JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

PENGEMBANGAN MODEL CAPSULEGAN UNTUK PENGHAPUSAN HUJAN CITRA TUNGGAL

Haryanto Hidayat*[1)], Munawir[2)], Muhammad Taufik Dwi Putra[3)], Arief Suryadi Satyawan[4)]
1. Program Studi Teknik Komputer, Universitas Pendidikan Indonesia, Indonesia
2. Program Studi Teknik Komputer, Universitas Pendidikan Indonesia, Indonesia
3. Program Studi Teknik Komputer, Universitas Pendidikan Indonesia, Indonesia
4. Badan Riset dan Inovasi Nasional, Indonesia

Article Info
Kata Kunci: *De-raining; Capsule Network; Generative Adversarial Network*; *Computer Vision*; *Image-to-Image Translation*

Keywords: *De-raining; Capsule Network; Generative Adversarial Network; Computer Vision; Image-to-Image Translation*

Article history:
Received 21 March 2024
Revised 4 April 2024
Accepted 18 April 2024
Available online 1 June 2024

DOI :
https://doi.org/10.29100/jipi.v9i2.5534

* Corresponding author.
Haryanto Hidayat
E-mail address:
haryanto@upi.edu

ABSTRAK
Pengaruh cuaca hujan pada kualitas gambar sering menjadi masalah di bidang *Computer Vision* (CV), karena informasi-informasi penting yang diperlukan oleh algoritma CV menjadi hilang. Berbagai macam solusi telah diusulkan oleh para peneliti untuk menyelesaikan masalah tersebut, mulai dari menggunakan filter tradisional hingga penerapan metode *Deep Learning*. Penerapan algoritma *Deep Learning*, seperti *Deep Convolutional Generative Adversarial Network* (DCGAN) digunakan karena tingkat kualitas gambar yang diproduksi sangat baik, tetapi masih ada kekurangan yaitu hilangnya informasi spasial antar komponen hujan, sehingga tidak dapat mengidentifikasi dimana letak garis hujan yang menyebabkan tersisanya garis hujan pada gambar. *Capsule Network* (CapsNet) menjadi solusi dalam permasalahan tersebut, dengan memperhatikan hubungan antara detail parsial dengan objek global, informasi-informasi spasial pada gambar seperti posisi dan rotasi antar objek dapat dipertahankan, dengan begitu penggunaan CapsNet pada arsitektur akan memberikan pengaruh yang cukup signifikan. Dengan menggabungkan kedua metode tersebut akan didapatkan model *de-raining* yang dapat menghasilkan gambar lebih tajam sekaligus menghilangkan garis hujan secara efektif. Kami menggabungkan CapsNet pada bagian arsitektur *Discriminator* untuk pengklasifikasian yang lebih baik. Hasil perbandingan dengan model lain menunjukkan bahwa penggabungan kedua arsitektur tersebut menghasilkan gambar yang lebih baik dibandingkan dengan kebanyakan model *Deep Learning* lain. Meskipun begitu, masih terdapat kekurangan yaitu gambar yang dihasilkan masih memiliki efek *blur* dan sisa hujan akibat proses pelatihan yang tidak stabil.

ABSTRACT
The effect of rainy weather on image quality is often a problem in the field of Computer Vision (CV) because important information required by the CV algorithm is lost. Various solutions have been proposed by researchers to solve this problem, ranging from using traditional filters to applying Deep Learning methods. The application of Deep Learning algorithms, such as the Deep Convolutional Generative Adversarial Network (DCGAN) is being used because the level of image quality produced is excellent, however, there are still shortcomings i.e the loss of spatial information between rain components, so it cannot identify where the rain streaks are, which causes rain lines to remain on the image. Capsule Network (CapsNet) is a solution to this problem, by paying attention to the relationship between partial details and global objects, spatial information in the image such as position and rotation between objects can be maintained, so the use of CapsNet in architecture will have a quite significant impact. By combining these two methods, a de-raining model can be obtained which can produce sharper images while effectively removing rain streaks. We incorporate CapsNet in the Discriminator part of the architecture for better classification. Comparison results with other models show that combining the two architectures produces better images compared to most Deep Learning models. However, there are still shortcomings, that is the generated image still has a blur effect and residual rain due to the unstable training process.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

I. PENDAHULUAN

CUACA hujan merupakan cuaca yang sering ditemui dalam kehidupan sehari-hari, hujan dapat mempengaruhi kualitas gambar yang diambil menggunakan kamera, yaitu dengan memberi efek hamburan, buram, dan defleksi cahaya [1]. Masalah tersebut menyebabkan turunnya kualitas gambar yang ditangkap [2], sehingga membuat konteks gambar menjadi tidak jelas dan gambar menjadi tidak dapat digunakan karena sulit untuk mengenali objek pada gambar secara akurat. Oleh karena itu, garis-garis hujan pada gambar seringkali menjadi masalah dalam kinerja bidang penglihatan citra (*computer vision*), seperti deteksi objek [3], sistem kemanan [4], kendaraan pintar [5], dan pencitraan udara [6], karena algortima *computer vision* (CV) gagal dalam memperoleh informasi penting yang dibutuhkan.

Image de-raining merupakan suatu metode yang bertujuan untuk memperbaiki kualitas gambar yang rusak akibat hujan dengan cara menghapus atau menambal garis-garis hujan dengan prediksi latar belakang aslinya [7], sehingga gambar dapat terlihat lebih jelas dan informasi yang hilang dapat didapatkan kembali dengan mudah. Gambar yang dihasilkan dari *image de-raining* kemudian dapat diolah kembali untuk menerapkan algoritma CV. Berbeda dengan metode *video de-raining* seperti yang dilakukan oleh penelitian [8], [9], [10], [11], [12], [13] yang sebagian besar menggunakan *frame* sebelumnya yang tidak terkena garis-garis hujan untuk menambal garis-garis hujan pada *frame* setelahnya, *image de-raining* dianggap lebih sulit karena tidak ada referensi latar belakang gambar seperti *frame* pada video [14]. Banyak peneliti sudah mencoba berbagai macam cara untuk menyelesaikan masalah *image de-raining* dengan metode usulannya masing-masing.

Penelitian yang dilakukan oleh Chen dkk. dan Kang dkk. menunjukkan bahwa penggunaan metode *image de-raining* tradisional menggunakan filter memberikan hasil yang tidak cukup karena pemetaan transformasi linier terbatas pada kasus hujan yang kompleks [15], [16]. Kedua penelitian tersebut menganalisis karakteristik hujan yang sama, seperti garis putih yang serupa dan berulang, yang berarti proses *de-raining* tidak dapat dilakukan secara optimal pada gambar yang memiliki arah, kepadatan, dan ukuran yang berbeda [17]. Mengikuti penelitian tersebut, penelitian-penelitian *image de-raining* yang menggunakan metode *Gaussian Mixture Model* (GMM) [18], *Discriminative Sparse Coding* [19], *Lightweight Pyramid of Networks* (LPNet) [20], *Sparse Transformer Network* [21], dan *Semi-supervised Tansfer Learning* [22] telah mencoba untuk memecahkan masalah *image de-raining*. Namun, tidak ada satupun yang memberikan hasil cukup memuaskan.

Perkembangan metode *deep learning* (DL) yang pesat akhirnya membuat titik terang untuk permasalahan *image de-raining*. Kinerja yang ditunjukkan metode DL memberikan hasil yang mengejutkan, potensi pemanfaatan DL dalam image de-raining telah banyak diujicobakan. Penelitian pertama, yang dilakukan oleh Fu dkk. menggunakan jaringan syaraf tiruan, yaitu *Convolutional Neural Network* (CNN), untuk mempersingkat proses pemetaan dari *input* ke *output* sehingga proses pelatihan menjadi lebih mudah. Kemudian, untuk menghasilkan gambar yang lebih baik, penelitian ini berfokus pada detail frekuensi tinggi saat pelatihan, yaitu dengan menghilangkan keterlibatan latar belakang gambar dengan memfokuskan pada struktur hujan pada gambar. Dengan menggunakan metode ini hasil yang diperoleh lebih baik dibandingkan dengan metode tradisional, karena dapat menangani hujan yang mempunyai arah dan rotasi berbeda, namun metode ini masih lemah dalam menangani hujan yang deras [23].

Penelitian lain yang menggunakan metode DL dilakukan oleh Qian dkk., penelitian tersebut telah menghasilkan restorasi gambar yang sangat baik dengan teknik *Generative Adversarial Network* (GAN). Penelitian ini sedikit berbeda dengan pendekatan *image de-raining* lain karena yang dihilangkan bukanlah garis-garis hujan, melainkan tetesan air hujan, yang berarti daerah yang perlu diperbaiki lebih besar dibandingkan garis-garis hujan biasa. Penelitian ini menggunakan metode *Attentive* GAN, model jaringan generatif mempelajari pola gambar tetesan air hujan pada gambar kemudian menerapkannya pada gambar *input* untuk menghasilkan gambar tanpa hujan [24].

Metode-metode *image de-raining* yang dikembangkan kebanyakan tidak dapat mengatasi masalah hujan lebat, biasanya metode-metode tersebut masih meninggalkan sisa-sisa garis hujan pada gambar, namun penelitian yang dilakukan oleh Li dkk. membuktikan lagi bahwa metode GAN sangat ampuh. Mereka mengusulkan metode baru dengan jaringan 2 tahap. Pertama, mereka menggunakan *guided filter framework* untuk menguraikan gambar menjadi 2 bagian, yaitu latar belakang dan *masking* garis-garis hujan. Kedua, mereka menyempurnakan gambar yang sudah dihilangkan garis hujannya menggunakan GAN untuk memulihkan detail gambar. Hasil penelitian ini dapat menghilangkan hujan lebat secara efektif [25].

Penelitian yang menggabungkan antara CNN dan GAN dilakukan oleh Zhang dkk., arsitektur yang terdiri dari *generator, discriminator*, dan *refined perceptual loss*. Model yang diusulkan menggunakan arsitektur CNN dalam penerapannya dilatih secara *adversarial* sehingga model generatifnya dapat menghasilkan restorasi gambar tanpa hujan [26]. Meskipun penelitian tersebut dapat menghilankan sebagian besar garis-garis hujan pada gambar, tetapi belum cukup untuk menghilangkan garis hujan dengan posisi rotasi yang berbeda, karena informasi spasial antar objek hujan tidak didapatkan, itulah kelemahan menggunakan arsitektur CNN. Sabour dkk. kemudian

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

memperkenalkan Capsule Network (CapsNet) sebagai alternatif yang memberikan solusi daripada kekurangan CNN [27]. Pada CapsNet terdapat *dynamic routing* yang dapat mempertahankan informasi spasial dan invarian terhadap rotasi objek [28].

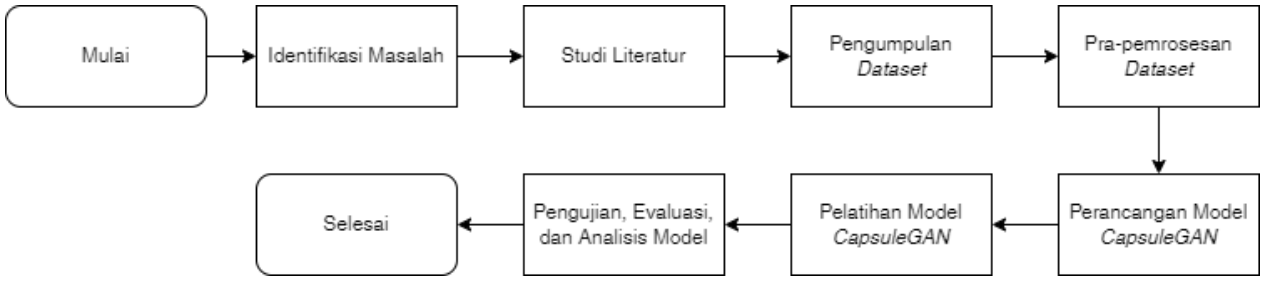
Penelitian Jaiswal dkk. berkesperimen dengan menggabungkan metode GAN dengan CapsNet untuk mereproduksi gambar angka tulisan tangan menggunakan dataset *Modified National Institute of Standards and Technology database* (MNIST). Penelitian ini menggunakan arsitektur *Deep Convolutional GAN* (DCGAN) pada bagian generator dan menggunakan CapsNet pada bagian *discriminator* untuk menangkap hubungan spasial antar vektor untuk mendapatkan hasil klasifikasi yang lebih baik. Hasil penelitian menunjukkan bahwa gambar yang direkonstruksi lebih tajam dibandingkan dengan model yang tanpa menggunakan CapsNet [29]. Meskipun hasil penelitian tersebut sudah unggul pada dataset MNIST, tetapi permasalahan berlanjut pada gambar yang memiliki warna 3-channel (RGB) yang lebih kompleks. Penelitian Xiang dkk. menunjukkan bahwa arsitektur CapsNet belum bisa merekonstruksi gambar RGB dengan baik [30].

Dalam memberikan solusi terhadap permasalahan tersebut, kami mengusulkan pengembangan model *image de-raining* menggunakan penggabungan model CapsNet dan GAN untuk menghasilkan rekonstruksi gambar tanpa hujan yang lebih baik dengan mempertahankan informasi spasial antar objek yang selanjutnya akan dinotasikan sebagai CapsuleGAN.

II. METODOLOGI PENELITIAN

A. Tahapan Penelitian

Penelitian ini dilakukan dengan beberapa tahap agar penelitian yang dilakukan berjalan secara sistematis dan mendapatkan hasil yang diharapkan. Tahapan penelitian yang dilakukan digambarkan pada Gambar 1.



Gambar. 1. Tahapan Penelitian

Studi literatur dilakukan dengan mengumpulkan, membaca, dan mempelajari studi-studi terkait yang mendukung penelitian. Setelah studi literatur dirasa sudah cukup, selanjutnya adalah mencari *dataset* yang tepat untuk digunakan pada penelitian, hal ini sangat penting karena dapat mempengaruhi hasil penelitian, apakah sesuai dengan tujuan penelitian. Hal selanjutnya merupakan pra-pemrosesan *dataset*, ini merupakan langkah awal dalam pembuatan arsitektur model, karena *dataset* yang digunakan harus sesuai dengan model yang akan dibuat, proses ini meliputi mengubah ukuran gambar, mengatur ukuran *batch* data, dan mengubah tipe *dataset* menjadi bentuk *array* agar dapat diproses oleh model. Ketika semua data sudah siap, langkah selanjutnya merupakan desain dan impelementasi model CapsuleGAN, hal ini meliputi perancangan model *Generator* dan *Discriminator*, fungsi *loss*, *optimizer*, dan fungsi aktifasi yang digunakan. Model yang sudah dibuat kemudian dilatih menggunakan *dataset* yang sudah disiapkan sebelumnya, pelatihan dilakukan selama iterasi yang sudah diatur. Model yang sudah dilatih kemudian dievaluasi dan analisis berdasarkan hasil gambar yang dihasilkan yaitu menggunakan pengukuran nilai evaluasi metrik maupun hasil inpeksi visual.

B. Dataset Gambar Penelitian

Dataset yang kami gunakan terbagi menjadi 2 macam, yaitu *dataset* hujan sintetis dan *dataset* hujan asli. Masing-masing *dataset* digunakan untuk evaluasi model yang dibuat, sedangkan untuk pelatihan model digunakan *dataset* hujan sintetis.

1) Dataset Hujan Siintetis

Karena sulit untuk mendapatkan *dataset* gambar hujan yang berpasangan dengan gambar tanpa hujan di dunia nyata, kami menggunakan *dataset* sintetis publik *Rain1200* [31]. *Dataset* tersebut terdiri dari 4.000 pasang gambar untuk *training* dan 1.200 pasang gambar untuk *testing* dengan masing-masing berpasangan antara gambar hujan dan gambar tanpa hujan. Penggunaan *dataset* sintetis yang berpasangan diperlukan untuk evaluasi model, karena proses tes memerlukan perbandingan antara gambar *input* (gambar hujan), gambar yang di-

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

generate dari model yang dibuat, dan gambar target (gambar tanpa hujan). Gambar 2 menunjukkan contoh *dataset* gambar hujan sintetis yang digunakan dalam penelitian ini.

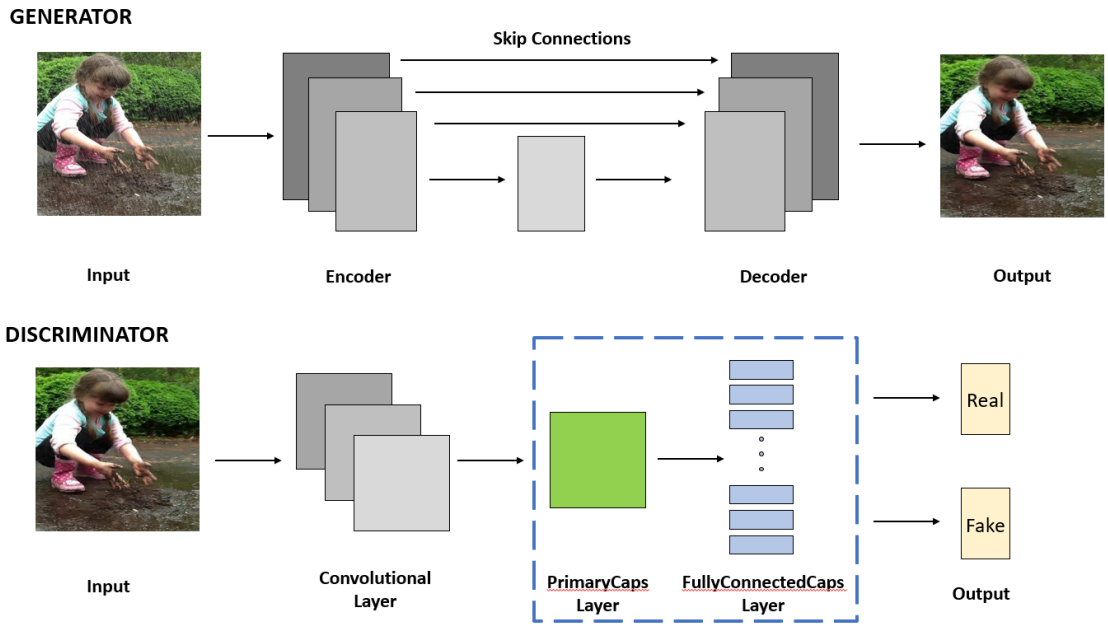
Gambar. 2. Sampel *Dataset Rain 1200*

2) Dataset Hujan Asli

Evaluasi model juga diterapkan pada *dataset* hujan asli untuk mengukur seberapa baik model yang dibuat untuk masalah hujan di dunia nyata. *Dataset* hujan asli digunakan untuk evaluasi secara inspeksi visual, *dataset* yang digunakan dikumpulkan dari internet.

C. Arsitektur CapsuleGAN

Pengembangan yang kami usulkan pada model ini dimulai dari pengembangan arsitektur dasar yang dibuat. Karena ini adalah tugas *image-to-image translation*, kami mengadopsi arsitektur DCGAN "U-Net" [32] (atau lebih dikenal dengan Pix2Pix) sebagai dasar dari arsitektur ini. Arsitektur ini terdiri dari 2 bagian besar, yaitu *Generator* dan *Discriminator*. Bagian *Generator* terdiri dari jaringan generatif yang mencoba untuk menghasilkan gambar tanpa hujan semirip mungkin dengan gambar aslinya, sedangkan pada bagian *Discriminator* akan memvalidasi



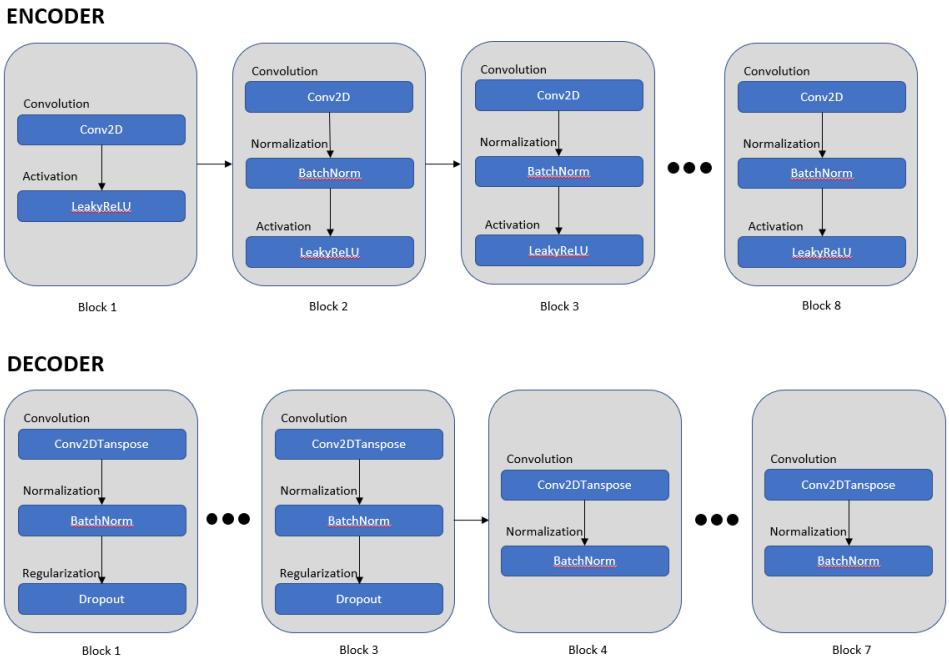
Gambar. 3. Arsitektur *CapsuleGAN*

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

apakah gambar yang dihasilkan oleh *Generator* sudah terlihat asli. Kerangka keseluruhan model yang diusulkan ditunjukkan pada Gambar 3. Arsitektur DCGAN "U-Net" hanya terdiri dari *convolutional layer* pada bagian *Generator* maupun *Discriminator*-nya, pengembangan yang kami usulkan terdapat pada bagian *Discriminator* yaitu dengan menggabungkan arsitektur DCGAN "U-Net" dengan CapsNet, bagian yang dikembangkan ditandai dengan kotak bergaris yaitu bagian *PrimaryCaps Layer* dan *FullyConnectedCaps Layer*.

1) Model Generator

Model *Generator* yang kami gunakan mengimplementasikan model yang sama dengan Pix2Pix [32] dibagi menjadi 2 bagian besar, yaitu *encoder* dan *decoder*. Pada bagian *encoder*, gambar *input* diteruskan ke *hidden layer* yang terdiri dari beberapa blok *convolutional layer*, masing-masing (kecuali *layer* pertama) blok terdiri dari *convolutional layer* yang kemudian diteruskan ke *layer batch normalization* dan diaktifkan menggunakan *Leaky ReLu*. Sebelum melanjutkan ke bagian *decoder*, ada satu *layer bottleneck* yaitu *layer downsample* terakhir untuk menghubungkan antara bagian *encoder* dan *decoder*, pada *layer* ini digunakan arsitektur blok yang sama seperti bagian *encoder*. Selanjutnya, pada bagian *decoder*, gambar akan di rekonstruksi kembali dengan menggunakan *transposed convolutional layer* sehingga ukuran panjang dan lebar gambar sesuai dengan *input*. Masing-masing *transposed convolutional layer* diikuti oleh *batch normalization layer* dan *layer* pertama diberikan *dropout* sebesar 0,5 untuk mencegah model yang dibuat mengalami *overfitting*. Gambar yang direkonstuksi pada bagian *decoder* merupakan hasil dari *de-raining* yaitu gambar tanpa hujan. Arsitektur lebih jelas model *Generator* digambarkan pada Gambar 4.



Gambar. 4. Arsitektur *Encoder* dan *Decoder* pada *Generator*

2) Model Discriminator dengan Capsule Network

Model *Discriminator* memiliki tugas sebagai pengklasifikasi gambar antara gambar asli tanpa hujan (*ground truth*) dengan hasil gambar yang diproduksi oleh model *Generator*. Pada umumnya, arsitektur *Discriminator* terdiri dari *convolutional layer* yang mana hanya berfokus pada *local neighbourhood*. Fitur yang diekstrak menggunakan *convolutional layer* tidak memperhatikan informasi spasial antara hubungan bagian dan keseluruhan (*part-to-whole*) atau dalam kata lain konteks antara komponen hujan dan gambar latar belakang, hal ini menyebabkan model mengenali garis-garis hujan hanyalah sebuah *noise* acak yang tidak berpola, tetapi pada kenyataannya garis-garis hujan tersebut memiliki pola. Disinilah peran *capsule layer* untuk menangkap informasi spasial antara komponen hujan dan gambar latar belakang.

Penggunaan *capsule layer* melengkapi kekurangan *convolutional layer* dengan mengganti nilai keluaran dari skalar menjadi nilai vektor atau matriks sehingga dapat menangkap keberagaman pola bentuk dan arah komponen hujan pada gambar. Algoritma *dynamic routing* menghitung kesamaan antara kapsul yang diberikan dengan kapsul yang berhubungan, jika kapsul tersebut memiliki kesamaan yang lebih besar, *transformation matrices* yang terhubung dengan unit-unit kapsul akan diberikan *weight* lebih besar, menangkap fitur dengan ukuran berbeda. Penerapan *capsule layer* pada bagian *Discriminator* dapat membedakan antara gambar asli atau palsu secara global maupun lokal, karena *capsules* dapat mengenali hubungan *part-to-whole* dengan baik.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

Panjang *output* vektor dari sebuah kapsul menunjukkan probabilitas adanya entitas yang hadir pada gambar. Penggunaan fungsi non-linear yang disebut dengan *squash function* digunakan untuk memastikan vektor yang pendek menyusut hingga mendekati 0 dan vektor yang panjang menyusut hingga mendekati 1. Persamaan (x) menunjukkan perhitungan *squash function*, dengan v_j adalah vektor *output* dari kapsul ke-*j* dan s_j adalah total input.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|^2} \tag{1}$$

Berbeda dengan arsitektur *Discriminator* yang diimplementasikan pada Pix2Pix [32] yang hanya menggunakan *convolutional layer*, pengembangan model *Discriminator* yang kami bangun terdiri dari *convolutional layer*, *primary capsule layer*, dan *fully-connected capsule layer*. Pada bagian *convolutional layer*, *Discriminator* akan menganalisis masing-masing gambar secara *pixel-to-pixel*, kemudian diteruskan ke bagian *primary capsule layer* untuk memetakan *convolutional layer* sebelumnya menjadi pemetaan dimensi kapsul, dengan begitu *transformation matrices* vektor didapatkan. Pada bagian ini bertujuan untuk mengekstrak fitur-fitur penting yang terdapat pada gambar, seperti pemetaan, posisi, dan kepadatan garis-garis hujan. Kemudian terakhir pada bagian *fully-connected capsule layer* akan menggabungkan *layer-layer* sebelumnya dan mengelurkan *output* berupa 2 unit kapsul yang dapat mengklasifikasi apakah *input* yang diterima asli atau palsu.

Karena bagian *discriminator* menggunakan *capsule network*, *loss function* yang digunakan pada bagian *discriminator* juga perlu disesuaikan. Kami menggunakan *margin loss function* sesuai dengan yang didiskusikan pada [27], penggunaan *loss function* ini berlandaskan *output* yang berupa vektor perlu dihitung panjang vektor yang didapat untuk mengetahui probabilitas entitas tersebut hadir. *Margin Loss* dinotasikan pada persamaan (2):

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda\,(1 - T_k)\ \max(0, \|v_k\| - m^-)^2 \tag{2}$$

dimana T_k adalah label target (1 menunjukkan ada entitas yang muncul, sedangkan 0 sebaliknya), v_k adalah vektor setiap kapsul pada *layer* terakhir, m^+ dan m^- mewakili margin untuk prediksi positif dan negatif, dan λ adalah faktor penurunan bobot untuk kapsul yang tidak ada.

D. Evaluasi Metrik

1) Peak Signal-to-Noise Ratio (PSNR)

Evaluasi metrik PSNR mengukur rasio antara nilai maksimal sinyal pada gambar asli dengan nilai distorsi yang mempengaruhi kualitas gambar [33]. Rasio pengukuran PSNR menggunakan satuan desibel (dB). Metrik PSNR dinotasikan pada persamaan (3).

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \tag{3}$$

$$MAX = 2^{jumlah\ bit} - 1 \tag{4}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{5}$$

Pada persamaan (4), MAX merupakan nilai maksimal piksel yang termuat dalam gambar. Sebagai contoh, bila gambar yang diukur merupakan gambar 8-bit, maka nilai maksimalnya adalah 255. Nilai MAX piksel gambar tersebut kemudian dibagi dengan nilai *Mean Squared Error* (MSE) ditunjukkan pada persamaan (5). Sesuai dengan namanya, MSE menghitung rata-rata pangkat dua dari *error*, *error* yang dimaksud merupakan perbedaan nilai yang didapat dan nilai yang diprediksi. Metrik evaluasi PSNR yang lebih besar menandakan gambar yang dihasilkan lebih menyerupai gambar aslinya, dengan begitu semakin besar nilai PSNR, maka semakin baik model yang dibuat.

2) Structural Similiarity Index (SSIM)

Evaluasi metrik SSIM merupakan alat pengukuran terbaru yang dirancang berdasarkan tiga faktor yaitu pencahayaan, kontras, dan struktur agar lebih sesuai dengan cara kerja sistem visual manusia [34]. Rentang nilai metrik evaluasi SSIM dimulai dari 0 dan maksimalnya adalah 1, semakin besar nilai SSIM, semakin baik struktur gambar yang dihasilkan. Persamaan SSIM dinotasikan pada persamaan (6).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x{}^2 + \mu_y{}^2 + C_1)(\sigma_x{}^2 + \sigma_y{}^2 + C_2)} \tag{6}$$

Persamaan (6) merupakan bentuk penggabungan antara perhitungan dari faktor pencahayaan, kontras, dan

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

struktur gambar, dengan μ_x dan μ_y adalah rata-rata lokal, σ_x dan σ_y merupakan standar deviasi, dan C merupakan nilai konstanta.

III. HASIL DAN PEMBAHASAN

Pelatihan model dilakukan menggunakan *library Tensorflow* pada GPU NVIDIA T4, dengan *epoch* sebanyak 25 kali dan *learning rate* diatur sebesar $3e-4$ untuk *Generator* dan $1e-4$ untuk *Discriminator* menggunakan *optimizer Adam*. Evaluasi performa model diukur menggunakan sistem evaluasi metrik *Peak Signal-to-Noise Ratio* (PSNR) dan *Structural Similiarity Index* (SSIM) Kami membandingkan performa model yang telah dibuat dengan penelitian-penelitian sebelumnya, yaitu GMM [18], CNN [23], JORDER [35], DDN [4], dan CGAN [26].

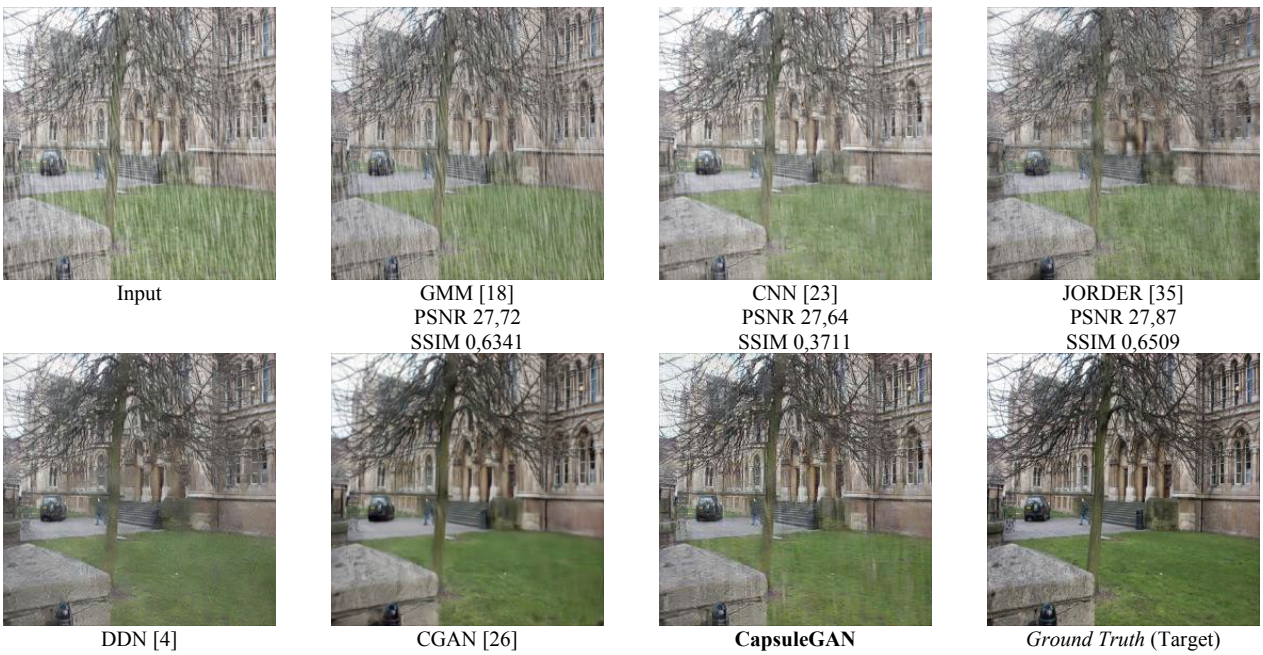
Evaluasi dilakukan dengan membandingkan rata-rata nilai-nilai evaluasi metrik pada gambar hujan antara model yang dibuat dan model-model dari penelitian lain. Gambar hujan yang digunakkan untuk evaluasi adalah gambar hujan sintetis, untuk gambar hujan asli tanpa sintetis kami gunakan evaluasi secara visual karena tidak ada gambar pasangan tanpa hujan yang membuat evaluasi metrik tidak dapat digunakan. Tabel I menunjukkan hasil perbandingan kuantitatif model yang diusulkan dengan 5 model lain yang menggunakan metode berbeda. Nilai yang dibandingkan adalah rata-rata nilai metrik evaluasi PSNR dan SSIM pada *dataset* tes *Rain1200*. Nilai yang dicetak tebal merupakan nilai terbaik dari evaluasi metrik tersebut, sedangkan nilai yang digaris bawahi merupakan terbaik kedua. Hasil perbandingan menunjukkan bahwa model CapsuleGAN yang kami usulkan paling unggul dalam metrik evaluasi PSNR, namun metrik evaluasi SSIM menunjukkan bahwa model CGAN dan DDN lebih baik.

Gambar tes yang kami gunakan sebagai perbandingan diambil dari penelitian model *de-raining* CGAN [26] untuk hasil perbandingan yang seimbang.

TABEL I
PERBANDINGAN KUANTITATIF EVALUASI METRIK DENGAN DATASET RAIN1200 [31]

Model	PSNR	SSIM
GMM [18]	22,27	0,7413
CNN [23]	19,12	0,6013
JORDER [35]	21,09	0,7525
DDN [4]	23,28	<u>0,8208</u>
CGAN [26]	<u>24,34</u>	**0,8430**
Model yang diusulkan (CapsuleGAN)	**25,43**	0,803

Pada gambar 5 perbandingan dengan model lain, model CapsuleGAN unggul dalam metrik evaluasi SSIM

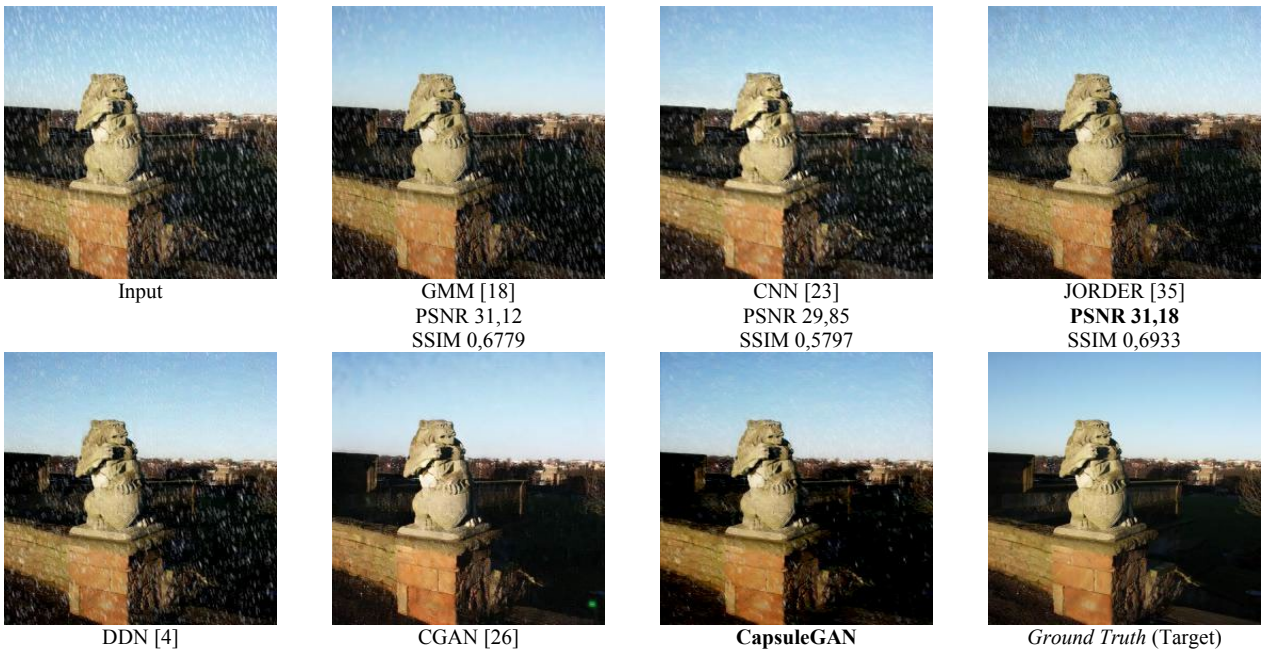


Gambar. 5. Perbandingan Kualitatif Hasil *De-raining* Pada Gambar *Test* 1

dengan nilai terbaik yaitu 0,7842, namun pada metrik evaluasi PSNR masih kalah dengan model CGAN dengan perbedaan tipis yaitu sebesar 0,68. Secara inspeksi visual, model CapsuleGAN masih meninggalkan beberapa

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

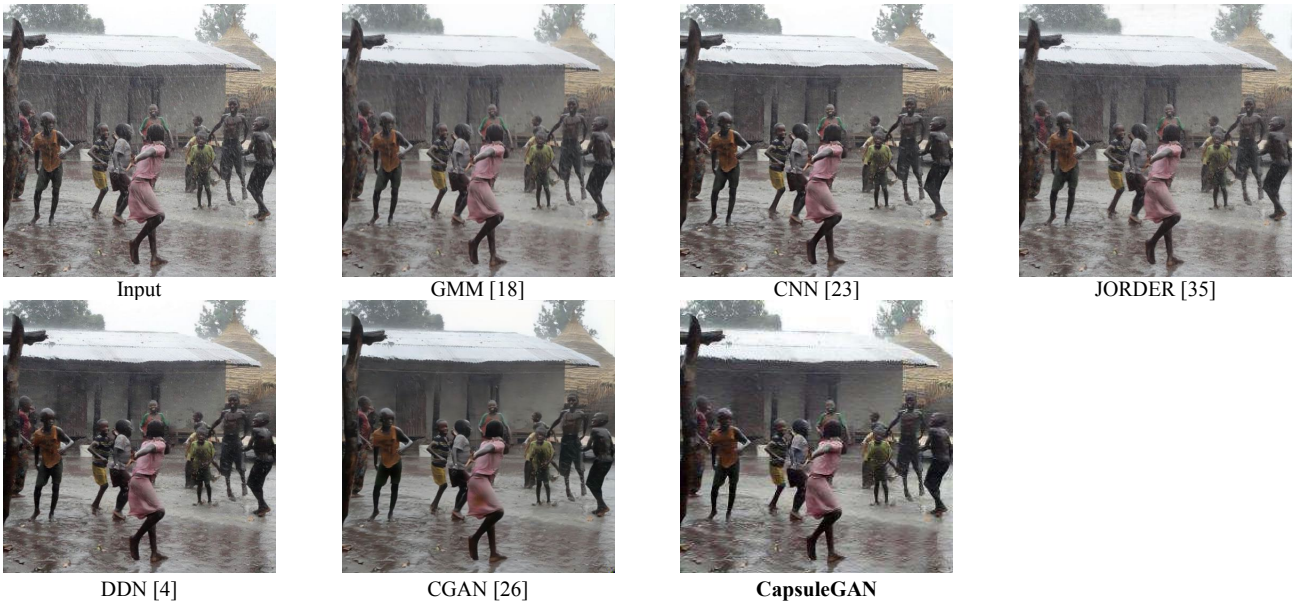
garis-garis hujan jika dibandingkan dengan model CGAN, tetapi dari bentuk struktural gambar, model CapsuleGAN menghasilkan gambar yang lebih tajam dibandingkan dengan model CGAN, hal tersebut dapat dilihat pada bagian jendela dan model, model CGAN memberikan efek halus terhadap gambar, sedangkan model CapsuleGAN memberikan hasil lebih tajam yang menyerupai gambar target.



Gambar. 6. Perbandingan Kualitatif Hasil *De-raining* Pada Gambar *Test* 2

Gambar 6 menunjukkan secara langsung bagaimana hasil inspeksi gambar secara visual dengan membandingkan *state-of-the-art* masing-masing model yang dibandingkan. Model CapsuleGAN mendapatkan nilai PSNR sebesar 29,35 dan SSIM sebesar 0,6835. Dapat dilihat bahwa dibandingkan dengan model *de-raining* GMM, CNN, JORDER, dan DNN yang masih menyisakan air hujan dan terdapat efek *blur* pada gambar, model CGAN dan model yang kami buat memberikan hasil lebih baik dengan menghasilkan gambar yang lebih tajam serta minim rintik hujan. Meskipun sudah unggul daripada model lain, model yang kami buat masih kalah dalam perbandingan dengan model CGAN, model yang kami buat masih menyisakan sedikit *noise* berwarna putih dari rintik hujan.

Perbandingan hasil *de-raining* juga perlu dilakukan pada data gambar hujan asli. Gambar 7 menunjukkan perbandingan hasil *de-raining* pada data gambar hujan asli, model CapsuleGAN berhasil menghasilkan gambar

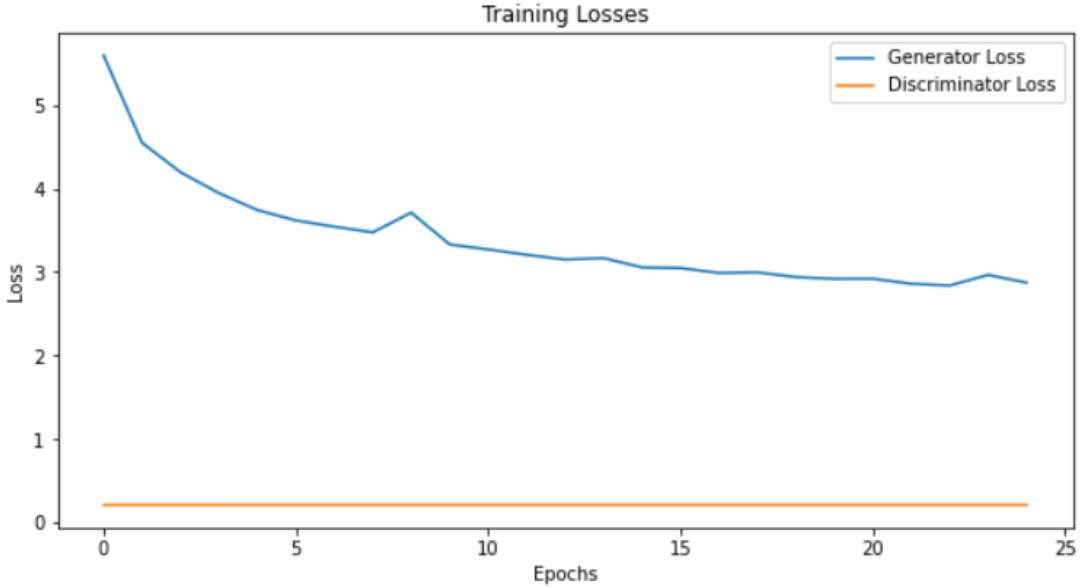


Gambar. 7. Perbandingan Kualitatif Hasil *De-raining* Pada *Dataset* Hujan Asli

yang lebih tajam dibandingkan model yang lain, ditunjukkan dengan warna yang dihasilkan lebih cerah dibandingkan model lain yang masih terpengaruh oleh efek hujan sehingga terlihat agak pucat.

Beberapa evaluasi yang telah dilakukan menunjukkan model CapsuleGAN yang dibuat mengungguli model-model deep learning GMM [18], CNN [23], JORDER [35], dan DDN [4]. Tetapi pada beberapa aspek, model CGAN [26] lebih unggul daripada model CapsuleGAN. Penelitian CGAN menggunakan arsitektur dasar yang sama seperti yang kami gunakan, mereka menggunakan arsitektur DCGAN "U-Net" seperti yang dibahas pada bagian metode penelitian, yang membedakan penelitian mereka adalah penggunaan loss function yang dikustomisasi. Hasil gambar yang dihasilkan oleh model CGAN lebih minim hujan dibandingkan dengan yang kami buat, hal tersebut dapat terjadi karena loss function yang digunakan cocok dengan arsitektur CGAN. Pada sisi lain, meskipun model kami masih menyisakan hujan pada gambar, gambar yang kami hasilkan lebih tajam dibandingkan model CGAN, hal tersebut karena pengembangan arsitektur dasar yang kami usulkan dengan menambahkan CapsNet pada bagian Discriminator.

Evaluasi lain untuk mengukur performa model dilakukan dengan memonitor setiap *loss* selama iterasi yang dihasilkan dari masing-masing model *Generator* dan *Discriminator*. Gambar 8 menunjukkan rangkuman *loss*



Gambar. 8. Grafik *Generator Loss* dan *Discriminator Loss*

selama iterasi bahwa *loss Discriminator* lebih stabil dibandingkan dengan *loss Generator* selama pelatihan, hal tersebut dapat dipengaruhi oleh beberapa faktor, diantaranya *gradient vanishing, mode collapse,* dan data pelatihan yang kurang [36]. Hal ini menunjukkan bahwa model *Generator* dan *Discriminator* yang telah dibuat kurang seimbang. *Loss Discriminator* yang sangat rendah merupakan tanda bahwa model *Discriminator* yang dibuat sangat handal dalam mengklasifikasikan gambar, hal tersebut membuat model *Generator* sulit untuk mengelabui *Discriminator*, sehingga *Generator* tidak dapat menghasilkan gambar yang lebih baik lagi, ditandai dengan Loss *Generator* yang tinggi.

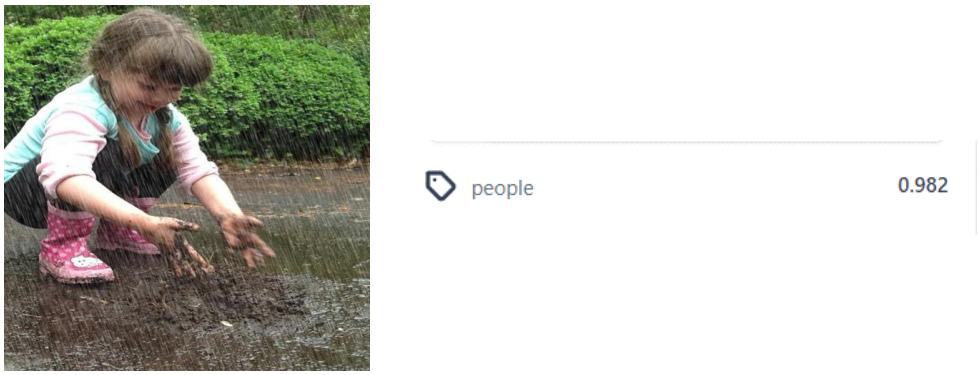
Pada Tabel II menunjukkan *loss* lengkap setiap iterasi dalam pelatihan. Meskipun *Loss Generator* mengalami penurunan dari iterasi ke-1 sampai ke-16, pada iterasi ke-17 sampai seterusnya *Loss Generator* mulai menunjukkan hasil yang kurang signifikan ditandai dengan kemiringan nilai penuruan yang kurang dari 0.01 (Gambar 6a), dan pada *epoch* ke-24 ada lonjakan *Loss Generator* yang cukup signifikan (Gambar 6b), hal tersebut menandakan *Generator* kesulitan untuk memproduksi gambar yang lebih baik.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 2, Juni 2024, Pp. 1001-1012

TABEL II
GENERATOR LOSS DAN DISCRIMINATOR LOSS

Epoch	Generator Loss	Discriminator Loss
1	5,5937	0,2135
2	4,5477	0,2136
3	4,1951	0,2136
4	3,9449	0,2135
5	3,7424	0,2135
6	3,6180	0,2135
7	3,5433	0,2135
8	3,4754	0,2135
9	3,7115	0,2135
10	3,3311	0,2135
11	3,2711	0,2134
12	3,2073	0,2134
13	3,1501	0,2134
14	3,1677	0,2134
15	3,0562	0,2134
16	3,0482	0,2134
17	2,9901	0,2134
18	2,9967	0,2134
19	2,9409	0,2134
20	2,9195	0,2134
21	2,9207	0,2133
22	2,8609	0,2133
23	2,8394	0,2133
24	2,9664	0,2133
25	2,8740	0,2133

Sebagai bukti bahwa penggunaan model *deraining* yang kami bangun dapat meningkatkan kualitas gambar agar dapat digunakan pada algoritma CV. Kami menguji perbandingan keterbacaan gambar sebelum dan setelah dilakukan *deraining*. Pengujian ini dilakukan menggunakan API Clarifai (https://www.clarifai.com/) sebagai platform penerapan algoritma CV. Sistem pengujian ini adalah memberikan *input* gambar ke algortima CV, kemudian algoritma CV akan mengeluarkan hasil deteksi berupa klasifikasi dan nilai probabilitas gambar tersebut masuk dalam klasifikasi yang terdaftar. *Treshold* klasifikasi diatur sebesar 0,98 untuk menghindari terlalu banyaknya kelas yang muncul pada hasil klasifikasi dan menunjukkan kelas yang paling cocok dengan konteks gambar.



a) Hasil Deteksi Pada Gambar Sebelum Proses *De-raining*

b) Hasil Deteksi Pada Gambar Sebelum Proses *De-raining*

Gambar. 9. Evaluasi Model Pada Penerapan Algoritma *Computer Vision*

Pada Gambar 9 dapat dilihat bahwa penggunaan model *deraining CapsuleGAN* dapat meningkatkan keterbacaan gambar dengan dibuktikannya kenaikan nilai probabilitas klasifikasi. Gambar 8a menunjukkan hasil klasifikasi gambar hujan yang hanya masuk pada kelas "*people*" dengan nilai 0,982. Pada gambar 8b yang merupakan hasil *de-raining* menunjukkan ada kenaikan nilai dan kelas yang bertambah. Kenaikan nilai pada kelas "*people*" sebesar 0,006 menjadi 0,988 dan 1 kelas tambahan "*child*" dengan nilai 0,996.

Dari beberapa hasil evaluasi model, didapatkan bahwa model CapsuleGAN yang diusulkan masih memiliki beberapa batasan. Pelatihan model yang tidak stabil menjadi masalah utama pada penelitian ini. Beberapa kemungkinan yang menyebabkan tidak stabilnya proses pelatihan dapat terjadi karena kurangnya data pelatihan, arsitektur model yang tidak seimbang, dan pengaturan pelatihan model yang tidak sesuai. Model CapsuleGAN yang kami kembangkan masih merupakan bentuk sederhana yaitu dengan menggabungkan dua arsitektur CapsNet dan GAN tanpa banyak menyesuaikan jenis dan jumlah layer yang digunakan. Penggunaan arsitektur CapsNet pada penelitian ini mengikuti penelitian utama yang dilakukan oleh [27] sehingga perlu adanya penyesuaian lebih lanjut untuk membuat arsitektur model yang lebih seimbang. Selain itu, pengaturan pelatihan model seperti penggunaan loss function dan optimizer perlu diteliti lebih lanjut. Loss Function sederhana yang kami gunakan pada penelitian ini belum efektif untuk digunakan pada kasus penghapusan hujan pada gambar, sehingga selain pengembangan arsitektur model yang digunakan, pengembangan loss function yang digunakan juga berperan penting.

IV. KESIMPULAN

Pada penelelitian ini kami mengusulkan sebuah pengembangan model *image de-raining* menggunakan *Generative Adversarial Network* (GAN). Untuk mengembangkan pemetaan hujan menjadi lebih baik, kami menambahkan *Capsule Network* pada bagian *Discriminator* untuk mempertahankan informasi-informasi spasial antar komponen hujan sehingga model yang dibuat dapat mengenali karakteristik garis-garis hujan seperti posisi dan kepadatannya, dengan begitu garis-garis hujan akan lebih mudah dipetakan. Experimen dilakukan dengan membandingkan *state-of-the-art* model secara evaluasi metrik menggunakan dataset sintetis, inspeksi visual menggunakan gambar hujan asli (*real world*), evaluasi *loss* model yang dibuat, dan evaluasi penerapan algoritma *Computer Vision*. Perbandingan menggunakan data sintetis menunjukkan model yang kami buat lebih unggul daripada kebanyakan model *deep learning* serupa, namun hasil gambar dan evaluasi metrik masih menunjukkan bahwa model CapsuleGAN masih kalah dibandingkan dengan model CGAN karena masih menyisakan *noise* putih pada gambar. Pada perbandingan dengan dataset asli, model CapsuleGAN lebih unggul dibandingkan dengan model yang lain karena berhasil menghilangkan rintik hujan sekaligus menghasilkan gambar yang lebih tajam. Evaluasi pada monitoring *loss* model, model CapsuleGAN masih belum stabil karena *loss* model *Generator* yang masih tinggi dan *loss* model *Discriminator* yang sangat rendah, menandakan bahwa adanya ketidakstabilan pada model yang dibuat. Namun, dari hasil perbandingan evaluasi gambar, model yang kami buat lebih unggul daripada kebanyakan model *deep learning* lainnya, yang berarti model CapsuleGAN cukup berhasil membuktikan bahwa penggunaan *Capsule Network* pada GAN berpengaruh untuk pembuatan model *de-raining*. Selain itu, pada evaluasi penerapan algoritma *Computer Vision*, model CapsuleGAN dapat membuat *classifier* menangkap lebih banyak informasi pada gambar yang sudah melewati proses *de-raining.*

Meskipun hasil penelitian sudah menunjukkan hasil yang cukup memuaskan, tetapi masih ada beberapa faktor yang dapat dikembangkan dari penelitian ini untuk mendapatkan hasil yang lebih baik. Pengembangan lebih lanjut dari model yang paling utama adalah menstabilkan proses *training* model. Pengembangan ini dapat dilakukan melalui variabel-variabel hampir dari semua aspek, meliputi dataset yang lebih beragam, penggunaan model *Generator* dan *Discriminator* yang lebih ringan, pembuatan *Loss Function* yang dikustomisasi, dan pemilihan *Optimizer* yang lebih sesuai. Masing-masing variabel dapat mempengaruhi kinerja model yang dibuat, seperti waktu komputasi yang lebih cepat dan hasil kualitas gambar yang lebih baik. Selain daripada itu, pengembangan model ini dapat mengarah kepada *video de-raining*. Hal tersebut dapat bermanfaat untuk kasus penghapusan hujan secara *real-time*, seperti penggunaan dalam *autonomous vehicle* dan *surveilance system*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Badan Riset dan Inovasi Nasional (BRIN) yang telah memberi dukungan *financial* terhadap penelitian ini. Penelitian ini merupakan bagian dari project "Sistem Pendeteksian Objek dan Telekomunikasi untuk Kendaraan Listrik Otonom" milik BRIN.

DAFTAR PUSTAKA

[1] X. Jiao, Y. Liu, J. Gao, X. Chu, R. Liu, and X. Fan, "PEARL: Preprocessing Enhanced Adversarial Robust Learning of Image Deraining for Semantic Segmentation," May 2023, [Online]. Available: http://arxiv.org/abs/2305.15709

[2] Y. Gou, P. Hu, J. Lv, J. T. Zhou, and X. Peng, "Multi-Scale Adaptive Network for Single Image Denoising," *Adv Neural Inf Process Syst*, vol. 35, pp. 14099–14112, Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.04313

[3] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive Image Deraining Networks: A Better and Simpler Baseline," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3937–3946, 2019, [Online]. Available: https://github.com/csdwren/PReNet.

[4] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3855–3863, 2017.

[5] S. Li *et al.*, "Single Image Deraining: A Comprehensive Benchmark Analysis," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3838–3847, 2019, [Online]. Available: https://github.com/lsy17096535/Single-Image-Deraining

[6] H. Zhang and V. M. Patel, "Convolutional sparse & low-rank coding-based rain streak removal," in *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, Institute of Electrical and Electronics Engineers Inc., May 2017, pp. 1259–1267. doi: 10.1109/WACV.2017.145.

[7] X. Chen, J. Pan, J. Lu, Z. Fan, and H. Li, "Hybrid CNN-Transformer Feature Fusion for Single Image Deraining," *In Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, pp. 378–386, 2023, [Online]. Available: www.aaai.org

[8] X. Zhang, H. Li, Y. Qi, W. Kheng Leow, and T. Khim Ng, "Rain Removal In Video By Combining Temporal And Chromatic Properties," *IEEE International Conference on Multimedia and Expo*, 2006.

[9] A. K. Tripathi and S. Mukhopadhyay, "Removal of rain from videos: a review," *Signal Image Video Process*, vol. 8, no. 8, pp. 1421–1430, Nov. 2014, doi: 10.1007/s11760-012-0373-6.

[10] V. Santhaseelan and V. K. Asari, "Utilizing Local Phase Information to Remove Rain from Video," *Int J Comput Vis*, vol. 112, no. 1, pp. 71–89, Mar. 2015, doi: 10.1007/s11263-014-0759-8.

[11] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A Novel Tensor-based Video Rain Streaks Removal Approach via Utilizing Discriminatively Intrinsic Priors," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] J. Bossu, N. Hautière, and J. P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *Int J Comput Vis*, vol. 93, no. 3, pp. 348–367, Jul. 2011, doi: 10.1007/s11263-011-0421-7.

[13] J. H. Kim, J. Y. Sim, and C. S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015, doi: 10.1109/TIP.2015.2428933.

[14] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video Desnowing and Deraining Based on Matrix Decomposition," *n Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4210–4219, 2017.

[15] Y. L. Chen and C. T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., 2013, pp. 1968–1975. doi: 10.1109/ICCV.2013.247.

[16] L. W. Kang, C. W. Lin, and Y. H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012, doi: 10.1109/TIP.2011.2179057.

[17] K. Jiang *et al.*, "Multi-Scale Progressive Fusion Network for Single Image Deraining," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8345–8355, Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.10985

[18] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain Streak Removal Using Layer Priors," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2736–2744, 2016.

[19] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," *Proceedings of the IEEE international conference on computer vision*, pp. 3397–3405, 2015.

[20] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight Pyramid Networks for Image Deraining," *IEEE Trans Neural Netw Learn Syst*, vol. 31, no. 6, pp. 1794–1807, May 2018, [Online]. Available: http://arxiv.org/abs/1805.06173

[21] X. Chen, H. Li, M. Li, and J. Pan, "Learning A Sparse Transformer Network for Effective Image Deraining," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5896–5905, 2023, [Online]. Available: https://github.

[22] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised Transfer Learning for Image Rain Removal," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3877–3886, 2019, [Online]. Available: https://www.photoshopessentials.com/photo-effects/rain/

[23] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017, doi: 10.1109/TIP.2017.2691802.

[24] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive Generative Adversarial Network for Raindrop Removal from a Single Image," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2482–2491, Nov. 2017, [Online]. Available: http://arxiv.org/abs/1711.10098

[25] R. Li, L.-F. Cheong, and R. T. Tan, "Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning *," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1633–1642, 2019.

[26] H. Zhang, V. Sindagi, and V. M. Patel, "Image De-raining Using a Conditional Generative Adversarial Network," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 11, pp. 3943–3956, Jan. 2017, [Online]. Available: http://arxiv.org/abs/1701.05957

[27] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," *Adv Neural Inf Process Syst*, vol. 30, Oct. 2017, [Online]. Available: http://arxiv.org/abs/1710.09829

[28] E. Xi, S. Bing, and Y. Jin, "Capsule Network Performance on Complex Data," *arXiv preprint*, Dec. 2017, [Online]. Available: http://arxiv.org/abs/1712.03480

[29] A. Jaiswal, W. Abdalmageed, Y. Wu, and P. Natarajan, "CapsuleGAN: Generative Adversarial Capsule Network," *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, [Online]. Available: https://github.com/hindupuravinash/the-gan-zoo

[30] C. Xiang, M. Su, C. Zhang, F. Wang, M. Yang, and Z. Niu, "E-CapsGan: Generative adversarial network using capsule network as feature encoder," *Multimed Tools Appl*, vol. 81, no. 18, pp. 26425–26442, Jul. 2022, doi: 10.1007/s11042-022-12279-3.

[31] H. Zhang and V. M. Patel, "Density-aware Single Image De-raining using a Multi-stream Dense Network," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 695–704, 2018, [Online]. Available: https://github.com/hezhangsprinter/DID-MDN

[32] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.07004

[33] U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *Journal of Computer and Communications*, vol. 07, no. 03, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.

[34] D. R. I. M. Setiadi, "PSNR vs SSIM: imperceptibility quality assessment for image steganography," *Multimed Tools Appl*, vol. 80, no. 6, pp. 8423–8444, Mar. 2021, doi: 10.1007/s11042-020-10035-z.

[35] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep Joint Rain Detection and Removal from a Single Image," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1357–1366, Sep. 2017, [Online]. Available: http://arxiv.org/abs/1609.07769

[36] A. Jabbar, X. Li, and B. Omar, "A Survey on Generative Adversarial Networks: Variants, Applications, and Training," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–49, 2021.