Translasi Citra Malam Menjadi Siang Menggunakan *Deep Convolutional Generative Adverserial Network*

Pratama Yoga Santosa[1], Ema Rachmawati[2], Tjokorda Agung Budi Wirayuda[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung

[1]tagato@student.telkomuniversity.ac.id, [2]emarachmawati@telkomuniversity.ac.id, [3]cokagung@telkomuniversity.ac.id

Abstrak

Banyak penelitian yang sedang berfokus untuk pengolahan citra pada keadaan cahaya rendah atau minim, agar bisa menghasilkan citra yang bagus dan jelas. Maka penelitian ini bertujuan untuk mentranslasikan citra yang diambil pada keadaan minim cahaya atau pada malam hari sekalipun dapat menghasilkan suatu citra yang jelas atau seperti citra dengan kualitas cahaya bagus atau diambil pada kondisi siang hari. Untuk mewujudkannya, penelitian ini menggunakan dua kategori dataset, yaitu citra yang diambil dengan dengan keadaan siang dan dataset dengan keadaan malam hari yang kemudian dataset tersebut dilatih menggunakan DCGAN (*Generative Adverserial Network*). Dengan metode ini, mesin akan dilatih dengan masukan awal adalah citra malam, kemudian akan masuk ke *generator* GAN untuk selanjutnya diproses sehingga menghasilkan suatu citra siang hari dan kemudian dibandingkan, apakah hasilnya sudah mirip dengan citra siang hari yang terdapat pada *discriminator* GAN. Kemudian model dievaluasi dengan menghitung nilai dari SSIM atau akurasi dan nilai *loss*-nya menggunakan L2 untuk menentukan apakah performa DCGAN yang dibangun telah mendapat hasil yang baik atau tidak.

Kata kunci: *computer vision*, DCGAN, *discriminator*, *generator*, citra, dataset

Abstract

A lot of research is currently focused on image processing in low or low light conditions, in order to produce good and clear images. So this study aims to translate images taken in low light conditions or even at night to produce a clear image or like an image with good light quality or taken in daytime conditions. To make it happen, this study uses two categories of datasets, namely images, which is taken with day conditions and datasets with night conditions which are then trained using DCGAN (Generative Adverserial Network). With this method, the machine will be trained with the initial input is a night image, then it will enter the GAN generator for further processing so that it produces a daytime image and then compare, whether the results are similar to the daytime image found on the GAN discriminator. Then the model is evaluated by calculating the value of the SSIM or its loss value using L2 to determine whether the DCGAN performance that was built has got good results or not.

Keywords: computer vision, DCGAN, discriminator, generator, image, dataset

1. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi yang sangat pesat di zaman sekarang sangat mempengaruhi berbagai bidang, terutama pengolahan citra. Banyak orang berlomba – lomba membuat suatu produk *computer vision* khususnya dalam pengolahan citra untuk membantu pekerjaan manusia. *Computer vision* merupakan salah satu bidang dari *artificial intelligence* yang berhubungan tentang pemrosesan atau transformasi kamera diam atau bergerak yang digunakan untuk menghasilkan suatu keputusan atau representasi baru yang dilakukan untuk mencapai tujuan tertentu [1]. Pada era sekarang perkembangan *computer vision* sendiri sangat pesat, salah satu contoh terbesarnya adalah *autonomous car* yang di dalamnya terdapat berbagai macam metode kompleks dari *computer vision* [2] yang kemudian dijadikan satu dan menghasilkan produk tersebut.

Tetapi masalah muncul apabila mobil tersebut tidak bisa berkendara di dalam gelap atau minim cahaya. Karena dalam praktiknya, pengenalan tempat bisa terhambat atau terganggu dengan adanya perubahan tampilan

karena cuaca dan pencahayaan [3] sehingga dapat menyebabkan kecelakaan. Bukan hanya itu, translasi dari malam ke siang juga bisa diimplementasikan pada fotografi untuk mempermudah pengeditan, maka penelitian ini dapat digunakan untuk peningkatan kualitas *low-light* foto agar mendapat hasil foto yang lebih cerah [4].

Maka dari itu untuk mengatasi masalah tersebut, diperlukan sebuah sistem yang dapat melakukan translasi citra malam menjadi siang untuk membantu memecahkan masalah-masalah yang telah disebutkan. Hal ini juga didasarkan pada penelitian sebelumnya, dimana terdapat penelitian tentang translasi citra malam menjadi siang menggunakan model CAN (*Context Aggregation Network*) dan CycleGAN yang masing-masing dari penelitian tersebut menunjukan bahwa sistem yang bisa mentranslasi citra malam menjadi siang dapat dibangun. Karena kedua penelitian tersebut kami mengusulkan sebuah sistem translasi yang sama menggunakan model DCGAN untuk kasus *supervised image*.

1.2 Topik dan Batasan

Dalam penelitian ini, terdapat beberapa rumusan masalah seperti berikut :

1. Bagaimana cara untuk membangun suatu sistem pada mesin maupun computer bisa mentranslasikan citra malam menjadi siang.
2. Parameter apa saja yang mempengaruhi DCGAN tersebut bisa menghasilkan hasil dan performa yang baik.

Kemudian berikut adalah batasan masalah dalam dilakukannya penelitian ini, yaitu sebagai berikut:

1. Jumlah dataset yang digunakan hanya sebatas 6613 citra siang dan 6613 citra malam yang identik, hanya berbeda masalah waktu pengambilan.
2. Dalam penelitian ini hanya melingkupi translasi citra pada ruangan outdoor.
3. Dataset yang dibangun diambil tanpa menggunakan pencahayaan tambahan, hanya mengandalkan pencahayaan senatural mungkin.
4. Terdapat beberapa dataset yang diambil dengan mengekstrak frame video *timelapse* dan beberapa menggunakan kamera ponsel.
5. Pelatihan menggunakan citra dengan bentuk persegi.
6. Penelitian ini hanya berfokus merubah citra malam menjadi siang hari, bukan gelap menjadi terang.

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk membangun suatu sistem yang dapat mentranslasikan citra malam menjadi citra siang hari menggunakan DCGAN.

2. Studi Terkait

2.1 *Night to Day Translation*

Sebelumnya, terdapat beberapa penelitian yaitu *Night-to-Day Image Translation for Retrieval-based Localization (2019)* [3] dan *Learning to See in The Dark (2018)* [4]. Pada kedua penelitian tersebut, masing-masing menggunakan metode yang berbeda. Pada penelitian nomor dua, digunakan metode CAN [5] untuk mempercepat dan memberikan hasil lebih baik dalam *feature extraction*, dimana terdapat beberapa *processing* data, yaitu *high-low image enhancement*. Dimana dalam prosesnya, citra gelap akan dikonversi menjadi citra dengan ber-*noise* kemudian dikonversi menjadi citra yang terang. Pada penelitian ini, menghasilkan hasil yang cukup baik, yaitu nilai akurasi menggunakan SSIM sebesar 79%. Kemudian pada penelitian nomor 3, menggunakan metode TodayGAN [3], dimana dalam penelitian tersebut menggunakan Oxford *Robotcars Dataset* yang berisi video hasil rekaman *autonomous car* yang sedang dikerjakan oleh Oxford University, dari penelitian tersebut menghasilkan akurasi mencapai 52.9% menggunakan DenseVLAD [6]. Dari dua penelitian tersebut dibuktikan bahwa sistem translasi citra malam menjadi siang dapat dibangun menggunakan metode *deep learning.*

2.2 *Generative Adversarial Network*

GAN atau *Generative Adversarial Network* merupakan salah satu metode dalam *deep learning* yang menggunakan prinsip *unsupervised learning* [7]. Prinsip kerja dari GAN sendiri bisa dianalogikan seperti pemalsu uang (*generator*) membuat uang palsu, kemudian terdapat polisi (*discriminator)* yang bertugas mengecek apakah uang palsu atau bukan. Sehingga secara ilmiah cara kerja GAN adalah memiliki 2 model yang saling berkompetisi, dimana salah satu modelnya yaitu *generator* akan berusaha untuk membangkitkan suatu citra baru berdasarkan target, kemudian model lainnya yaitu *discriminator* akan mendeteksi apakah gambar yang dihasilkan oleh generator terdeteksi sebagai gambar (asli atau mirip seperti target) atau tidak. Kedua model ini akan meningkatkan kemampuannya hingga generator mampu menghasilkan gambar yang tidak dapat dibedakan dengan gambar targetnya [7].

2.3 *Convolutional Neural Network*

Convolutional Neural Network dianalogikan sebagai jaringan syaraf tiruan karena memiliki neuron yang mampu mengoptimalkan jaringan melalui pembelajaran. Setiap neuron akan menerima input dan melakukan

operasi perhitungan [8]. Namun perbedaannya, pada *Convolutional Neural Network* terdapat *hidden layer* yaitu *layer* diantara *input layer* dengan *output layer*. Pada *hidden layer* inilah pembelajaran dilakukan untuk mendapatkan bobot yang sesuai sehingga menghasilkan *error* yang minimum [9]. Sehingga dalam pembelajarannya akan terjadi pembaharuan bobot sehingga nantinya meningkatkan hasil akurasi dan mengurangi *error* yang terjadi selama pembelajaran.

2.4 *Batch Normalization*

Batch Normalization adalah mekanisme yang bertujuan untuk menstabilkan distribusi atau mengurangi distribusi input ke lapisan jaringan yang diberikan selama pelatihan [10] dengan rumus atau formula yang dapat dilihat sebagai berikut.

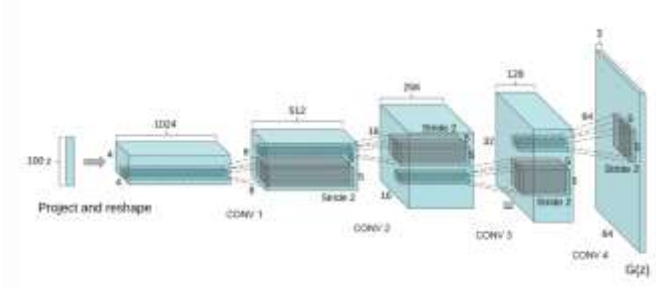
$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} X_i \ \ and \ \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(X_i - \mu_B)^2$$
$$\bar{X} = \frac{X_i^{(k)} - \mu_B^{(k)}}{\sqrt{\sigma_V^{(k)^2} + \epsilon}} \quad (1)$$

Metode ini berguna untuk mempercepat atau mengoptimalkan proses *training* [10]. Ini terjadi karena terjadi penghitungan matematis dengan membagi setiap nilai inputan (nilai rata-rata inputan(x_i) dikurangi nilai variansi inputan(μ_B) kemudian dibagi dengan standar deviasi (σ_B), sehingga akan mendapatkan skala nilai yang lebih kecil, sehingga lebih cepat untuk di proses. Dengan menggunakan metode ini kita bisa mengurangi penggunaan *dropout* pada arsitektur GAN, karena konsep pemakaian *batch normalization* ini hampir mirip dengan *dropout.*

2.5 *Deep Convolutional Generative Adversarial Network*

DCGAN merupakan metode GAN yang dikembangkan pertama kali pada tahun 2016. Metode ini adalah hasil penyempurnaan dari metode sebelumnya yaitu Convolutional GAN [11]. Metode ini dipilih pada penelitian ini, karena diharapkan model yang didapatkan pada penelitian ini bisa lebih dinamis dan *generative* dalam melakukan translasi citra, kemudian metode ini memiliki proses pelatihan yang lebih stabil dengan menghilangkan *fully connected layer* yang membuat jaringan bisa melakukan pembelajaran tambahan *downsampling* dengan spasial [11] yang bisa dilihat pada **Gambar 1**. Cara kerja DCGAN mirip dengan GAN biasa dengan menerapkan arsitektur dari CNN, akan tetapi ada beberapa perubahan dalam arsitekturnya sendiri [11], yaitu :

1. Mengganti pooling layer menjadi *convolutional stride*.
2. Menggunakan *batch normalization* pada *generator* dan *discriminator.*
3. Menghapus *fully connected layer.*
4. Menggunakan fungsi aktifasi ReLU pada semua layer kecuali untuk output layer menggunakan TanH.
5. Menggunakan LeakyReLU untuk semua layer *discriminator.*



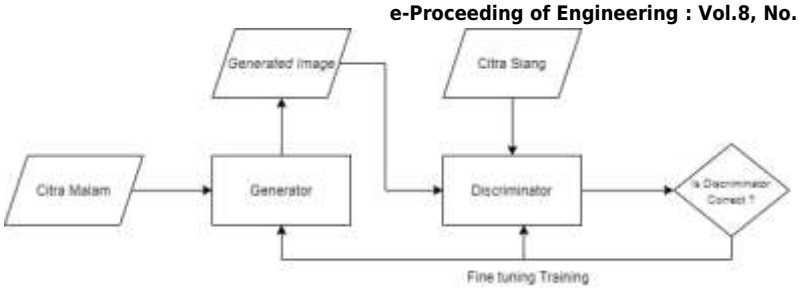
Gambar 1. Arsitektur DCGAN

3. Studi Terkait

Dalam bagian ini akan menjelaskan mengenai hal tentang desain dari sistem atau metode yang akan dibangun pada penelitian ini.

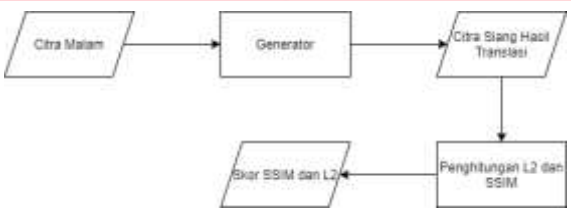
3.1 Desain Sistem

Sistem yang akan dibangun pada penelitian ini akan menghasilkan citra siang hasil translasi dari citra malam sesuai yang telah dijelaskan pada BAB I, gambaran umum dari proses pelatihan yang dilakukan pada penelitian ini dapat dilihat sebagai berikut.



Gambar 2. Skema Proses Pelatihan

Dari **Gambar 2** dapat dilihat gambaran dalam proses pelatihan diawali dengan citra malam menjadi *input* bagi *generator*, kemudian *generator* akan melakukan pembangkitan citra siang berdasarkan bentuk dari citra malam. Setelah itu citra hasil pembangkitan menjadi *input* bagi *discriminator*, dimana *discriminator* mendapatkan dua *input* yaitu citra hasil pembangkitan *generator* dan citra siang yang asli dan menghasilkan menghasilkan nilai label citra. Di sini *discriminator* akan melakukan penilaian melalui pembandingan dari kedua citra tersebut dan menghasilkan keputusan atau label dalam bentuk nilai *pixel* yang digunaan untuk melakukan *update* bobot pada *generator* dan *discriminator* melalui penghitungan dari total *loss*. Kemudian setelah dilakukan pelatihan, maka dilanjutkan dengan pengujian dengan proses seperti berikut.

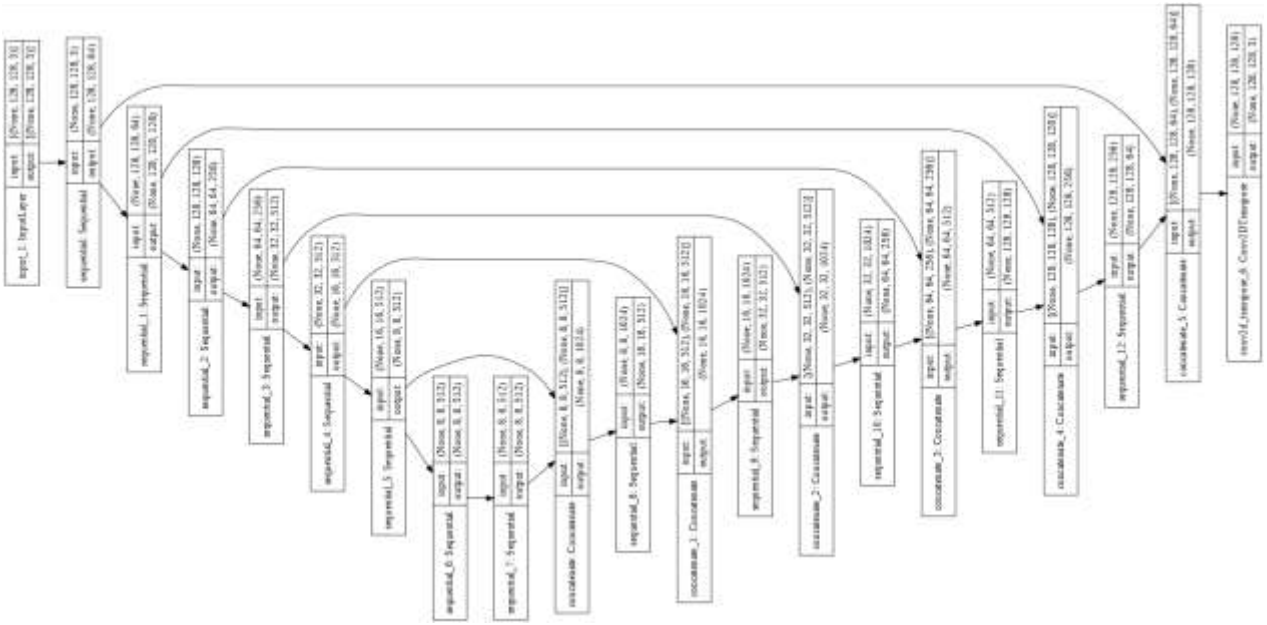


Gambar 3. Skema Proses Pengujian

Pada **Gambar 3**, menjelaskan bahwa proses prngujian dilakukan menggunakan *generator* model yang telah didapatkan dari proses pelatihan dimana citra malam sebagai *input* masuk ke *generator* dan dari *generator* menghasilkan citra siang hasil *generate* atau translasi. Setelah itu dari citra tersebut dilakukan penghitungan dan evaluasi nilai *pixel* menggunakan L2 dan SSIM.

3.2 Generator

Arsitektur *generator* pada penelitian menggunakan input citra 128x128 dengan 7 blok konvolusi dan 6 blok konvolusi pembalik sehingga menghasilkan citra hasil *generate* dengan ukuran 128x128. Di sini juga menerapkan metode *skip connection* yang bertujuan untuk mempertahankan atau menyimpan fitur secara saat dilakukan downsample yang kemudian fitur-fitur tersebut dikembalikan atau digabungkan kembali saat dilakukan *downsample*. Berikut adalah gambaran dari arsitektur DCGAN yang dibangun pada penelitian ini.

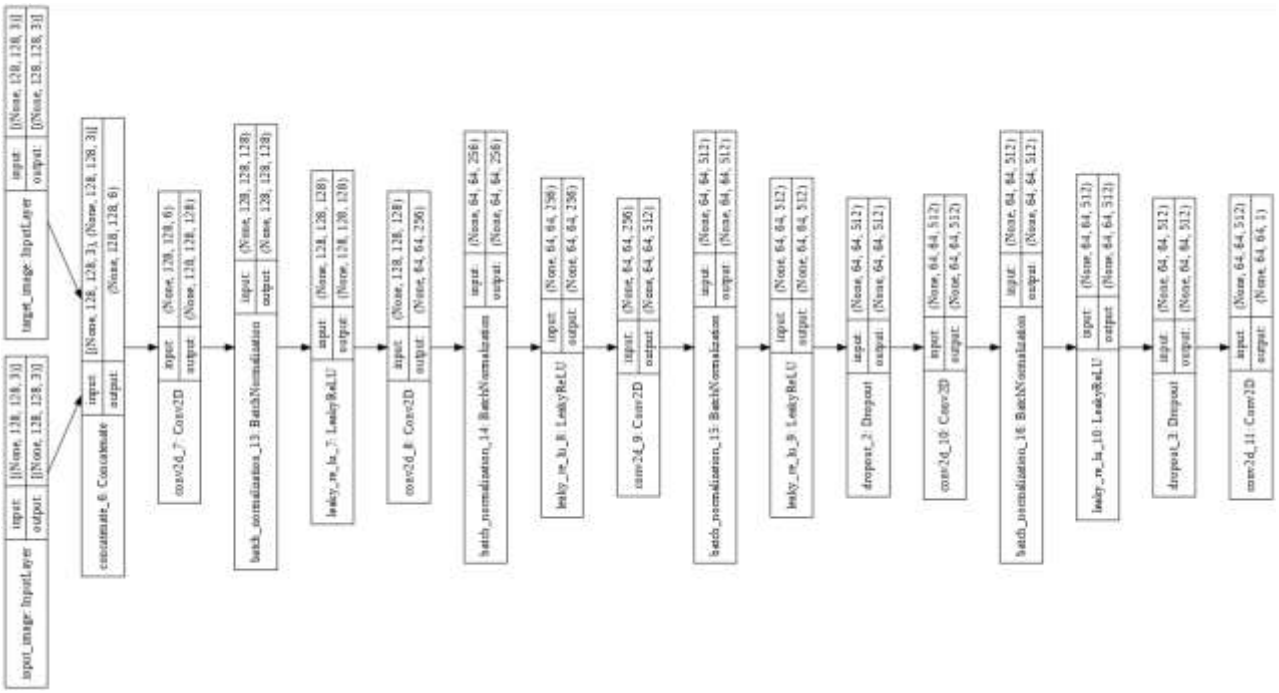


Gambar 4. Arsitektur Generator

Gambar 4 menjelaskan dimana blok konvolusi terdiri atas Conv2D 5x5xN (N merupakan jumlah filter yang digunakan), fungsi aktivasi LeakyReLU, dan Batch Normalization. Sedangkan pada konvolusi pembalik terdiri dari Conv2DTranspose 5x5xN dengan fungsi aktivasi ReLu dan Batch Normalization, akan tetapi pada blok terakhir tidak menggunakan Batch Normalization dan menggunakan fungsi aktivasi sigmoid.

3.2 Discriminator

Untuk arsitektur Discriminator bisa dilihat pada **Gambar 5** yang terdiri atas tumpukan blok konvolusi dengan filter yang semakin besar menggunakan Conv2D dengan ukuran kernel 5x5 dan N filter. Di setiap bloknya terdiri atas Conv2D, Batch Normalization, dan fungsi aktivasi LeakyReLu. Tetapi sama dengan Generator, pada blok terakhir tidak digunakan Batch Normalization dan fungsi aktivasi menggunakan sigmoid.



Gambar 5. Arsitektur Discriminator

3.3 Sample Data

Data yang akan digunakan dalam penelitian ini adalah data citra atau gambar yang identik akan tetapi berbeda ISO atau waktu pengambilan gambarnya, yaitu satu diambil saat siang hari dan satunya diambil pada malam hari. Data tersebut didapat dari hasil https://github.com/amitrajitbose/day-night-image-classifier/ dan *dataset* yang kami bangun sendiri dengan mengekstrak frame dari suatu video siang dan malam suatu tempat *outdoor*. **Gambar 6** merupakan contoh dataset citra identik yang akan digunakan dalam pelatihan dengan perbedaan pengambilan waktu yaitu siang dan malam.

Gambar 6a. Contoh *Dataset* Malam

Gambar 6b. Contoh *Dataset* Siang

3.4 Fungsi Loss

Pada tahap ini terdapat beberapa fungsi *loss* yang digunakan dalam peng-*update*-an bobot pada *discriminator* dan *generator*, yaitu :

a. L1 Loss

$$\text{L1 } loss = \frac{1}{N}\sum_{i}^{N} |g_i - x_i| \tag{2}$$

L1 *loss* atau dikenal sebagai (*mean absolute error)* digunakan untuk mengetahui perbedaan absolut pixel kedua citra yaitu citra asli (x_i) dan citra hasil pembangkitan *generator* (g_i), sehingga nantinya model bisa terlatih untuk menghasilkan pixel semirip mungkin. Hasil penjumlahan tersebut akan dicari nilai rata-ratanya dengan dibagi jumlah pixelnya (N).

b. L2 Loss

$$\text{L2 } loss = \frac{1}{N}\sum_{i}^{N} (g_i - x_i)^2 \tag{3}$$

L2 *loss* atau *(mean square error)* digunakan sama seperti L1 yaitu untuk memaksa model menghasilkan pixel semirip mungkin dengan aslinya dan mempertahankan fitur-fitur yang dimiliki citra berdasar perbedaan kuadrat antar pixel (*Euclidean distance*) [12].

c. PSNR Loss

$$MSE = \frac{1}{N}\sum_{i=1}^{N} X_i \; \frac{1}{m\,n}\sum_{j=0}^{m-1}\sum_{k=0}^{n-1}[G(z_i)_{(j,k)} - x_{i(j,k)}]^2$$
$$PSNR = 20 \times log_{10}(MAX_I) - 10 \times log_{10}(MSE) \tag{4}$$

PSNR *loss* digunakan untuk mengurangi noise pada citra yang hasil translasi, sehingga bisa menghasilkan gambar yang lebih realistic dengan menghitung kemungkinan nilai maximum (terbaik) dari suatu citra. Dimana m dan n adalah panjang dan tinggi citra, MAX_I adalah nilai maximum dari citra sedangkan $G(z_i)$ merupakan citra hasil pembangkitan g*enerator*.

d. Total Loss

Dari kedua fungsi *loss* diatas, selanjutnya kedua fungsi *loss* tersebut digabungkan dengan melakukan pembobotan pada masing-masing fungsi *loss*. Sehingga total fungsi *loss* dalam penelitian ini yaitu :

$$\textbf{Total } \boldsymbol{Loss} = (\lambda \times L1) + (\lambda \times L2) + PSNR \tag{5}$$

Dimana λ merepresentasikan bobot atau sebagai *hyperparameter* dengan nilai 100 yang berguna untuk memberi hasil bentuk yang lebih tajam dan halus. Penelitian ini juga mengikuti penelitian sebelumnya [13] yang menyatakan bahwa penelitian tersebut mendapatkan hasil visualisasi yang baik menggunakan nilai λ sebesar 100. Untuk pembuktiannya penelitian mencoba mengganti nilai λ menjadi 1 dan 50, ternyata mendapatkan hasil visualisasi yang kurang baik dibandingkan λ dengan nilai 100.

3.5 Tahap Pengujian

Setelah melakukan tahap pelatihan, maka ini merupakan tahap akhir dari penelitian ini. Pada tahap ini model hasil pelatihan diuji terhadap suatu citra malam yang ditranslasikan menjadi citra siang. Untuk mengetahui kemampuan dari model GAN, dilakukanlah pengukuran dengan menggunakan beberapa metode penghitungan yaitu SSIM dan L2 Norm.

- **SSIM**

SSIM (*Structural Similarity*) biasa digunakan untuk pengukuran kualitas citra yang didasarkan dari pengukuran degradasi kualitas dari suatu citra *input* yang juga bisa dijadikan sebagai nilai akurasi.

$$SSIM_{(x,y)} = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \tag{6}$$

Dari formula diatas menjelaskan bahwa μ_x merupakan nilai rata-rata gambar x, μ_y adalah nilai rata-rata gambar y, σ_x^2 merupakan nilai variansi gambar x, kemudian σ_y^2 adalah nilai variansi gambar y, dan yang terakhir σ_{xy} yaitu nilai kovarian dari gambar x dan y.

- **L2 Norm**

L2-norm digunakan sebagai pembanding untuk mengukur kesamaan fitur antara kedua citra dengan perhitungan kuadrat antar *pixel image* dimana semakin besar nilai L2 yang mendekati 0, maka citra tersebut semakin mendekati aslinya. Penjelasan rumus bisa dilihat pada **formula 3**.

4. Evaluasi

Dalam proses pelatihan dalam penelitian ini, terdapat beberapa parameter yang mempengaruhi hasil dari pelatihan model. Beberapa parameter yang menjadi fokus utama pada penelitian ini adalah ukuran citra, ukuran dari *downsampling* pada *generator*, dan *learning rate*. Sehingga menghasilkan tiga skenario tuning untuk mencari parameter-parameter yang terbaik sebagai berikut :

1. Skenario 1 : Membandingkan hasil antara citra ukuran 64x64 dengan 128x128.
2. Skenario 2 : Membandingkan hasil antara *Generator* dengan *downsample* mencapai 2 dan mencapai 8.
3. Skenario 3 : Membandingkan nilai *loss* antara *learning rate* $2e^{-4}$, $3e^{-4}$, dan $5e^{-4}$.

Untuk mengetahui kemampuan dari model yang diusulkan, maka dilakukan beberapa skenario eksperimen dalam pelatihan model, agar hasil dari setiap skenario dapat dibandingkan untuk menilai skenario mana yang memberikan hasil yang lebih baik. Skenario yang dimaksud adalah sebagait berikut :

1. Skenario Utama : Arsitektur DCGAN.
2. Skenario Pembanding : Pelatihan model menggunakan arsitektur PIX2PIX (Conditional GAN atau CGAN) [12].

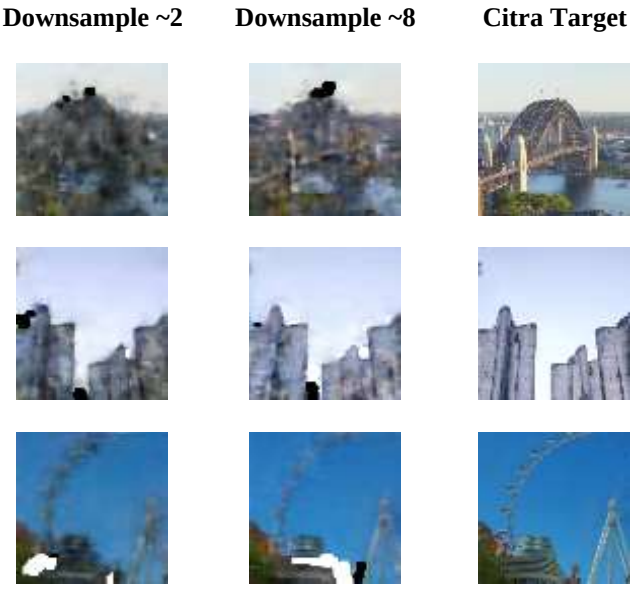
Setiap skenario dilatih dan diuji menggunakan *environment* yang sama, yaitu menggunakan Adam *optimizer* dengan ukuran per-*batch* yaitu 10. Setiap scenario dilatih menggunakan GPU dari NVIDIA GeForce RTX2060 sebanyak 500 epoch.

4.1. Hasil Pelatihan

Dalam penelitian ini, kami mencoba beberapa eksperimen untuk mendukung hipotesis kami dan mendapatkan parameter-parameter yang terbaik dalam pembangunan model DCGAN tersebut. Saat melakukan uji coba eksperimen, beberapa parameter yang dipertimbangkan adalah sejauh mana model melakukan *downsampling*, ukuran citra yang digunakan, dan mencari *learning rate* demi mendapatkan hasil yang terbaik. Pertama dilakukan uji coba perbandingan ukuran citra dan *downsampling* pada *datatrain* dan memberikan hasil seperti **Gambar 7**.

Gambar 7. Plot Hasil Uji Data Test

Dari visualisai **Gambar 7** penelitian ini membuktikan bahwa ukuran citra 128x128 lebih baik daripada ukuran citra 64x64 dikarenakan dengan semakin banyak jumlah *pixel*, maka semakin detail juga fitur yang akan di ekstrak dari suatu citra dikarenakan *generator* bekerja pada level *pixel*, kemudian melakukan penghitungan secara konvolusi dari masing-masing citra untuk mempelajari fitur yang terdapat pada citra tersebut. Tetapi dalam penelitian ini tidak melakukan penelitian pada ukuran 256x256 dikarenakan keterbatasan *resource* dari komputer yang digunakan untuk pelatihan, sehingga penelitian ini dilakukan dengan maksimal ukuran citra 128x128.



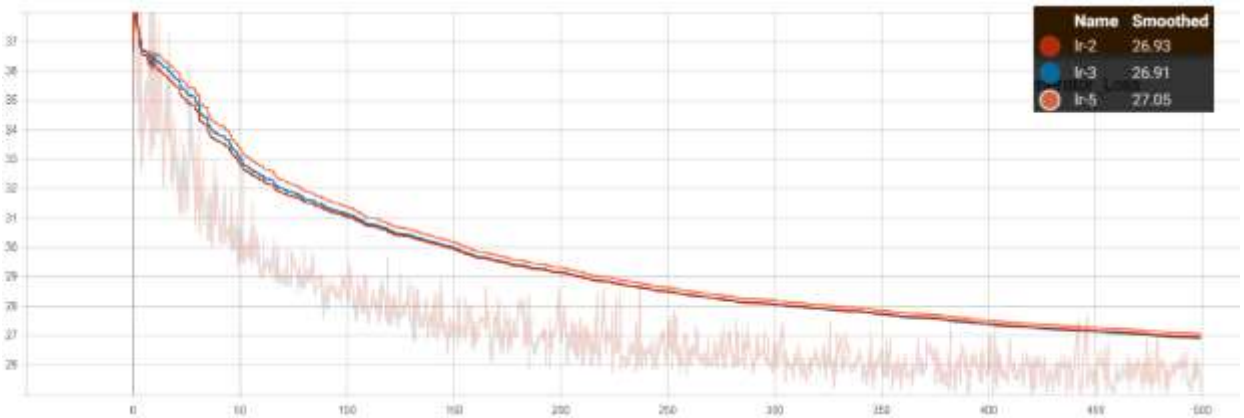
Gambar 8. Plot Hasil Uji Data Test

Kemudian dari **Gambar 8** menggunakan citra ukuran 128x128, dengan melakukan *downsampling* yang terlalu kecil yaitu mencapai ukuran *output* 2 membuat generator lebih lama untuk belajar dan menghasilkan kualitas yang lebih buruk daripada *generator* dengan *downsampling* hanya mencapai *output* 8. Hal ini terjadi dikarenakan dengan ukuran output yang semakin kecil, maka semakin sedikit pula fitur-fitur yang dapat diekstrak, tetapi penelitian ini tidak melakukan *downsampling* dengan ukuran lebih dari 8 dikarenakan penghitungan komputasi yang semakin tinggi sehingga memberatkan kinerja computer. Maka dari itu maksimal downsampling pada penelitian ini adalah 8. Setelah itu parameter penentu selanjutnya adalah ukuran citra dan *learning rate*, dalam pembangunan model *learning rate* sangat mempengaruhi seberapa cepat atau lambatnya model tersebut belajar. Hasil percobaan bisa dilihat pada tabel berikut.

Tabel 1. Perbandingan Akurasi Berdasarkan Ukuran Citra dan Learning Rate

Learning rate	Akurasi
Lr = $2e^{-4}$	36.27
Lr = $3e^{-4}$	**40.96**
Lr = $5e^{-4}$	40.55

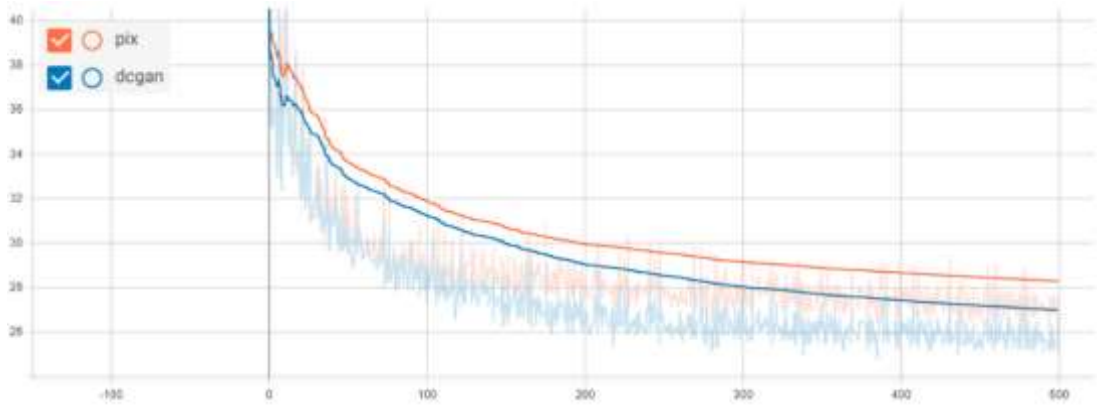
Dari **Tabel 1** didapatkan hasil bahwa dengan menggunakan *learning rate* sebesar **$3e^{-4}$** model dapat menghasilkan akurasi tertinggi yaitu mencapai **40.96 %**. Hal ini dikarenakan dengan *learning rate* yang terlalu kecil maka model akan lama untuk mempelajari suatu citra sedangkan apabila *learning rate* terlalu besar, model akan terlalu cepat untuk belajar, sehingga bisa dibilang menjadi kurang "teliti" dalam belajar.



Gambar 9. Plot Grafik Pelatihan *Tuning Learning Rate*

Pada **Gambar 9** merupakan hasil visualisasi penurunan nilai *loss* (sumbu y) terhadap banyak epoch (sumbu x), dimana saat melakukan *tuning learning rate* terlihat bahwa ketiga *learning rate* memberi hasil yang selalu menurun, akan tetapi walaupun ketiga grafik hampir sama, terdapat grafik yang menunjukan penurunan nilai

yang lebih baik, yaitu learning rate $3e^{-4}$ dimana nilai *loss* akhirnya pada epoch ke 500 mencapai **26.9**.

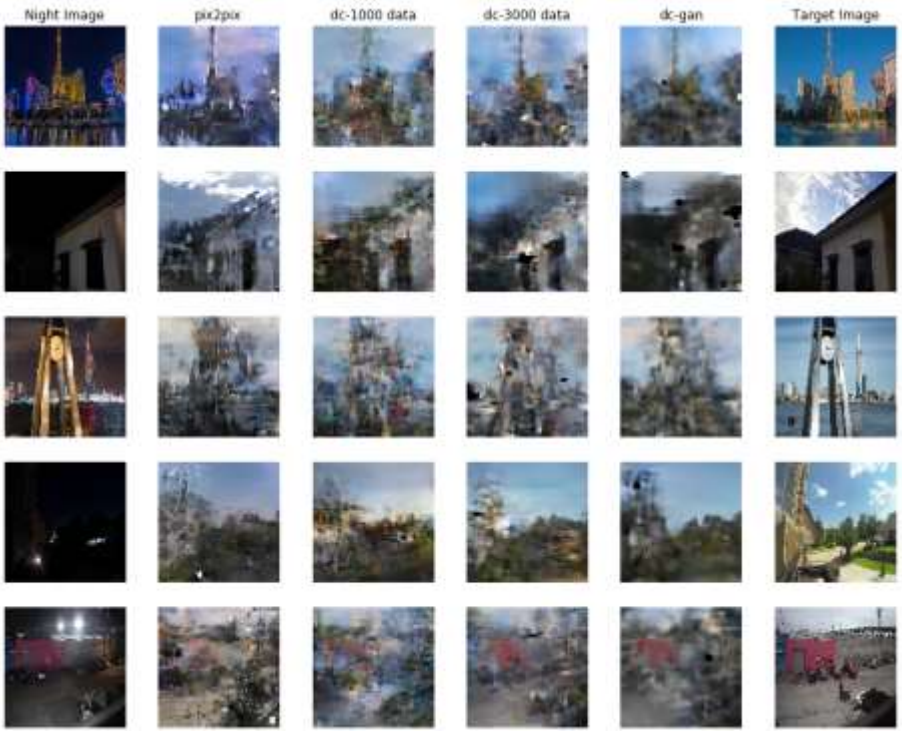


Gambar 10. Plot Grafik Pelatihan Model DCGAN dan Pix2Pix

Setelah melakukan pencarian parameter atau bisa disebut *hyperparameter tuning*, didapatkan hasil bahwa parameter terbaik dalam membangun model DCGAN ini adalah ukuran citra 128x128, *downsampling* model mencapai 8, dan menggunakan *learning rate* sebesar $3e^{-4}$. Dan setelah mendapatkan parameter-parameter yang dibutuhkan, proses pelatihan pun dilakukan dengan *environtment* dan parameter yang telah disebutkan. Penelitian dilakukan dengan melatih dua, yaitu DCGAN sebagai topik utama penelitian dan PIX2PIX [12] sebagai model pembanding kinerja dalam penelitian ini. Hasil grafik pelatihannya dapat dilihat pada **Gambar 10** yang merupakan visualisasi penurunan nilai *loss* (sumbu y) terhadap banyak epoch pelatihan (sumbu x). Dari grafik tersebut penurunan *loss* DCGAN lebih signifikan daripada PIX2PIX, dimana hasil akhirnya DCGAN mencapai 26.98, sedangkan PIX2PIX masih di angka 28,28.

4.2. Hasil Pengujian

Setelah proses pelatihan selesai, selanjutnya adalah melakukan pengujian pada *Generator* menggunakan data uji akan tetapi kami juga menggunakan skenario pembanding yaitu menggunakan model PIX2PIX. Kemudian didapat hasil pengujian dan perbandingan menggunakan arsitektur PIX2PIX dapat dilihat pada **Gambar 11**.



Gambar 11. Plot Hasil Uji Data Test

Tabel 2. Perbandingan Hasil SSIM dan L2-norm antara PIX2PIX dan DCGAN

Matriks Evaluasi	PIX2PIX	Real DCGAN
SSIM (akurasi)	34.38	**40.8**
L2 Norm	6.8	**6.29**

Dalam penelitian ini evaluasi dilakukan menggunakan SSIM dan L2 norm saja dikarenakan untuk metode yang lain seperti L1 dan PSNR digunakan untuk peng-*update* an bobot, sehingga pengevaluasian *error* cukup menggunakan L2, sedangkan akurasinya menggunakan SSIM dibanding PSNR dikarenakan penghitungan pada penelitian ini untuk menyatakan bahwa suatu citra tersebut dianggap citra siang adalah berdasarkan kualitas nilai *pixel* dan degradasi citra hasil pembangkitan terhadap citra target, sedangkan PSNR yang menghitung berdasarkan kuat sinyal antar *pixel* yang difokuskan untuk perbaikan *noise* citra, dianggap lebih baik digunakan untuk pembobotan saat pelatihan daripada evaluasi. Kemudian dari hasil visualisasi maupun evaluasi, model DCGAN memberikan hasil yang sedikit lebih baik daripada model PIX2PIX. Dari hasil pada **Tabel 2** didapatkan hasil bahwa DCGAN memiliki nilai SSIM atau akurasi yang tinggi yaitu mencapai 40% dengan *loss* sebesar ~6.3%. Sedangkan pada PIX2PIX penghitungan akurasi menggunakan SSIM hanya mendapatkan akurasi sekitar 34% dengan *loss* sebesar ~6.8%.

4.3. Analisa Hasil Pengujian

Dalam perlakuan masing-masing skenario, kami menggunakan metodologi yang sama dan masing-masing parameter metodologi memiliki alasan kenapa teknik tersebut digunakan. Dalam pelatihan generator, untuk melakukan update bobot, digunakan nilai total loss yang didapatkan dari **L1 loss + L2 loss + PSNR Loss**. L1 loss atau bisa disebut *(mean absolute error)* bekerja pada level pixel, sehingga nilai pixel dari citra yang dihasilkan dibuat semirip mungkin dengan pixel target. Untuk L2 loss digunakan sama seperti L1, akan tetapi menghitung *euclidean distance* fitur dari setiap output layer untuk mempertahankan fitur-fitur yang dimiliki citra, sehingga mengurangi kemungkinan rusaknya fitur setalah ditranslasi oleh *generator*. Dan kemudian PSNR loss digunakan untuk mengurangi *noise* pada citra hasil translasi, sehingga dapat menghasilkan citra yang lebih realistik. Sehingga dengan menggabungkan ketiga fungsi *loss* tersebut didapatkan hasil bahwa model DCGAN memiliki akurasi yang lebih baik daripada PIX2PIX berdasarkan nilai loss evaluasinya (L2-norm) dan nilai akurasi evaluasinya menggunakan SSIM.

Jika dilihat dari hasil data *test* pada **Gambar 11** memang belum menghasilkan hasil translasi yang maksimal. Hal tersebut terjadi dikarenakan beberapa hal yaitu :

1. Jumlah data yang terbatas dikarenakan *paired* data atau data yang identik dengan keadaan berbeda waktu yaitu malam dan siang masih susah untuk dicari, sehingga kami membangun data sendiri dengan mengekstrak frame dari video timelapse kemudian di-*augmentasi* dan hanya menghasilkan data sebanyak 6613 citra malam dan siang.

2. Kurangnya pelatihan untuk model dikarenakan spesifikasi dari komputer yang belum bisa melakukan training dalam jumlah besar, sehingga dalam perlakuan skenarionya hanya dilakukan sebanyak 500 epoch.

Berdasar dua poin diatas menunjukan kekurangan dalam segi sumber daya yang terbatas. Tantangan terbesar dalam translasi citra malam ke siang merupakan bagaimana cara melatih model untuk memunculkan sesuatu yang mungkin tidak terlihat di malam hari, sehingga dibutuhkan dataset yang sangat banyak. Hal ini terbukti dari hasil visualisasi atau plotingan citra, dari penggunaan data sebanyak 1000 data hingga menggunakan semua data membuktikan bahwa model bisa dan memiliki peluang untuk menghasilkan citra yang lebih baik apabila data latih yang digunakan semakin banyak dan bervariasi.

Kemudian banyak pelatihan juga berpengaruh untuk menghasilkan kualitas citra yang lebih baik [3]. Berdasarkan paper yang menjadi acuan, dua poin diatas sangatlah berpengaruh karena pada paper tersebut, dijelaskan bahwa pelatihan dilakukan menggunakan data gambar dengan jumlah lebih 20 juta citra dengan pelatihan selama 400-500 epoch [3] akan tetapi menggunakan metode *unsupervised* dimana kedua citra siang dan malam tidak perlu identic. Sedangkan pada penelitian yang lain menggunakan citra sebanyak lebih dari 18 ribu citra dengan banyak pelatihan mencapai 4000~5000 epoch [4], dimana datasetnya adalah dua citra identik dengan berbeda pengambilan ISO sehingga citra yang digunakan adalah citra yang gelap dan citra yang lebih terang bukan siang dan malam.

Akan tetapi baik secara visual dan penghitungan akurasi dari pada **Gambar 11** dan **Tabel 2**, pembangunan model menggunakan DCGAN menghasilkan hasil yang lebih baik dibandingkan model pembandingnya yaitu PIX2PIX dimana untuk model tersebut biasa digunakan untuk kasus *image translation* akan tetapi kurang baik pada kasus translasi siang ke malam.

5. Kesimpulan

Berdasarkan hasil penelitan yang telah dilakukan, didapatkan beberapa kesimpulan sebagai berikut

a. DCGAN dapat digunakan untuk membangun sistem yang mentranslasikan citra malam ke siang pada kasus *supervised* atau citra yang identik.
b. Berdasarkan hasil penelitian, model DCGAN dapat mentranslasikan citra ke malam ke siang lebih baik daripada model PIX2PIX dengan akurasi DCGAN mencapai 40% sedangkan PIX2PIX hanya sebesar 34%.

Saran dari peneliti untuk pengembangan dan penelitian berikutnya adalah untuk menambah jumlah data yang digunakan, melakukan *training* lebih lama, dan mencoba eksperimen untuk ukuran citra yang lebih besar dari 128 x 128 atau melakukan *training* dengan proses memecah citra ukuran berapapun secara bertahap menjadi persegi seperti susunan *puzzle* kemudian hasil akhirnya adalah dengan menyatukan Kembali potongan-potongan *puzzle* tersebut sehingga citra yang dihasilkan dapat memiliki kualitas yang lebih baik lagi.

Referensi

[1] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, vol. 1. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc, 2008.

[2] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2722–2730, doi: 10.1109/ICCV.2015.312.

[3] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. V. Gool, "Night-to-Day Image Translation for Retrieval-based Localization," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, May 2019, pp. 5958–5964, doi: 10.1109/ICRA.2019.8794387.

[4] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to See in the Dark," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, Jun. 2018, pp. 3291–3300, doi: 10.1109/CVPR.2018.00347.

[5] W. Cheng, W. Yang, M. Wang, G. Wang, and J. Chen, "Context Aggregation Network for Semantic Labeling in Aerial Images," *Remote Sensing*, vol. 11, no. 10, p. 1158, May 2019, doi: 10.3390/rs11101158.

[6] F. Magliani, T. Fontanini, and A. Prati, "A Dense-Depth Representation for VLAD descriptors in Content-Based Image Retrieval," *arXiv:1808.05022 [cs]*, Aug. 2018, Accessed: Jan. 31, 2021. [Online]. Available: http://arxiv.org/abs/1808.05022.

[7] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," pp. 1–9, 2014.

[8] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *arXiv:1511.08458 [cs]*, Dec. 2015, Accessed: Nov. 17, 2020. [Online]. Available: http://arxiv.org/abs/1511.08458.

[9] J. Yim, J. Ju, H. Jung, and J. Kim, "Image Classification Using Convolutional Neural Networks With Multi-stage Feature," in *Robot Intelligence Technology and Applications 3*, vol. 345, J.-H. Kim, W. Yang, J. Jo, P. Sincak, and H. Myung, Eds. Cham: Springer International Publishing, 2015, pp. 587–594.

[10] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How Does Batch Normalization Help Optimization?," *arXiv:1805.11604 [cs, stat]*, Apr. 2019, Accessed: Nov. 17, 2020. [Online]. Available: http://arxiv.org/abs/1805.11604.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv:1511.06434 [cs]*, Jan. 2016, Accessed: Nov. 17, 2020. [Online]. Available: http://arxiv.org/abs/1511.06434.

[12] S. D. Indradi, A. Arifianto, and K. N. Ramadhani, "Face Image Super-Resolution Using Inception Residual Network and GAN Framework," in *2019 7th International Conference on Information and Communication Technology (ICoICT)*, Kuala Lumpur, Malaysia, Jul. 2019, pp. 1–6, doi: 10.1109/ICoICT.2019.8835253.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *arXiv:1611.07004 [cs]*, Nov. 2018, Accessed: Nov. 17, 2020. [Online]. Available: http://arxiv.org/abs/1611.07004.

Lampiran