diberikan oleh model dengan dimensi laten 300 dan ukuran window SSIM 11.
4. Untuk *screw*, variasi dimensi laten dan ukuran window memberikan peningkatan AUC yang tidak signifikan, yaitu sebesar 0,0091.
5. Untuk *zipper*, variasi dimensi laten dan ukuran window memberikan peningkatan AUC yang signifikan, yaitu sebesar 0,0807.
6. Untuk *grid*, variasi dimensi laten dan ukuran window memberikan peningkatan AUC sebesar 0,0337.

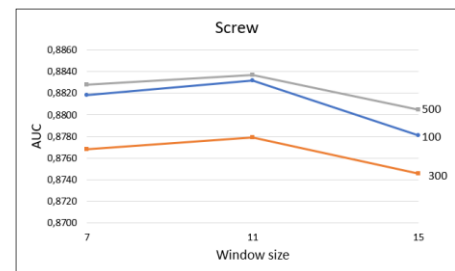
Representasi grafik *window size vs AUC* dari hasil yang diperoleh untuk *grid*, *zipper*, dan *screw* masing-masing diberikan oleh Gambar 19a, b, dan c.



(a)



(b)



(c)

Gambar 19. Grafik *window size vs AUC* untuk (a) *grid*, (b) *zipper*, (c) *screw*, menggunakan model dengan ukuran *bottleneck* 100, 300, 500.

Dari grafik tersebut diperoleh karakteristik berikut:

1. Untuk *grid*, peningkatan ukuran *bottleneck* berimplikasi pada turunnya AUC untuk window size yang sama
2. Untuk *zipper*, AUC dari model dengan ukuran *bottleneck* 300 melampaui AUC dari model dengan ukuran *bottleneck* 100 dan 500 untuk *window size* 11 dan 15.
3. Untuk *screw*, model dengan ukuran *bottleneck* 500 mengungguli model lainnya untuk semua *window size*.

PENUTUP

Secara umum, selain pengaruh dari ukuran *bottleneck* (dimensi laten) dan window size, nilai AUC juga dipengaruhi oleh faktor kompleksitas dari struktur objek. Semakin kompleks struktur objeknya (dalam hal ini *zipper* dan *screw* memiliki struktur objek yang lebih kompleks dibandingkan *grid*), maka penambahan ukuran *bottleneck* memberikan peningkatan pada AUC. Hal ini karena objek yang kompleks membutuhkan fitur-fitur ekstraksi yang lebih banyak.

Namun untuk objek yang kompleksitasnya lebih rendah (*grid*), peningkatan ukuran *bottleneck* mengakibatkan model merekonstruksi citra yang sangat mendekati citra inputnya. Akibatnya, anomali tidak tampak pada citra residual.

Demikian pula halnya dengan penambahan ukuran window SSIM. Secara umum peningkatan ukuran window sampai batas tertentu berimplikasi pada peningkatan AUC.

Penelitian selanjutnya dapat berupa modifikasi struktur autoencoder untuk memberikan AUC yang lebih baik, terutama untuk *zipper* dan *screw*, dan pengembangan untuk mengakomodir input citra berwarna. Pengembangan lebih jauh adalah mewujudkan aplikasi *end-user* untuk menerapkan metode ini pada dunia industri.

DAFTAR PUSTAKA

- [1] Y. LeCun, L. Bottou, Y. Bengio dan P. Haffner, "Gradientbased learning applied to document recognition", *Proceedings of the IEEE*, 86(11), 2278–2324, 1998.
- [2] A. Krizhevsky dan G. Hinton, "Learning multiple layers of features from tiny images", *Technical report*, University of Toronto, 2009.
- [3] A. Krizhevsky, I. Sutskever dan G.E. Hinton, "ImageNet classification with deep convolutional neural networks", *Proceedings of the 25th international conference on neural information processing systems*, vol. 1, 1097–1105, 2012.
- [4] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, "The MVTEC Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection", *International Journal of Computer Vision*, 129(4), hal. 1038-1059, 2021.
- [5] P. Bergmann, M. Fauser, D. Sattlegger dan C. Steger, "MVTEC AD - A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, hal. 9584-9592, 2019.
- [6] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger dan C. Steger, "Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders", *VISAPP*, hal. 372-380, 2019.
- [7] Z. Wang, A. C. Bovik, H.R. Sheikh, E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [8] C. Baur, B. Wiestler, S. Albarqouni, N. Navab, "Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images", *arXiv preprint arXiv:1804.04488*, 2018.
- [9] D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone, "Defect Detection in SEM Images of Nanofibrous Materials", *IEEE Transactions on Industrial Informatics*, 13(2):551–561, 2017.
- [10] T. Schlegl, P. Seebock, S.M. Waldstein, U. Schmidt-Erfurth, G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery", *International Conference on Information Processing in Medical Imaging*, hal. 146–157, Springer, 2017.
- [11] The MVTEC Anomaly Detection Dataset (MVTEC AD), tersedia daring pada <https://www.mvtec.com/company/research/datasets/mvtec-ad>, diakses pada Juni 2022.

- [12] Unsupervised-Anomaly-Detection-with-SSIM-AE, tersedia daring pada <https://github.com/PabloMaj/Unsupervised-Anomaly-Detection-with-SSIM-AE>, diakses pada Juni 2022.
- [13] AutoEncoder with SSIM loss, <https://github.com/plutoyuxie/AutoEncoder-SSIM-for-unsupervised-anomaly-detection->, diakses pada Juni 2022.