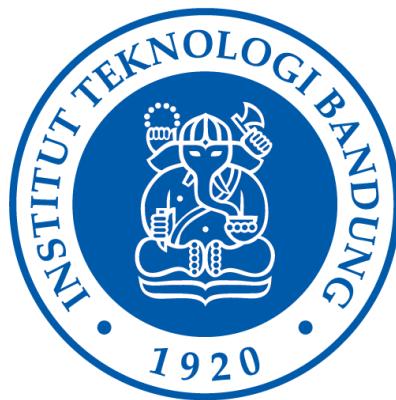


UJIAN AKHIR SEMESTER

IF5280 *Deep Learning*



Nama : Nabila Hannania

NIM : 23523018

Program Studi Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2024

Abstrak

Dalam kehidupan sehari-hari, gambar sering mengalami kerusakan fisik atau kehilangan informasi yang disengaja, baik karena faktor lingkungan maupun proses transmisi data. Pemulihan dan penghilangan informasi yang tidak diinginkan dari gambar penting untuk melestarikan warisan budaya, meningkatkan estetika, dan memastikan komunikasi visual yang efektif. Salah satu solusi untuk mengatasi masalah ini adalah dengan teknik image inpainting, yang dapat mengisi area yang hilang atau rusak pada gambar secara mulus menggunakan model generatif.

Dalam penelitian ini, tiga pendekatan dalam pengembangan model image inpainting dieksplorasi: Convolutional based Autoencoder, Partial Convolutional based Autoencoder, dan GAN. Setiap pendekatan memiliki kelebihan dan kelemahan yang berbeda. Uji kinerja dilakukan menggunakan metrik pengukuran kuantitatif seperti Mean Absolute Error, Mean Square Error, dan Dice Coefficient.

Hasil penelitian menunjukkan bahwa model Convolutional based Autoencoder memberikan kinerja terbaik untuk dataset CIFAR-10 karena arsitektur yang sederhana, yang cocok untuk dataset yang relatif kecil. Sementara itu, Model Partial Convolutional based Autoencoder efektif mengatasi mask dengan bentuk tidak teratur. Penggunaan ukuran batch yang lebih kecil pada model GAN juga terbukti meningkatkan stabilitas pelatihan, terutama pada dataset yang relatif kecil.

Latar Belakang

Dalam kehidupan sehari-hari, sering terjadi kerusakan atau kehilangan informasi pada gambar karena berbagai faktor. Salah satu jenis kerusakannya adalah kerusakan fisik, yang bisa terjadi pada gambar atau foto yang disimpan dalam media fisik seperti kertas atau film. Kerusakan ini bisa disebabkan oleh berbagai hal, mulai dari usia gambar itu sendiri, paparan air, hingga dampak lingkungan seperti kelembaban atau suhu ekstrem. Selain kerusakan fisik, informasi pada gambar sering dihilangkan secara sengaja karena terdapat objek yang tidak diinginkan yang perlu dihilangkan tanpa meninggalkan jejak yang mencolok. Contohnya, grafiti pada bangunan bersejarah atau orang yang tidak diinginkan dalam sebuah foto keluarga. Proses menghilangkan objek ini tidak hanya memerlukan keahlian teknis yang tinggi tetapi juga harus dilakukan dengan hati-hati agar tidak merusak integritas visual dari gambar tersebut. Kemudian, dalam konteks transmisi data gambar melalui jaringan, seringkali sebagian data bisa hilang atau rusak karena gangguan sinyal atau kegagalan koneksi. Hal ini bisa terjadi pada berbagai jenis komunikasi, mulai dari transfer gambar melalui internet hingga transmisi citra medis yang kritis.

Penting untuk mengatasi masalah kehilangan atau kerusakan pada bagian gambar ini. Pertama-tama, pemulihan gambar dan foto yang rusak dapat menjadi langkah penting dalam melestarikan warisan budaya dan sejarah. Misalnya, restorasi gambar-gambar kuno atau foto-foto keluarga yang rusak secara fisik dapat membantu menjaga integritas sejarah suatu keluarga atau komunitas. Kedua, menghilangkan objek yang tidak diinginkan dari gambar dapat meningkatkan kegunaan estetika gambar tersebut. Ini sering kali diperlukan dalam konteks profesional seperti fotografi, desain grafis, atau penggunaan gambar dalam media pemasaran. Penghilangan objek yang tidak diinginkan dapat membantu menciptakan gambar yang lebih bersih, estetis, dan fokus pada pesan yang ingin disampaikan. Terakhir, memperbaiki gambar yang korup atau rusak dapat memastikan komunikasi visual yang jelas dan efektif. Hal ini sangat penting dalam berbagai bidang seperti medis, di mana citra medis yang rusak bisa mengganggu proses diagnosis, atau dalam pengawasan, di mana citra satelit yang tidak sempurna bisa mengganggu pemantauan keamanan atau lingkungan.

Salah satu solusi yang dapat digunakan untuk mengatasi permasalahan pada gambar ini adalah dengan menggunakan teknik image inpainting. Image inpainting adalah teknik dalam pengolahan

citra digital yang digunakan untuk mengisi area yang hilang atau rusak pada gambar dengan cara yang mulus dan tidak terlihat oleh mata manusia. Terdapat beberapa pendekatan utama yang digunakan dalam image inpainting, salah satunya adalah pendekatan dengan model generatif. Pendekatan ini menggunakan model berbasis kecerdasan buatan, seperti Generative Adversarial Networks (GAN) dan AutoEncoder, untuk menghasilkan bagian gambar yang hilang dengan belajar dari dataset gambar yang besar.

Pada tugas ini, terdapat tiga pendekatan dalam pengembangan model image inpainting, yakni menggunakan Convolutional based Autoencoder, Partial Convolutional based Autoencoder, dan GAN. Setiap pendekatan memiliki keunggulan dan kelemahan yang berbeda. Oleh karena itu, untuk menguji kinerja model yang dikembangkan, beberapa metrik pengukuran kuantitatif digunakan, seperti Mean Absolute Error (MAE), Mean Square Error (MSE), dan Dice Coefficient.

Studi Literatur

Penelitian mengenai *image inpainting* sudah banyak dilakukan dan sudah dibahas cukup lama. Berikut ini beberapa penelitian yang membahas mengenai *image inpainting*.

1. Image Inpainting for Irregular Holes Using Partial Convolutions

Link Paper: <https://arxiv.org/pdf/1804.07723>

Paper ini memperkenalkan teknik partial convolutions yang mengutamakan piksel-piksel yang valid. Metode ini berhasil mengungguli pendekatan-pendekatan sebelumnya dalam menangani irregular masks tanpa perlu melakukan post-processing. Pendekatan ini menggunakan Partial Convolutional Layer yang hanya memperhitungkan piksel-piksel valid dalam setiap operasi konvolusi dan secara otomatis memperbarui mask setelah setiap proses konvolusi untuk secara bertahap mengisi lubang pada gambar. Arsitektur model yang diadopsi mirip dengan UNet dengan menggantikan semua Convolutional Layer dengan Partial Convolutional Layer. Arsitektur ini juga dilengkapi dengan Skip Links yang menggabungkan feature maps dan masks untuk melakukan inpainting dengan lebih efektif. Selain itu, penggunaan Partial Convolutional sebagai Padding memastikan bahwa konten yang dipulihkan di batas gambar tidak dipengaruhi oleh nilai-nilai tidak valid di luar gambar. Model yang dikembangkan juga memanfaatkan berbagai fungsi loss seperti Total Variation Loss, Perceptual Loss, Style Loss, Pixel Losses, dan Total loss (kombinasi dari semua loss).

Data yang digunakan dalam penelitian ini berasal dari beberapa sumber, termasuk ImageNet, Places2, dan CelebA-HQ datasets. Hasil dari model yang diuji dibandingkan dengan metode PatchMatch (PM), metode yang diusulkan oleh Iizuka et al. (GL), dan metode oleh Yu et al. (GntIpt). Hasil evaluasi menunjukkan peningkatan baik secara kualitatif maupun kuantitatif (L1, PSNR, dan SSIM) dibandingkan dengan metode-metode sebelumnya.

Kelebihan pendekatan yang digunakan pada paper ini:

- a) Partial convolutions hanya menggunakan piksel-piksel valid untuk operasi konvolusi, yang secara signifikan mengurangi artefak seperti perbedaan warna dan kekaburan.
- b) Tidak membutuhkan Post-Processing

- c) Dapat menangani dan merekonstruksi gambar dengan lubang berbentuk apa pun
- d) Menggunakan skip connections yang secara efektif menjaga prediksi yang detail, meningkatkan kualitas inpainting.

Kekurangan pendekatan yang digunakan pada paper ini:

- a) Pelatihan model membutuhkan sumber daya komputasi yang besar, termasuk GPU yang kuat dan waktu pelatihan yang panjang.
- b) Kinerja model sangat bergantung pada kualitas dan keragaman dataset pelatihan.
- c) Kesulitan menghadapi gambar yang memiliki struktur yang jarang, seperti gambar dengan pola berulang atau garis tipis, yang dapat mengakibatkan hasil yang kurang memuaskan.
- d) Mengalami kesulitan dalam menangani lubang yang sangat besar.

2. Image Inpainting using Partial Convolution

Link Paper: <https://arxiv.org/pdf/2108.08791>

Penelitian pada paper ini terinspirasi dari paper "*Image Inpainting for Irregular Holes Using Partial Convolutions*" untuk mengembangkan model *image inpainting* yang mampu memulihkan gambar dengan lubang-lubang berbentuk sembarang. Dalam penelitian paper ini, Partial Convolution Layers digunakan, di mana operasi konvolusi hanya dilakukan pada piksel yang tersedia (unmasked), diikuti dengan langkah pembaruan mask. Arsitektur model yang diterapkan adalah modifikasi dari UNet Architecture dengan total 7 lapisan (tujuh encoder dan tujuh decoder) untuk merekonstruksi gambar. Fungsi aktivasi yang digunakan pada encoder dan decoder ada berbagai macam, yaitu ReLU untuk semua lapisan encoding, Leaky ReLU untuk lapisan decoding kecuali lapisan pertama, dan tanpa aktivasi untuk lapisan decoding pertama. Selain itu, model yang dikembangkan menggunakan berbagai loss function (Total Variation Loss, Perceptual Loss, Style Loss, dan Pixel Losses) untuk meningkatkan proses restorasi gambar.

Data yang digunakan dalam penelitian ini adalah dataset CelebA-HQ/256, yang terdiri dari 30.000 gambar wajah manusia dengan resolusi 256x256 piksel. Dataset ini dibagi menjadi dataset pelatihan, validasi, dan pengujian dalam proporsi 70%, 15%, dan 15%. Model pada paper ini dilatih selama 34.500 iterasi dengan ukuran batch 3.

Kelebihan pendekatan yang digunakan pada paper ini:

- a) Partial convolution layers fokus pada unmasked pixels, memungkinkan model memahami dan merekonstruksi semantik gambar secara akurat.
- b) Dapat menangani dan merekonstruksi gambar dengan lubang berbentuk apa pun, sehingga sangat serbaguna untuk berbagai tugas image inpainting.
- c) Kombinasi partial convolution layers and advanced loss functions (Total Variation Loss, Perceptual Loss, Style Loss, Pixel Losses) memastikan bahwa gambar yang dipulihkan memiliki koherensi visual dan tetap berkualitas tinggi.
- d) Menggunakan beragam loss functions yang membantu dalam menjaga fitur-fitur struktural (sisi, sudut) dan fitur-fitur gaya (pola warna), yang menghasilkan inpainting berkualitas tinggi.

Kekurangan pendekatan yang digunakan pada paper ini:

- a) Pelatihan model membutuhkan sumber daya komputasi yang besar, termasuk GPU yang kuat dan waktu pelatihan yang panjang (sekitar 20 jam untuk 34.500 iterasi).
- b) Kinerja model sangat bergantung pada kualitas dan keragaman dataset pelatihan.

3. Image Inpainting with Cascaded Modulation GAN and Object-Aware Training

Link Paper: <https://arxiv.org/pdf/2203.11947v3>

Paper ini bertujuan untuk mengatasi keterbatasan teknik image inpainting saat ini, terutama dalam menangani celah besar dalam gambar yang rumit. Dengan memperkenalkan metode Cascaded Modulation GAN (CM-GAN), yang bertujuan untuk melampaui kinerja pendekatan state-of-the-art saat ini seperti ProFill, LaMa, dan CoModGAN, dengan menghasilkan gambar yang lebih realistik dan meyakinkan. CM-GAN terdiri dari encoder yang dilengkapi dengan Fourier convolution blocks dan dual-stream decoder yang dilengkapi dengan cascaded global-spatial modulation blocks pada skala yang berbeda. Model ini menggunakan global modulation untuk mensintesis struktur kasar dan spatial modulation untuk membuat penyesuaian yang lebih tepat agar menghasilkan gambar yang lebih halus. CM-GAN menunjukkan kinerja yang jauh lebih unggul dibandingkan dengan metode-metode yang ada saat ini, baik secara kuantitatif maupun kualitatif.

Kelebihan pendekatan yang digunakan pada paper ini:

- a) Kombinasi global modulation dan spatial modulation menghasilkan hasil inpainting yang lebih berkualitas.
- b) Penggabungan Fourier convolution blocks pada encoder memungkinkan model untuk menangkap dependensi jarak jauh sehingga meningkatkan kinerja inpainting secara keseluruhan.
- c) Masked R1 Regularization yang digunakan menstabilkan pelatihan adversarial, meningkatkan proses pelatihan dan menghasilkan kualitas inpainting yang lebih baik.

Kekurangan pendekatan yang digunakan pada paper ini:

- a) Penggunaan Fourier convolution dan dual-stream decoders dapat meningkatkan kompleksitas model.
- b) Skema object-aware mask generation yang digunakan membutuhkan anotasi segmentasi tingkat instansi, yang mungkin tidak tersedia atau sulit untuk dihasilkan.

4. WavePaint: Resource-Efficient Token-Mixer for Self-Supervised Inpainting

Link Paper: <https://arxiv.org/pdf/2307.00407v1>

Paper ini mengusulkan model baru yang lebih efisien secara komputasi untuk melakukan *image inpainting*, yaitu WavePaint. Model ini berbeda dari model berbasis transformer karena menggunakan WaveMix-based fully convolutional, dimana model ini lebih efisien dalam penggunaan sumber daya. Berbeda dengan metode berbasis GAN, WavePaint tidak memerlukan pelatihan adversarial, sehingga dapat mengurangi kompleksitas dan konsumsi sumber daya. Melalui eksperimen yang dilakukan pada paper ini, WavePaint mampu melampaui model-model besar seperti LaMa dan CoModGAN dalam kualitas rekonstruksi meskipun menggunakan jumlah parameter yang lebih sedikit dan daya komputasi yang lebih rendah.

Kelebihan pendekatan yang digunakan pada paper ini:

- a) WavePaint menggunakan parameter yang lebih sedikit sehingga biaya komputasinya lebih rendah dan waktu pelatihan serta inferensi yang lebih cepat.
- b) Berbeda dengan model berbasis GAN, WavePaint tidak memerlukan discriminator, sehingga dapat mengurangi kompleksitas proses pelatihan.
- c) WavePaint dapat dilatih dengan lebih cepat dan dengan sumber daya komputasi yang lebih sedikit dibandingkan dengan GAN dan model difusi.

Kekurangan pendekatan yang digunakan pada paper ini:

- a) Implementasi saat ini hanya berfokus pada large-mask inpainting dengan bentuk masks-nya diketahui. Kinerjanya mungkin tidak akan terlalu baik, jika digunakan pada blind mask inpainting di mana bentuk masks-nya tidak diketahui..
- b) Membutuhkan GPU memory yang besar (misalnya 80 GB A100 GPU)

5. Aggregated Contextual Transformations for High-Resolution Image Inpainting

Link Paper: <https://arxiv.org/pdf/2104.01431>

Paper ini membahas high-resolution image inpainting dengan fokus mengatasi masalah dalam content reasoning dan texture synthesis untuk masks yang ukurannya besar. Model yang diusulkan, Aggregated COntextual-Transformation GAN (AOT-GAN), bertujuan untuk melampaui pendekatan berbasis GAN yang ada dengan meningkatkan content reasoning dan texture synthesis. Generator pada model ini menggunakan Aggregated Contextual Transformation (AOT) blocks yang membagi kernel konvolusi, menerapkan laju dilatasi yang berbeda, dan menggabungkan transformasi untuk menangkap konteks yang jauh dan pola yang kaya. Sedangkan diskriminator-nya dilatih dengan mask-prediction task yang memaksanya membedakan patch yang nyata dari yang disintesis, sehingga dapat meningkatkan texture synthesis. AOT-GAN dilatih dengan mengkombinasikan reconstruction loss, adversarial loss, perceptual loss, and style loss. Berdasarkan eksperimen yang dilakukan, AOT-GAN berhasil mencapai peningkatan yang signifikan dibandingkan dengan metode state-of-the-art dalam hal FID dan metrik lainnya, terutama untuk gambar beresolusi tinggi.

Kelebihan pendekatan yang digunakan pada paper ini:

- a) AOT blocks menangkap distant contexts dan rich patterns, sehingga menghasilkan struktur dan sintesis tekstur yang lebih koheren.
- b) Mask-prediction task pada discriminator memastikan tekstur yang lebih jelas dan realistik di daerah-daerah hasil inpainting.

Kekurangan pendekatan yang digunakan pada paper ini:

- a) Kesulitan dalam memilih appropriate masks untuk gambar-gambar kompleks atau tugas-tugas interaktif dapat membatasi penerapan metode ini.

Dataset

1. Deskripsi Dataset

Pada tugas ini, dataset yang digunakan adalah CIFAR-10. CIFAR-10 merupakan dataset yang sering digunakan di bidang machine learning dan computer vision. Dataset ini dibuat oleh Canadian Institute for Advanced Research (CIFAR) dan secara luas digunakan untuk mengevaluasi serta membandingkan kinerja berbagai algoritma machine learning, terutama dalam tugas klasifikasi gambar. Dataset CIFAR-10 terdiri dari 60.000 gambar berwarna dengan resolusi 32x32 piksel. Dataset ini terbagi dalam 10 kelas, masing-masing kelas memiliki 6.000 gambar. Dari keseluruhan data tersebut, 50.000 gambar digunakan sebagai data pelatihan dan 10.000 gambar digunakan sebagai data pengujian. Pada data pengujian terdapat tepat 1.000 gambar yang dipilih secara acak dari setiap kelas. Dataset ini dapat diakses melalui tautan berikut: [CIFAR-10 Dataset](#) atau melalui library Keras dan PyTorch seperti yang ditunjukkan pada kode tugas.

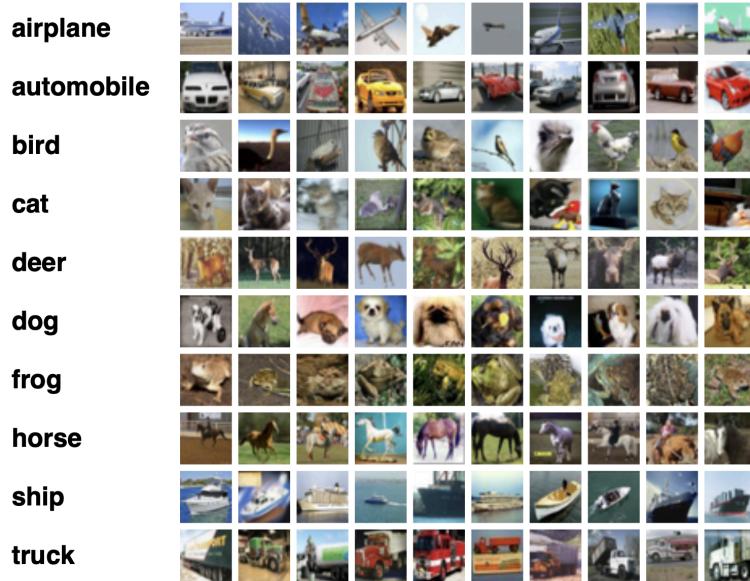
```
# Load the CIFAR-10 dataset  
(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()
```

Cara Mengakses Data CIFAR melalui Keras

```
# Download & read data  
train_data = torchvision.datasets.CIFAR10(root='./data', train=True, download=True,  
transform=transform)  
test = torchvision.datasets.CIFAR10(root='./data', train=False, download=True,  
transform=transform)
```

Cara Mengakses Data CIFAR dari torchvision (PyTorch)

Adapun alasan pemilihan dataset ini adalah karena ukurannya yang kecil (baik dari segi jumlah maupun resolusi), sehingga memudahkan proses pelatihan, membuat pelatihan menjadi lebih cepat, dan sumber daya yang dibutuhkan tidak terlalu banyak.



Contoh Data CIFAR-10

2. Pembagian Data

Data CIFAR-10 dibagi menjadi tiga set data, yaitu data pelatihan (train), validasi (validation), dan pengujian (test). Data pengujian diperoleh langsung dari dataset CIFAR-10 yang sudah disediakan. Sedangkan data validasi diambil dari sebagian data pelatihan, yaitu sebesar 20% dari data pelatihan digunakan sebagai data validasi. Pembagian data ini dilakukan dengan membagi secara langsung tanpa melakukan pemilihan data secara acak.

```

validation_ratio = 0.2
num_train = x_train.shape[0]
num_val = int(num_train * validation_ratio)
num_train = num_train - num_val

# Split the training set into training and validation sets
# Manually split the data
x_val = x_train[:num_val]
y_val = y_train[:num_val]
x_train = x_train[num_val:]

```

```
y_train = y_train[num_val:]
```

Kode Pembagian Data menjadi 2 Set

Detail hasil pembagian data yang telah dilakukan dan detail ukuran data dapat dilihat pada gambar di bawah ini.

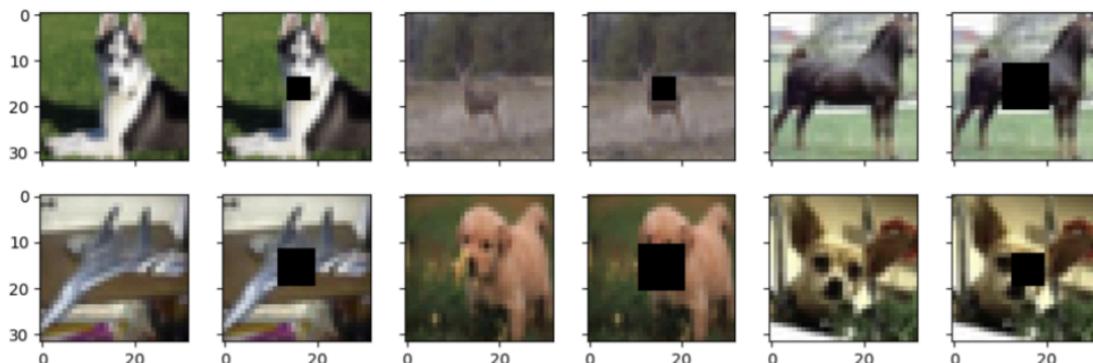
```
Training data shape: (40000, 32, 32, 3), Training sample shape: 40000  
Validation data shape: (10000, 32, 32, 3), Validation sample shape: 10000  
Test data shape: (10000, 32, 32, 3), Test sample shape: 10000
```

Hasil Pembagian Data CIFAR-10

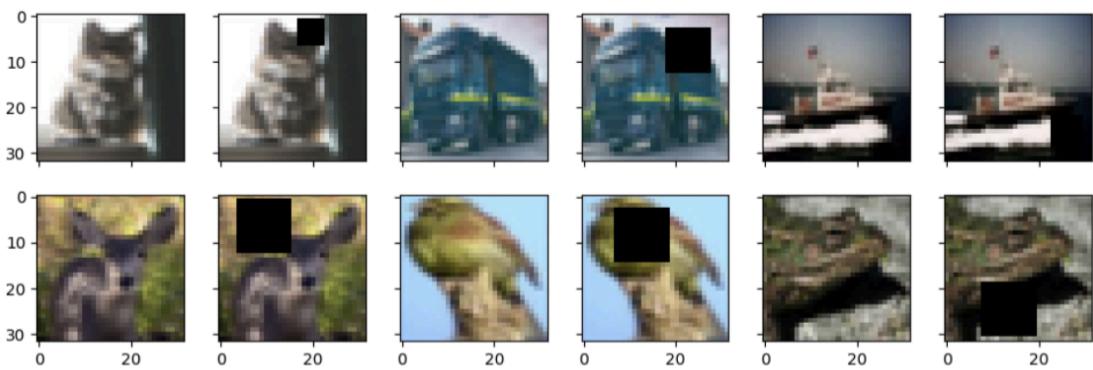
3. Preprocessing Data

Selain melakukan pembagian data, preprocessing yang dilakukan adalah menerapkan masking pada dataset. Proses masking ini bertujuan agar model image inpainting yang dibangun dapat "mengisi" bagian yang kosong tersebut sehingga gambar menjadi utuh. Ukuran masking untuk setiap gambar berbeda-beda karena dilakukan secara acak. Dalam tugas ini, terdapat tiga jenis masking yang digunakan:

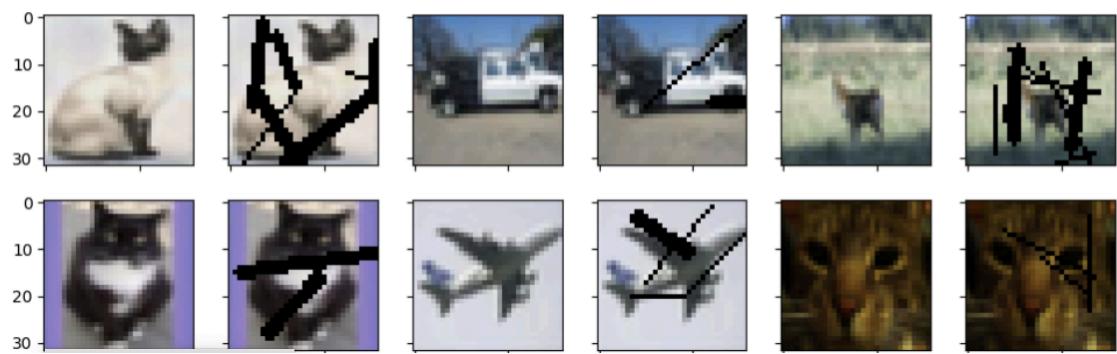
- Regular hole (box) mask pada tengah gambar.



- Regular hole (box) mask pada lokasi acak pada gambar.



- Irregular hole mask pada lokasi acak pada gambar.



Metode

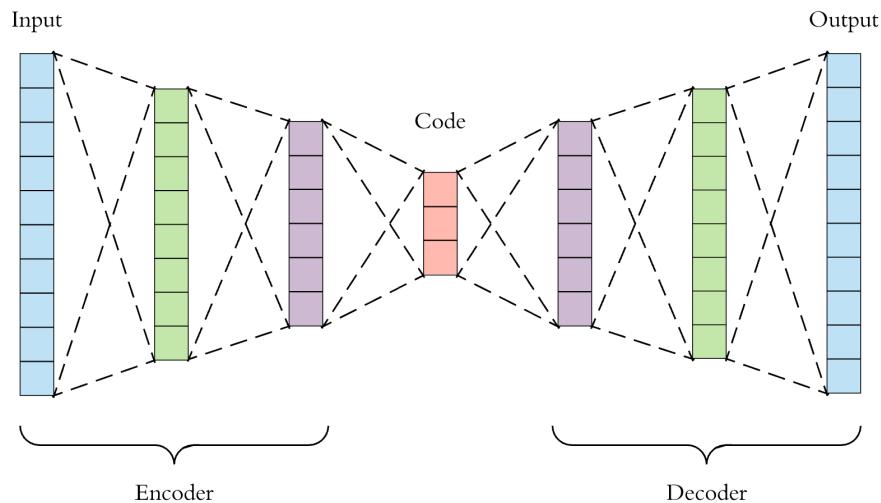
1. Metode Pengembangan Model

Pada tugas ini, terdapat 3 metode pengembangan model yang digunakan, yaitu dengan menggunakan Convolutional based Autoencoder, Partial Convolutional based Autoencoder, dan GAN.

a) Convolutional based Autoencoder

Referensi:

https://wandb.ai/wandb_fc/articles/reports/Introduction-to-image-inpainting-with-deep-learning--Vmlldzo1NDI3MjA5

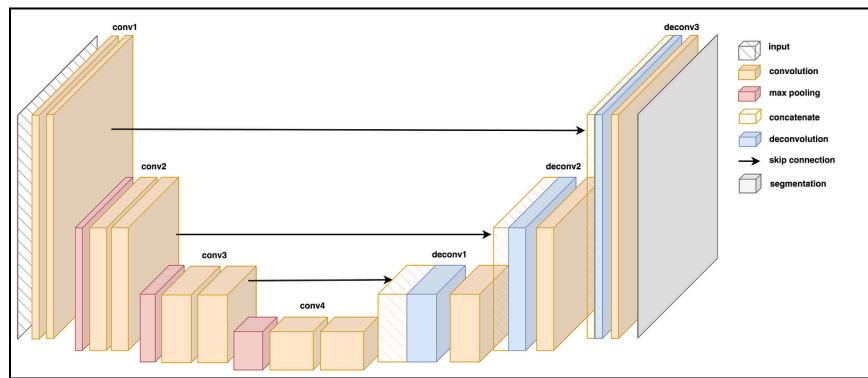


Arsitektur Autoencoder

Metode ini dipilih karena CNN memanfaatkan filter konvolusional yang efektif dalam menangkap pola dan fitur lokal dalam gambar. Kemampuan ini menjadi sangat berharga dalam konteks image inpainting, di mana pemahaman tentang konteks lokal di sekitar area yang hilang menjadi kunci untuk menghasilkan konten yang koheren secara visual. Selain itu, lapisan konvolusi menggunakan shared weight (filter), sehingga membuatnya efisien secara komputasi dan mengurangi risiko overfitting dibandingkan dengan lapisan fully connected. Keefisienan ini sangat penting ketika berurusan dengan data gambar yang memiliki dimensi yang tinggi.

Alasan lainnya adalah karena arsitektur autoencoder, yang terdiri dari encoder yang mampu mereduksi dimensi gambar input menjadi representasi laten yang lebih rendah dan decoder yang mampu merekonstruksi gambar dari representasi tersebut. Struktur ini sangat cocok untuk image inpainting karena encoder dapat menangkap konteks global dari gambar, sedangkan decoder dapat menghasilkan bagian yang hilang dengan memanfaatkan informasi dari representasi laten. Dengan demikian, kombinasi antara CNN dan arsitektur autoencoder menjadikan metode ini sangat sesuai untuk tugas inpainting.

Model Convolutional based Autoencoder adalah model yang kompleks, terdiri dari generator dan discriminator yang memanfaatkan lapisan convolution dan membentuk arsitektur UNet. Arsitektur UNet sendiri terdiri dari beberapa lapisan yang diatur sedemikian rupa sehingga membentuk struktur seperti huruf U.



Arsitektur UNet

Encoder (Kiri):

- Input : gambar dengan bagian yang hilang.
- Jalur kontraksi: terdiri dari beberapa ConvBlocks, dimana tiap ConvBlock berisi:
 - Dua lapisan konvolusional untuk ekstraksi fitur.
 - Lapisan pooling opsional (misalnya, MaxPooling2D) untuk melakukan downsampling pada peta fitur.
 - Lapisan aktivasi (mis., ReLU) untuk memperkenalkan non-linearitas.
 - Padding (disarankan 'same') untuk mempertahankan dimensi spasial.

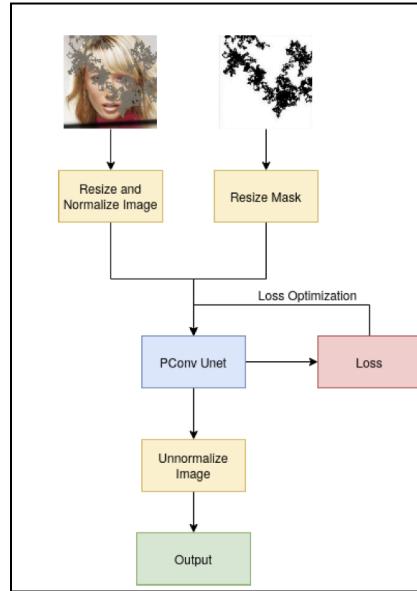
Decoder (Kanan):

- Jalur ekspansi: Jalur ini menggunakan UpConvBlocks untuk melakukan upsampling feature map dan memulihkan informasi spasial, dimana tiap UpConvBlocks biasanya berisi:
 - Dua lapisan konvolusional untuk penyempurnaan fitur.
 - Transposed convolution layer (Conv2DTranspose) untuk upsampling.
 - Lapisan Concatenation untuk menggabungkan upsampled features dengan fitur-fitur yang sesuai dari jalur kontrak (skip connections).
 - Lapisan aktivasi (mis., ReLU) untuk memperkenalkan non-linearitas.
 - Padding (disarankan 'same') untuk mempertahankan dimensi spasial.

b) Partial Convolutional based Autoencoder

Referensi:

https://wandb.ai/wandb_fc/articles/reports/Introduction-to-image-inpainting-with-deep-learning--Vmlldzo1NDI3MjA5
<https://arxiv.org/pdf/2108.08791>



Pipeline Pendekatan dengan Partial Convolutional based Autoencoder

Metode ini dipilih karena Partial Convolution memperbarui mask secara dinamis selama proses konvolusi. Partial Convolution hanya mempertimbangkan piksel yang valid (tidak hilang) untuk

konvolusi, sehingga dapat menangani dan merekonstruksi gambar dengan lubang (masks) berbentuk apa pun.

Partial Convolution Layers

Partial convolution, seperti namanya, mirip dengan convolution layer, hanya saja ia hanya melakukan operasi konvolusi pada piksel yang piksel masukannya saat ini tidak disamarkan atau "tersedia". Partial Convolutional Layers menggunakan binary mask untuk menentukan piksel mana pada gambar masukan yang valid dan harus dipertimbangkan selama konvolusi.

$$x' = \begin{cases} W^T(X \circledast M) \frac{\text{sum}(\mathbf{1})}{\text{sum}(M)} + b, & \text{if } \text{sum}(M) > 0 \\ 0, & \text{if otherwise} \end{cases}$$

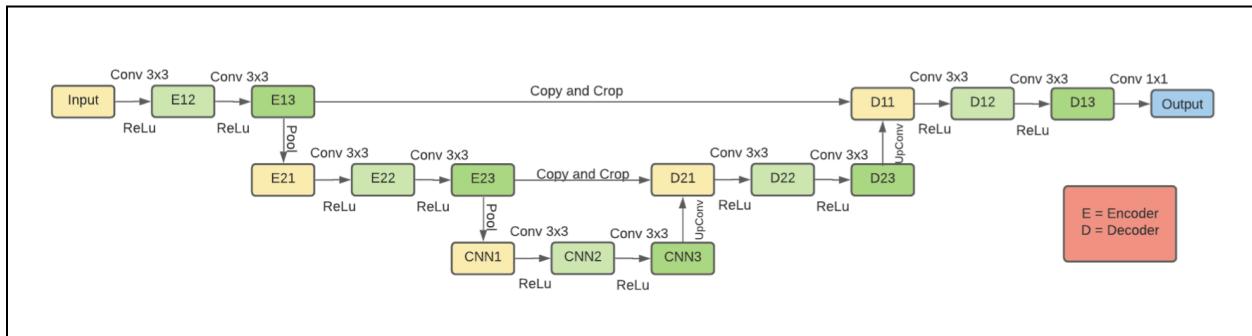
Perhitungan Partial Convolution

Mask Update Step

Setelah setiap Partial Convolution layer diikuti dengan mask update step. Mask update step adalah langkah untuk memperbarui nilai mask dari 0 menjadi 1, jika ada setidaknya satu masukan valid yang sesuai dengan piksel yang dipertimbangkan.

$$m' = \begin{cases} 1, & \text{if } \text{sum}(M) \\ 0, & \text{if otherwise} \end{cases}$$

UNet Architecture



Arsitektur UNet

Model Partial Convolutional based Autoencoder adalah model yang kompleks, terdiri dari generator dan discriminator yang memanfaatkan lapisan partial convolution dan membentuk arsitektur UNet. Arsitektur UNet sendiri terdiri dari beberapa lapisan yang diatur sedemikian rupa sehingga membentuk struktur seperti huruf U. Di dalamnya, terdapat jalur kontraksi dan jalur ekspansi yang masing-masing dilengkapi dengan encoder dan decoder. Jalur kontraksi bertanggung jawab untuk mengekstraksi fitur-fitur penting dari gambar asli, sementara jalur ekspansi bertugas untuk merekonstruksi gambar dengan menambahkan detail-detail yang hilang. Dalam arsitektur ini, setiap lapisan dangkal (encoder) terhubung secara langsung ke lapisan yang lebih dalam (decoder) melalui skip links. Hal ini memungkinkan aliran informasi yang langsung dan lebih efisien antara kedua bagian arsitektur tersebut. Dengan memanfaatkan koneksi ini, model Partial Convolutional based Autoencoder dapat menghasilkan hasil inpainting yang lebih akurat dan detail.

c) GAN (Generative Adversarial Network)

Referensi:

<https://github.com/AdeelH/gan-image-inpainting/tree/main>

http://iizuka.cs.tsukuba.ac.jp/projects/completion/data/completion_sig2017.pdf

Metode atau pendekatan ini menggunakan dua buah discriminator untuk melakukan adversarial training, dimana satu discriminator berfokus pada kualitas gambar yang telah selesai secara keseluruhan and discriminator lainnya berfokus pada wilayah lokal yang lebih kecil yang berisi bagian dari gambar.

Metode ini dipilih karena GAN dikenal dengan kemampuannya untuk menghasilkan gambar yang terlihat realistik dan berkualitas tinggi. Adversarial training mendorong generator untuk menghasilkan gambar yang tidak dapat dibedakan dari gambar nyata, yang sangat berguna untuk image inpainting yang tujuannya adalah mengisi bagian gambar yang hilang dengan mulus. Adapun penggunaan global discriminator dapat memastikan gambar yang dihasilkan secara keseluruhan terlihat realistik dan koheren. Kemudian, local discriminator memastikan hasil inpainting pada bagian yang hilang terlihat realistik pada skala yang lebih kecil (tekstur lebih tajam dan detail).

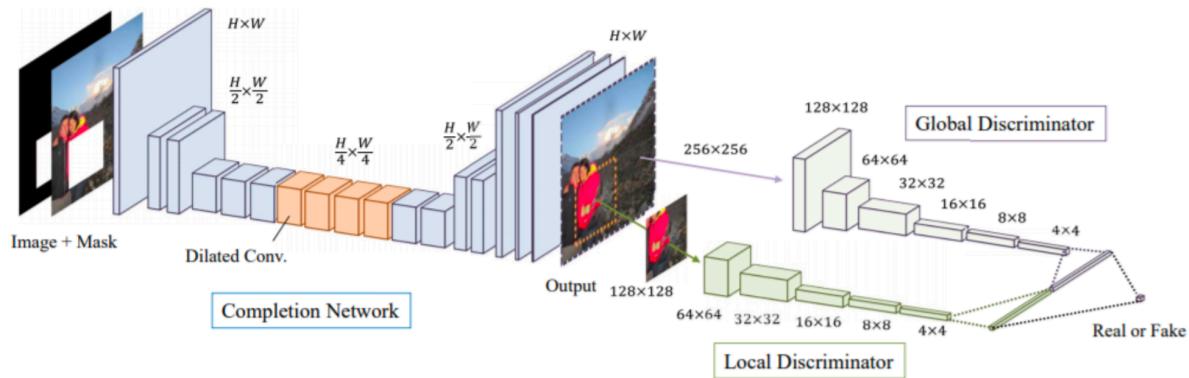
Generator

Generator-nya menggunakan convolutional layer dan arsitekturnya menyerupai autoencoder, dimana dilakukan sampling pada gambar masukan hingga $\frac{1}{4}$ ukuran aslinya menggunakan convolutional layer dengan stride 2 dan diperbesar lagi dengan lapisan deconvolution sehingga sebesar ukuran awal lagi. Ditengah generator, terdapat dilated convolution untuk meningkatkan bidang reseptif setiap neuron.

Discriminator

Ada dua macam discriminator yang digunakan:

- Global Discriminator: menerima keseluruhan gambar sebagai masukan dan menerapkan konvolusi dengan stride 2 dan memiliki lapisan linear berukuran 1024 sebagai lapisan terakhir.
- Local Discriminator: arsitekturnya mirip global discriminator, namun masukannya adalah bagian kecil dari gambar yang berisi dan berpusat di sekitar wilayah yang hilang yang telah diisi oleh generator.



Arsitektur GAN yang Digunakan

Masks

Mask digunakan untuk merepresentasikan bagian yang hilang pada gambar. Mask berupa array dengan nilai 0, di mana pada bagian yang hilang nilainya 1. Untuk pelatihan, mask dibuat secara

acak. Pada metode ini, bentuk mask yang digunakan adalah persegi panjang dengan tinggi dan lebar acak, yang berkisar antara 5 hingga 12 piksel.

Training

Untuk setiap batch, dihasilkan random masks. Masks yang dihasilkan digabung dengan gambar masukan (input). Proses pelatihannya dibagi menjadi 3 fase:

- Generator-only training dengan menggunakan squared error loss.
- Discriminator-only training dengan bobot Generator dibekukan
- Combined training dengan full GAN + MSE loss untuk generator and GAN loss untuk discriminator.

2. Metode Pengujian

Untuk menguji kinerja model yang dikembangkan, terdapat beberapa metrik pengukuran yang digunakan antara lain:

a) Mean Absolute Error (MAE)

Mean Absolute Error (MAE), atau bisa disebut juga L1 loss, adalah loss function dan metrik evaluasi yang paling sederhana dan mudah dipahami. Nilai MAE dihitung dengan cara menemukan rata-rata dari selisih absolut antara nilai yang diprediksi dan nilai sebenarnya pada seluruh kumpulan data. Secara matematis, MAE merepresentasikan rata-rata aritmatika dari absolute errors, yang hanya memperhatikan magnitudonya, tanpa memperdulikan arahnya. Nilai MAE yang rendah menandakan akurasi model yang lebih baik.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Rumus Mean Absolute Error (MAE)

dimana:

- y_i = Nilai sebenarnya
- \hat{y}_i = Nilai hasil prediksi
- n = banyak data pada kumpulan data

b) Mean Square Error (MSE)

Mean Square Error (MSE), atau bisa disebut juga L2 loss, adalah loss function yang paling umum dan metrik evaluasi yang penting. MSE menghitung error dengan cara menghitung perbedaan antara nilai prediksi dan nilai sebenarnya, kemudian dikuadratkan dan dijumlahkan untuk semua data yang ada. MSE juga dikenal sebagai Quadratic loss karena penalty-nya tidak sebanding dengan error-nya, tetapi dengan kuadrat dari error. Proses mengkuadratkan error membuat data outlier memiliki pengaruh yang lebih besar, menghasilkan gradien yang lebih halus untuk kesalahan kecil. Algoritma optimisasi memanfaatkan penalti ini untuk kesalahan besar karena dapat membantu menemukan nilai optimal untuk parameter menggunakan metode least squares. Nilai MSE tidak akan pernah negatif karena error-nya dipangkatkan. Nilai error berkisar dari nol hingga tak terhingga. MSE meningkat secara eksponensial dengan peningkatan error. Model yang baik akan memiliki nilai MSE yang mendekati nol, menunjukkan kecocokan yang lebih baik dengan data.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Rumus Mean Square Error (MSE)

dimana:

- y_i = Nilai sebenarnya
- \hat{y}_i = Nilai hasil prediksi
- n = banyak data pada kumpulan data

c) Dice Coefficient

Dice Coefficient, atau bisa disebut juga “Sørensen–Dice coefficient” adalah metrik pengukuran yang menghitung kemiripan antara dua set, A dan B. Nilai dice coefficient berkisar dari 0 hingga 1, dimana 1 menunjukkan bahwa kedua himpunan tersebut identik, dan 0 menunjukkan bahwa kedua himpunan tersebut tidak tumpang tindih.

$$Dice\ coefficient = 2 * |A \cap B| / (|A| + |B|)$$

Rumus Dice Coefficient

dimana:

- $|A|$ = Jumlah elemen pada set A
- $|B|$ = jumlah elemen pada set B
- $|A \cap B|$ = jumlah elemen yang terdapat pada kedua set

Eksperimen

Publikasi (Bonus)

Source code hasil eksplorasi tugas ini, sudah dipublikasikan pada git repository publik berikut:
<https://github.com/nabilahannania/image-inpainting-generative.git>

Lingkungan Eksperimen

Eksperimen dalam tugas ini dilakukan pada platform Kaggle dengan menggunakan GPU NVIDIA T4 16 GB.

Detail Eksperimen

Pada tugas ini, terdapat tujuh eksperimen yang dilakukan, dimana untuk tiap eksperimen terdapat perbedaan pada jenis model yang digunakan, jenis masks, atau ukuran batch size. Eksperimen 1 - 3 menggunakan model Convolutional based Autoencoder, eksperimen 4 - 5 menggunakan model Partial Convolutional based Autoencoder, dan eksperimen 6 - 7 menggunakan model GAN.

1. Eksperimen 1

Model : Convolutional based Autoencoder

Jenis masking: Regular hole (box) mask pada tengah gambar

Batch size: 32

Epoch: 20

Lama training: 14 menit 30 detik

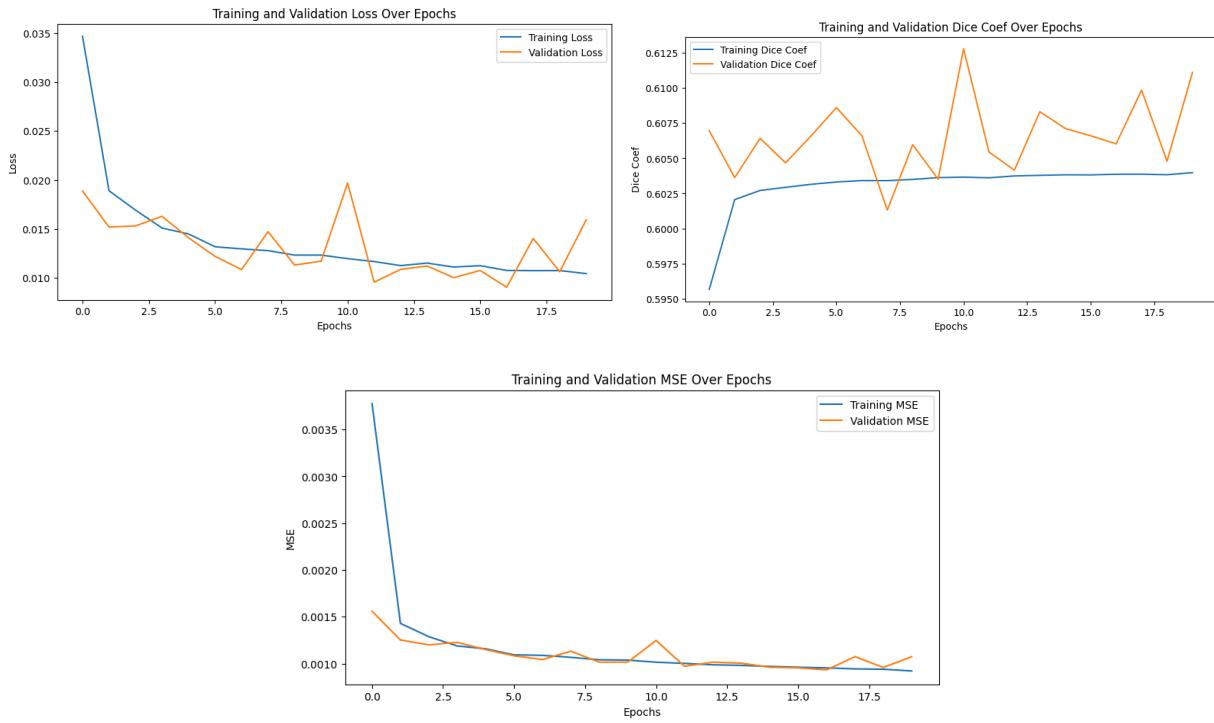
```
Epoch 20/20
1250/1250 [=====] - 40s 32ms/step - loss: 0.0104 - dice_coef: 0.6040 - mse: 9.2067e-04 - val_loss: 0.0159 -
Training time: 872.3138008117676 seconds
```

Detail pelatihan dan pengimplementasian eksperimen ini, dapat dilihat pada notebook [convolutional_autoencoder_regular_center.ipynb](#)

Contoh masukan (hasil masking):



Visualisasi hasil pelatihan:



Dapat dilihat dari plot yang ditampilkan, nilai loss dan error dari data training dan validation seiring dengan bertambahnya epoch secara umum menurun. Sedangkan untuk nilai dice coef pada data training tidak mengalami banyak perubahan setelah epoch pertama dan nilai dice coef pada data validation berubah - ubah, namun tidak ada pola yang jelas.

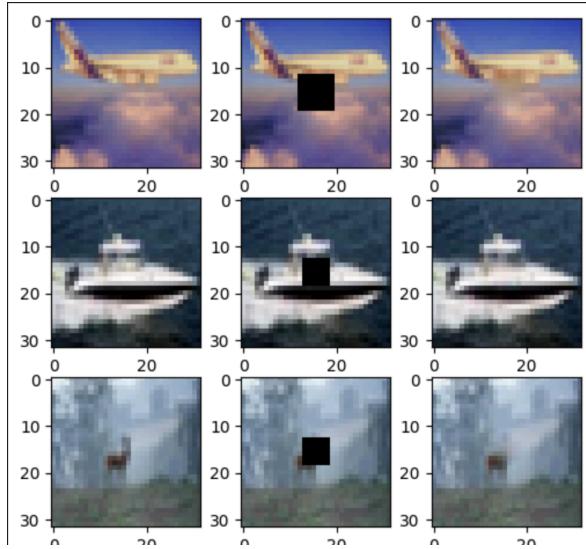
Hasil evaluasi pada data test:

Test loss (MAE): 0.015979589894413948

Test dice coef: 0.6109957098960876

Test MSE: 0.001079045468941331

Hasil prediksi beberapa sample:



2. Eksperimen 2

Model : Convolutional based Autoencoder

Jenis masking: Regular hole (box) mask pada lokasi acak pada gambar.

Batch size: 32

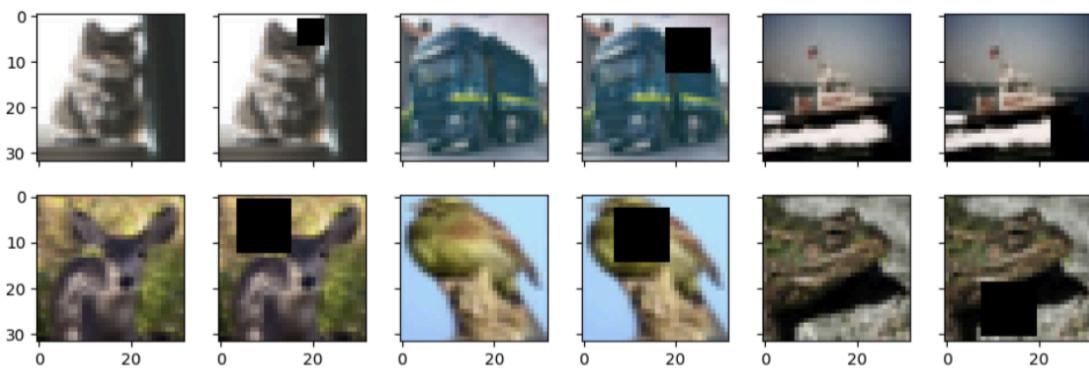
Epoch: 20

Lama training: 14 menit 36 detik

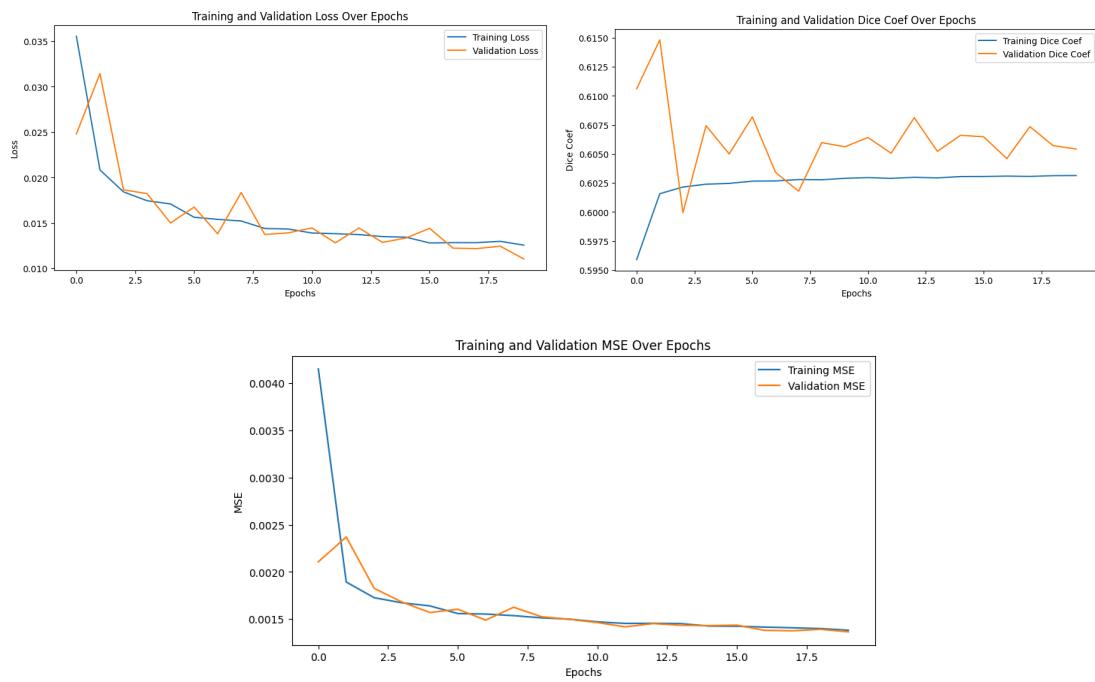
```
Epoch 20/20
1250/1250 [=====] - 42s 33ms/step - loss: 0.0126 - dice_coef: 0.6031 - mse: 0.0014 - val_loss: 0.0111 - val_dice_coef: 0.6031 - val_mse: 0.0014
Training time: 880.0207540988922 seconds
```

Detail pelatihan dan pengimplementasian eksperimen ini, dapat dilihat pada notebook [convolutional_autoencoder_regular_random.ipynb](#)

Contoh masukan (hasil masking):



Visualisasi hasil pelatihan:



Dapat dilihat dari plot yang ditampilkan, nilai loss dan error dari data training dan validation seiring dengan bertambahnya epoch secara umum menurun. Sedangkan untuk nilai dice coef pada data training tidak mengalami banyak perubahan setelah epoch pertama dan nilai dice coef pada data validation berubah - ubah, namun cenderung menurun.

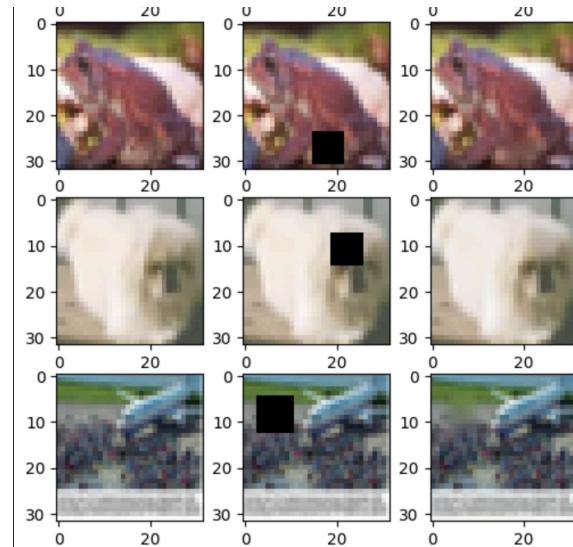
Hasil evaluasi pada data test:

Test loss (MAE): 0.011029115878045559

Test dice coef: 0.6054282188415527

Test MSE: 0.001355905900709331

Hasil prediksi beberapa sample:



3. Eksperimen 3

Model : Convolutional based Autoencoder

Jenis masking: Irregular hole mask pada lokasi acak pada gambar.

Batch size: 32

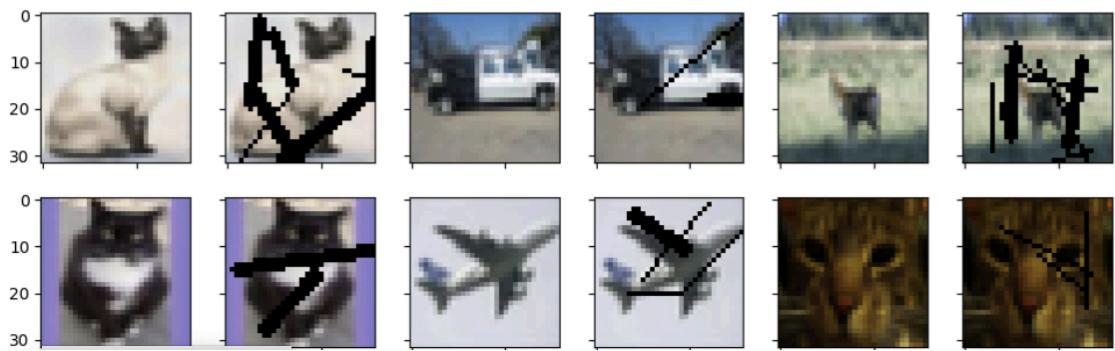
Epoch: 20

Lama training: 17 menit

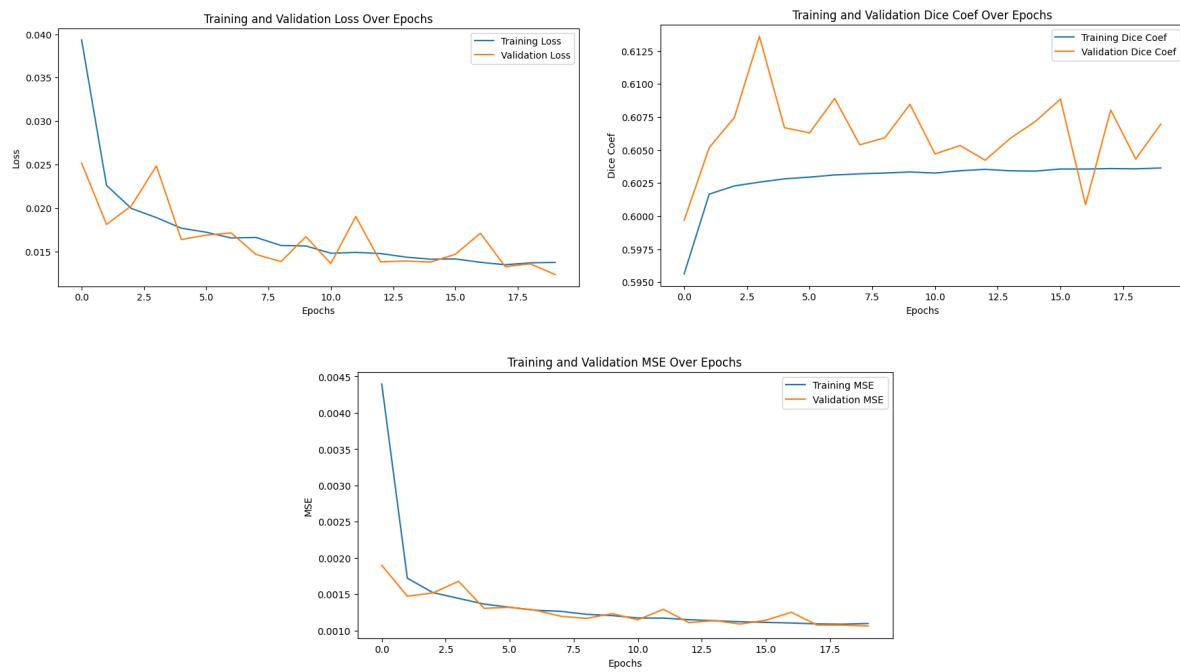
```
Epoch 20/20
1250/1250 [=====] - 49s 39ms/step - loss: 0.0138 - dice_coef: 0.6036 - mse: 0.0011 - val_loss: 0.0124 - val_dice_coef: 0.6036 - val_mse: 0.0011
Training time: 1018.4692196846008 seconds
```

Detail pelatihan dan pengimplementasian eksperimen ini, dapat dilihat pada notebook [convolutional_autoencoder_irregular.ipynb](#)

Contoh masukan (hasil masking):



Visualisasi hasil pelatihan:

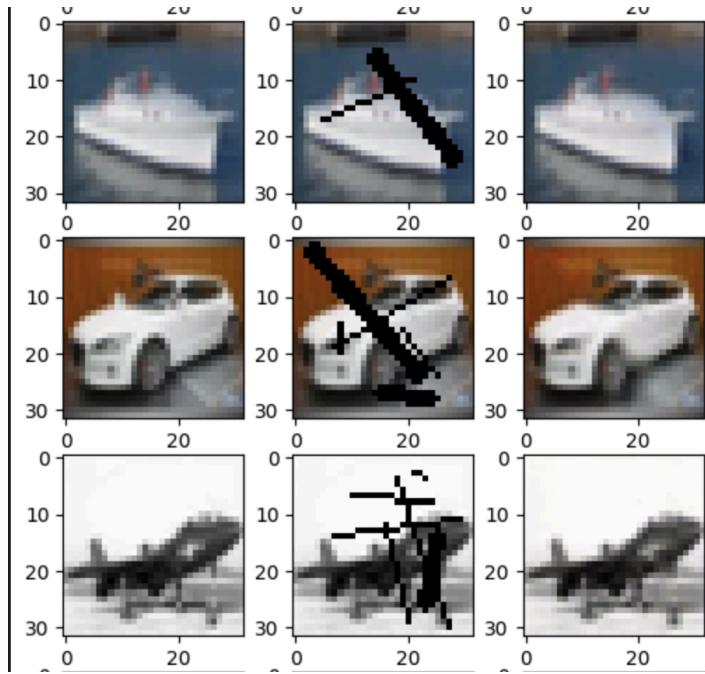


Dapat dilihat dari plot yang ditampilkan, nilai loss dan error dari data training dan validation seiring dengan bertambahnya epoch secara umum menurun. Sedangkan untuk nilai dice coef pada data training tidak mengalami banyak perubahan setelah epoch pertama dan nilai dice coef pada data validation berubah - ubah, namun cenderung menurun.

Hasil evaluasi pada data test:

Test loss (MAE): 0.012383288703858852
 Test dice coef: 0.6070067286491394
 Test MSE: 0.001062931027263403

Hasil prediksi beberapa sample:



4. Eksperimen 4

Model : Partial Convolutional based Autoencoder

Jenis masking: Regular hole (box) mask pada lokasi acak pada gambar

Batch size: 32

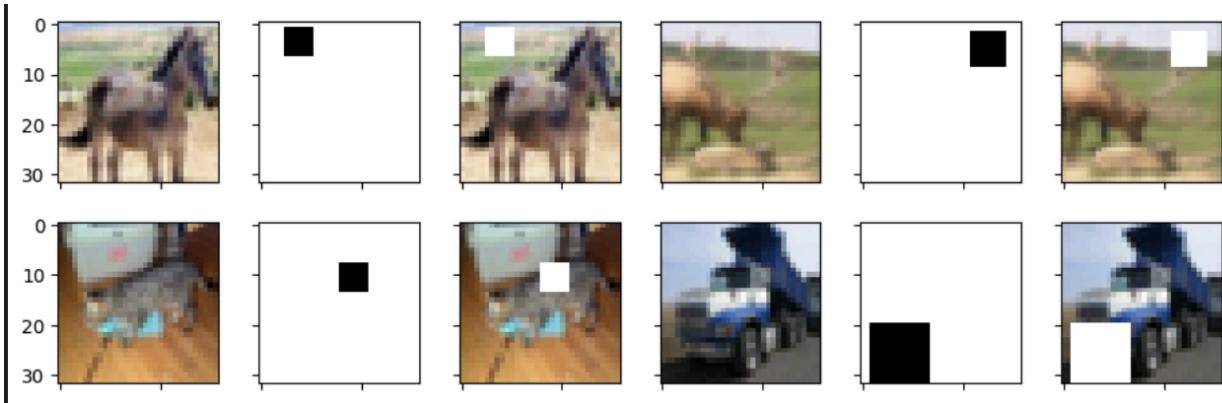
Epoch: 30

Lama training: 30 menit 30 detik

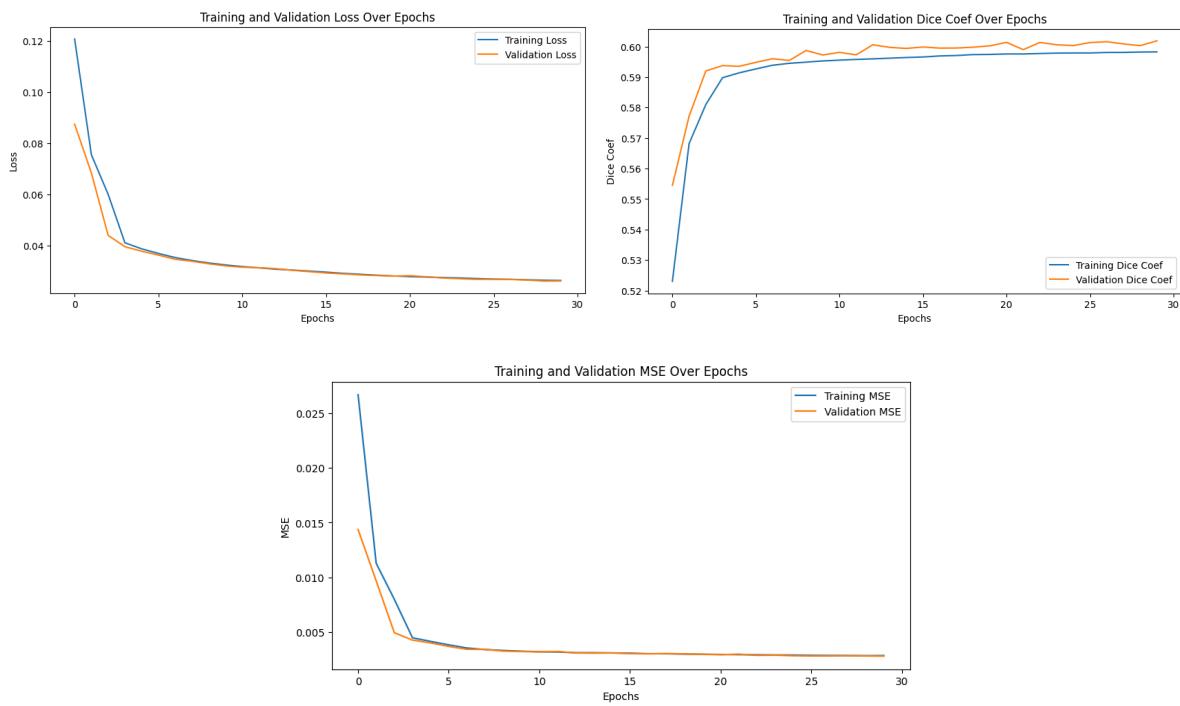
```
Epoch 30/30
1250/1250 [=====] - 57s 46ms/step - loss: 0.0264 - dice_coef: 0.5982 - mse: 0.0028 - val_loss: 0.0262 - val_dice_coef: 0.5982 - val_mse: 0.0028
Training time: 1834.614370584488 seconds
```

Detail pelatihan dan pengimplementasian eksperimen ini, dapat dilihat pada notebook [partial convolution autoencoder reguler center.ipynb](#)

Contoh masukan (hasil masking):



Visualisasi hasil pelatihan:



Dapat dilihat dari plot yang ditampilkan, nilai loss dan error dari data training dan validation seiring dengan bertambahnya epoch secara umum menurun. Sedangkan untuk nilai dice coef pada data training dan validation tidak mengalami banyak perubahan setelah epoch ketiga, namun cenderung meningkat.

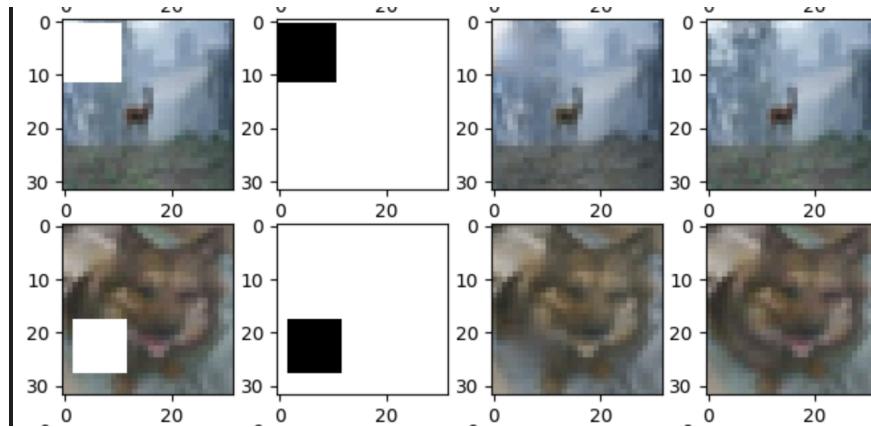
Hasil evaluasi pada data test:

Test loss (MAE): 0.02639421820640564

Test dice coef: 0.6020033955574036

Test MSE: 0.0028363869059830904

Hasil prediksi beberapa sample:



5. Eksperimen 5

Model : Partial Convolutional based Autoencoder

Jenis masking: Irregular hole mask pada lokasi acak pada gambar

Batch size: 32

Epoch: 30

Lama training: 31 menit 35 detik

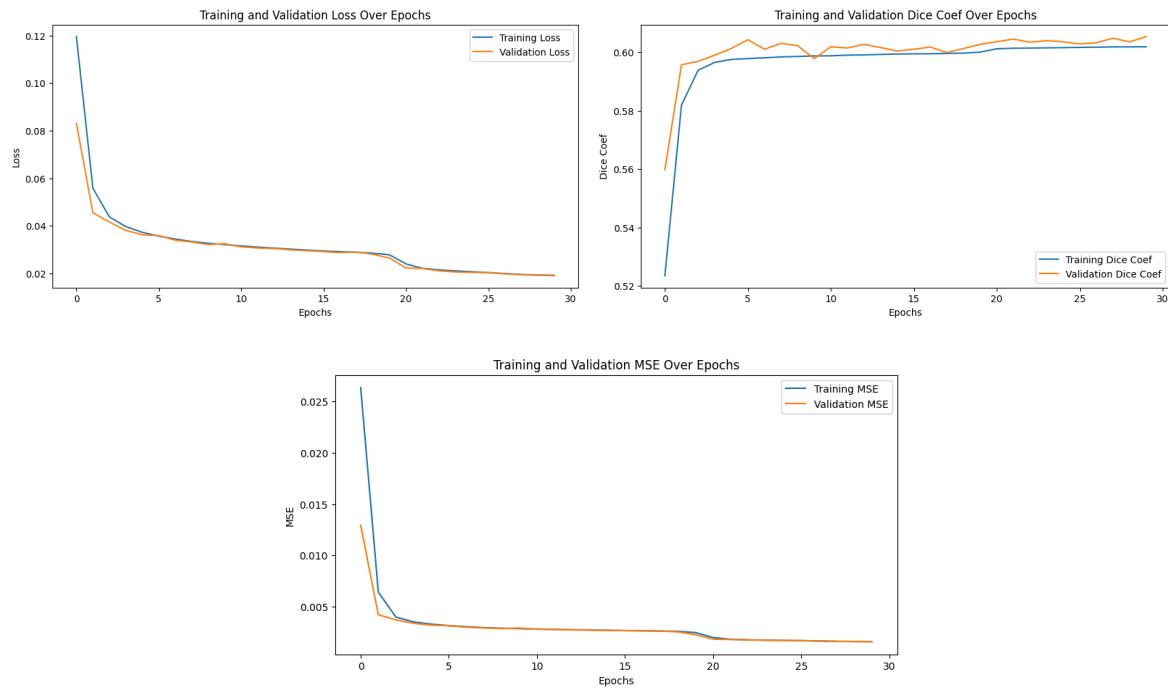
```
Epoch 30/30
1250/1250 [=====] - 58s 46ms/step - loss: 0.0192 - dice_coef: 0.6019 - mse: 0.0016 - val_loss: 0.0191 - val_dice_coef: 0.6019 - val_mse: 0.0016
Training time: 1896.3920753002167 seconds
```

Detail pelatihan dan pengimplementasian eksperimen ini, dapat dilihat pada notebook [partial_convolution_autoencoder_irreguler.ipynb](#)

Contoh masukan (hasil masking):



Visualisasi hasil pelatihan:



Dapat dilihat dari plot yang ditampilkan, nilai loss dan error dari data training dan validation seiring dengan bertambahnya epoch secara umum menurun. Sedangkan untuk nilai dice coef pada data training dan validation tidak mengalami banyak perubahan setelah epoch ketiga, namun cenderung meningkat.

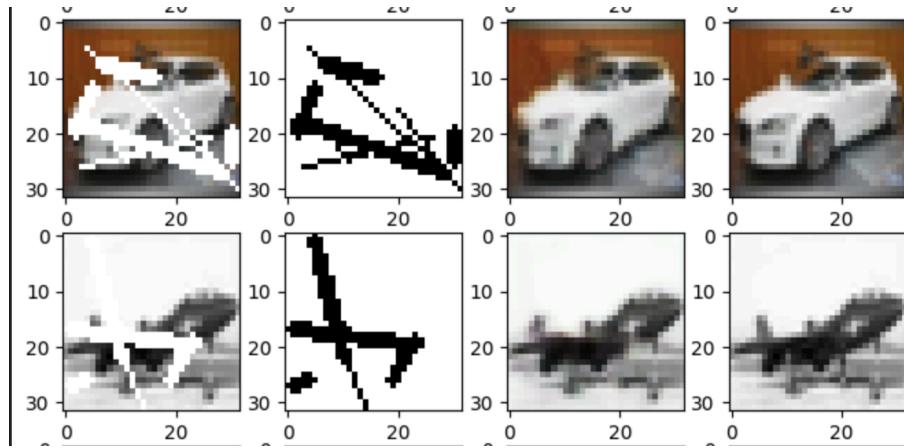
Hasil evaluasi pada data test:

Test loss (MAE): 0.019119178876280785

Test dice coef: 0.6052913665771484

Test MSE: 0.0015762039693072438

Hasil prediksi beberapa sample:



6. Eksperimen 6

Model : GAN

Jenis masking: Regular hole (box) mask pada lokasi acak pada gambar

Batch size: 32

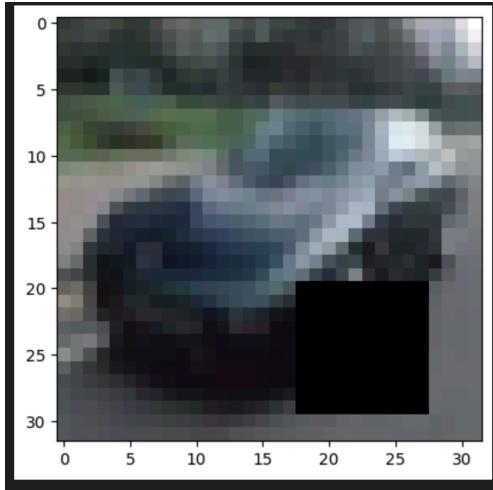
Epoch: 30

Lama training: 72 menit 50 detik

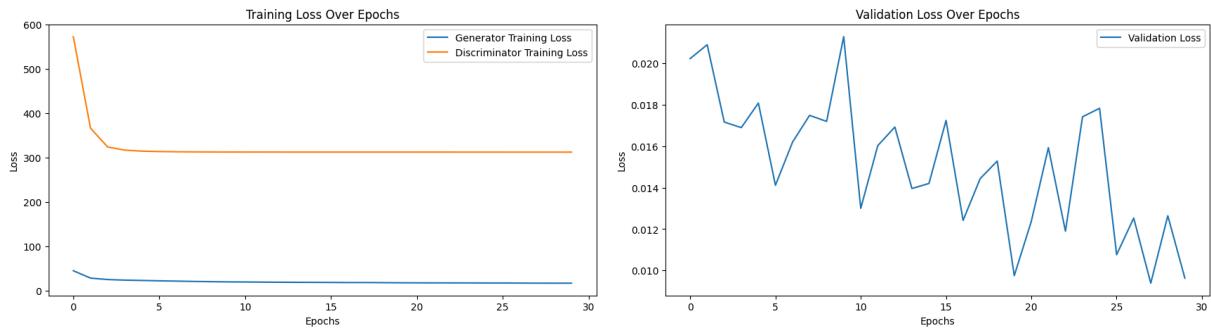
```
epoch: 29, g_loss: 16.7888, val_loss: 0.0096, d_loss: 312.5156, fake_err: 312.8291, real_err: 312.2045, time: 145.06
Training time: 4369.1283802986145 seconds
```

Detail pelatihan dan pengimplementasian eksperimen ini, dapat dilihat pada notebook [gan_regular_center_b32.ipynb](#)

Contoh masukan (hasil masking):



Visualisasi hasil pelatihan:

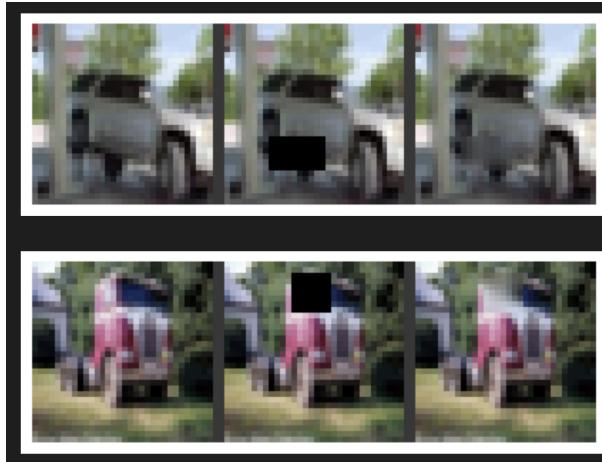


Dapat dilihat dari plot yang ditampilkan, nilai loss dari generator dan discriminator tidak mengalami banyak perubahan setelah epoch ketiga, namun ada penurunan yang sangat kecil. Sedangkan nilai loss pada data validation seiring dengan bertambahnya epoch secara umum menurun.

Hasil evaluasi pada data test:

```
Mean Squared Error (MSE): tensor(0.0129)
Mean Absolute Error (MAE): tensor(0.0792)
Dice Coefficient: 0.5755467746051912
```

Hasil prediksi beberapa sample:



7. Eksperimen 7

Model : GAN

Jenis masking: Regular hole (box) mask pada lokasi acak pada gambar

Batch size: 800

Epoch: 30

Lama training: 72 menit 20 detik

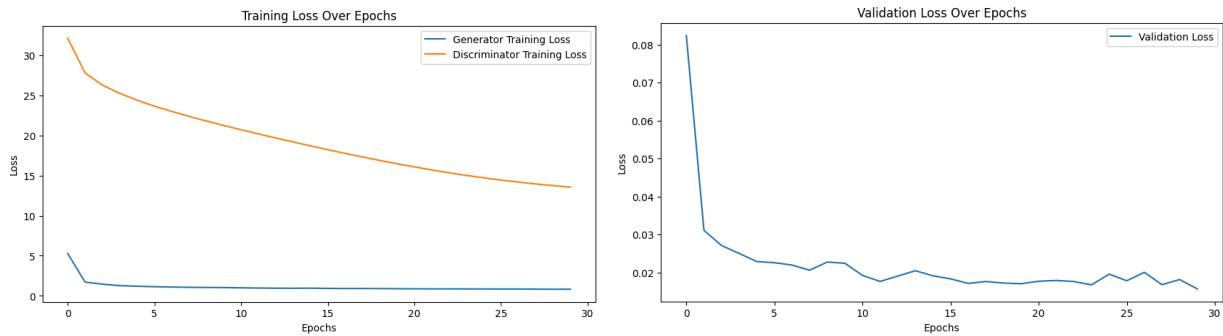
```
epoch: 29, g_loss: 0.8338, val_loss: 0.0157, d_loss: 13.5558, fake_err: 6.8411, real_err: 20.2706, time: 144.45
Training time: 4333.893573284149 seconds
```

Detail pelatihan dan pengimplementasian eksperimen ini, dapat dilihat pada notebook [gan_regular_center_b800.ipynb](#)

Contoh masukan (hasil masking):



Visualisasi hasil pelatihan:

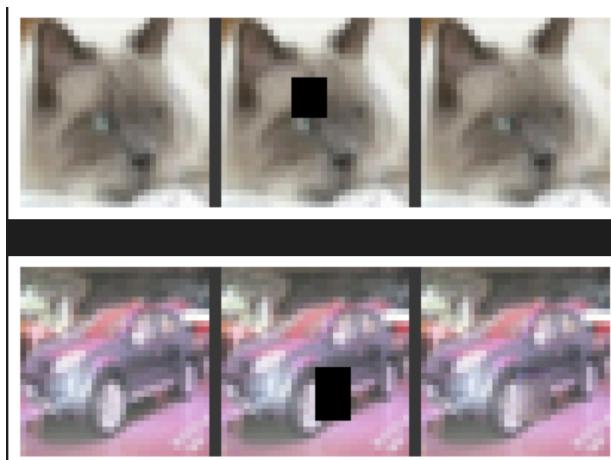


Dapat dilihat dari plot yang ditampilkan, nilai loss dari discriminator menurun cukup signifikan seiring bertambahnya epoch, namun nilai loss dari generator tidak mengalami banyak perubahan setelah epoch ketiga (penurunan sangat kecil). Sedangkan nilai loss pada data validation seiring dengan bertambahnya epoch secara umum menurun.

Hasil evaluasi pada data test:

Mean Squared Error (MSE): tensor(0.0161)
 Mean Absolute Error (MAE): tensor(0.0906)
 Dice Coefficient: 0.5637718994532561

Hasil prediksi beberapa sample:



Analisis Hasil Eksperimen

Tabel Rekap Hasil Pengujian Pada Data Test

Model	Masks	Epoch	Batch	MSE ↓	MAE↓	Dice C.↑
Convolutional based Autoencoder	Regular Center	20	32	0.00108	0.01598	0.61100
Convolutional based Autoencoder	Regular Random	20	32	0.00136	0.01103	0.60543
Convolutional based Autoencoder	Irregular	20	32	0.00106	0.01238	0.60701
Partial Convolutional based Autoencoder	Regular Random	30	32	0.00284	0.02639	0.60200
Partial Convolutional based Autoencoder	Irregular	30	32	0.00158	0.01912	0.60529
GAN	Regular Random	30	32	0.01290	0.07920	0.57555
GAN	Regular Random	30	800	0.01610	0.09060	0.56377

- Model yang memberikan kinerja paling baik pada seluruh metrik pengukuran adalah model Convolutional based Autoencoder. Meskipun untuk tiap metrik, pasangan jenis masks yang memberikan hasil terbaik berbeda beda (model Convolutional based Autoencoder baik untuk tiap jenis masking). Namun secara keseluruhan, model Convolutional based Autoencoder memberikan hasil inpainting paling baik. Hal ini menyatakan bahwa, model dengan vanilla CNN dapat pola gambar dengan lebih baik untuk dataset yang cukup sederhana (resolusi kecil).
- Model Partial Convolutional based Autoencoder memberikan hasil yang sedikit lebih buruk dibandingkan model Convolutional based Autoencoder. Padahal, model dengan partial convolution dilatih dengan epoch yang lebih banyak (pelatihannya lebih lama).

Hal ini dikarenakan Model Partial Convolutional based Autoencoder terlalu kompleks untuk data CIFAR-10 yang cukup sederhana. Lapisan partial convolution dirancang untuk gambar beresolusi tinggi yang lebih besar dari 256x256 piksel. Permasalahan ini sebenarnya bisa ditangani dengan menambah loss function lain dalam pelatihan (tidak hanya menggunakan L2 norm, tapi juga Total Variation Loss, Perceptual Loss, Style Loss, Pixel Losses)

- Pada model Partial Convolutional based Autoencoder, data dengan irregular masks memberikan hasil lebih baik dibandingkan dengan data regular mask pada semua metrik pengukuran. Hal ini karena partial convolution layers berfokus pada unmasked pixels, sehingga bentuk mask yang tidak beraturan lebih cocok.
- Model GAN memberikan hasil yang paling buruk dibandingkan model lainnya dengan perbedaan error yang cukup jauh. Hal ini dikarenakan model GAN memiliki kompleksitas paling tinggi, jika dibandingkan model lain, sehingga tidak cocok dengan data CIFAR-10 yang sederhana.
- Pada model GAN, penggunaan batch size yang lebih kecil menghasilkan model dengan kinerja yang lebih baik. Hal ini karena batch size yang lebih kecil dapat membantu meningkatkan stabilitas pelatihan GAN. Dengan menggunakan batch size yang lebih kecil, perubahan dalam setiap batch menjadi lebih kecil, sehingga lebih mudah bagi model untuk menyesuaikan parameter dan menghindari divergensi atau mode collapse.

Kesimpulan

Berdasarkan eksperimen yang telah dilakukan pada tugas ini, terdapat beberapa hal yang dapat disimpulkan:

1. Untuk melakukan tugas image inpainting dikembangkan tiga jenis model, yaitu Convolutional based Autoencoder, Partial Convolutional based Autoencoder, dan GAN. Adapun model yang memberikan kinerja terbaik untuk data CIFAR-10 adalah model Convolutional based Autoencoder karena memiliki arsitektur yang lebih sederhana, yang lebih sesuai untuk dataset yang relatif kecil.
2. Lapisan partial convolution pada model Partial Convolutional based Autoencoder berfokus pada unmasked pixels, sehingga cocok untuk mengatasi mask dengan bentuk tidak teratur.
3. Penggunaan ukuran batch yang lebih kecil dapat meningkatkan stabilitas pelatihan pada model GAN dengan 1 generator dan 2 discriminator, terutama pada dataset yang relatif kecil.

Hal yang sudah dipelajari selama melakukan eksplorasi:

1. Mempelajari mengenai konsep image inpainting, yaitu tugas atau teknik untuk merekonstruksi gambar dengan mengisi bagian yang hilang.
2. Mempelajari adanya variasi jenis lapisan convolution, yaitu partial convolution yang hanya memperhatikan unmasked pixels.
3. Dalam pembangunan model image inpainting (dan deep learning secara umum), pemilihan arsitektur model harus mempertimbangkan dataset yang digunakan. Jika model terlalu kompleks, sedangkan datanya sederhana, hasil modelnya tidak akan terlalu bagus (Pentingnya mempertimbangkan kompleksitas model sesuai dengan dataset yang digunakan).
4. Mempelajari adanya variasi model GAN yang memiliki 1 generator dan 2 discriminator.

Aspek yang dapat dieksplorasi dan diteliti ke depannya:

1. Penggunaan dataset yang lebih besar dan kompleks, seperti ImageNet atau CelebA, untuk melatih model yang ada.

2. Penggunaan irregular masking pada model GAN untuk memeriksa kemampuan model dalam mengatasi mask dengan bentuk tidak teratur.