

Analisis Metode Pix2Pix untuk Menyelesaikan Masalah Image to Image Translation pada Citra Satelit dan Citra Peta

Silverius Sony Lembang

Muhammad Takdim

Richard Enrico Sulieanto

CONTENTS

I	Pendahuluan	1
II	Rumusan Masalah	1
III	Tinjauan Pustaka	1
III-A	Generative Adversarial Network	1
III-B	Conditional Generative Adversarial Network	1
III-C	Pix2Pix GAN	2
III-D	U-Net	2
III-E	PatchGAN Discriminator	2
III-F	Loss Function	2
IV	Metode Penelitian	3
IV-A	Dataset	3
IV-B	Model	3
V	Hasil dan Pembahasan	3
VI	Kesimpulan	3
Appendix		3
A	Implementasi Model	3
B	Detail Pelatihan	4
References		4

LIST OF FIGURES

1	Arsitektur U-Net	2
2	Hasil Segmentasi Citra menggunakan <i>U-Net</i>	2
3	Hasil transformasi gambar peta menjadi citra satelit	3
4	Hasil transformasi citra satelit menjadi gambar peta	4

LIST OF TABLES

Analisis Metode Pix2Pix untuk Menyelesaikan Masalah Image to Image Translation pada Citra Satelit dan Citra Peta

Abstract—*Image to image translation* adalah masalah yang timbul dimana kita ingin memetakan suatu gambar *input* menjadi gambar *output* yang baru. Metode yang digunakan pada masalah ini adalah Pix2Pix yang merupakan implementasi dari metode *Conditional GAN*. Dalam *technical report* ini, kami melakukan penelitian terhadap masalah *image to image translation* dimana dataset yang digunakan adalah *dataset maps* yang berisi citra satelit dan citra peta yang merupakan representasi dari citra satelit. Berdasarkan penelitian ini, diperoleh hasil bahwa citra yang dihasilkan dari metode Pix2Pix ini cukup baik dimana citra yang dihasilkan mirip dengan citra aslinya.

I. PENDAHULUAN

Banyak masalah dalam *image processing*, *computer graphics* serta *computer vision* yang membutuhkan pengubahan dari *input image* menjadi *output image* yang sesuai. Misalnya *super resolution* dapat dianggap sebagai masalah pemetaan gambar *low resolution* ke *high resolution*, *colorizing* dapat dianggap sebagai masalah pemetaan gambar *gray scale* yang akan menjadi *color image* [2].

Adapun metode yang dapat digunakan untuk melakukan pemetaan dari suatu gambar *input* menjadi sebuah gambar *output* adalah metode *pix2pix*. *Pix2Pix GAN* telah didemonstrasikan pada berbagai tugas terjemahan gambar-ke-gambar seperti mengubah peta menjadi foto satelit, foto hitam putih menjadi warna, dan sketsa produk menjadi foto produk. Metode ini mampu menghasilkan citra yang sangat mirip dengan citra aslinya. Hal ini menunjukkan bahwa metode ini cukup ampuh untuk menyelesaikan masalah yang berkaitan dengan *image to image translation*. Dalam penelitian ini, kami berfokus untuk mengimplementasikan metode Pix2Pix pada *dataset map*. Diharapkan hasil yang diperoleh mampu menciptakan gambar sintetis yang menyerupai gambar aslinya yakni menghasilkan citra peta sintetis dan juga citra satelit sintetis.

II. RUMUSAN MASALAH

Dalam *technical report* ini, kami berfokus untuk mengetahui permasalahan dari *image to image translation*, yakni menerjemahkan gambar peta menjadi citra satelit dan sebaliknya. Dimana hal tersebut menjadi masalah yang menantang yang membutuhkan pengembangan model khusus dan *loss function* untuk jenis *translation* yang dilakukan.

III. TINJAUAN PUSTAKA

A. Generative Adversarial Network

GAN pertama kali diperkenalkan oleh [7] sebagai cara baru untuk melatih model generatif. Arsitektur GAN terdiri dari dua model, yaitu:

- Model generatif G yang menangkap distribusi data.
- Model diskriminatif D yang memperkirakan probabilitas bahwa sampel berasal dari data pelatihan dan bukan dari G .

Baik G dan D bisa menjadi fungsi pemetaan non-linier, seperti *multi-layer perceptron*.

Untuk mempelajari distribusi generator p_g melalui data x . generator membangun fungsi pemetaan dari distribusi *noise* sebelumnya $p_z(Z)$ ke ruang data sebagai $G(z; \theta_g)$. Dan *discriminator* $D(x; \theta_d)$, mengeluarkan skalar tunggal yang mewakili probabilitas bahwa x berasal dari data pelatihan, bukan p_g .

G dan D keduanya dilatih secara bersamaan, lalu penyesuaian parameter G untuk meminimalkan $\log 1 - D(G(z))$ dan penyesuaian parameter D untuk meminimalkan $\log D(X)$ seolah-olah mereka mengikuti min-max dua pemain permainan dengan nilai fungsi $V(G, D)$:

$$\min_G \max_D V(D, G) = \underset{G}{\text{Ex} \sim p_{\text{data}}(x)} [\log D(x)] + \underset{D}{E_{z \sim p_z(z)} [\log (1 - D(G(z)))]} \quad (1)$$

B. Conditional Generative Adversarial Network

GAN dapat diperluas ke model bersyarat jika generator dan diskriminator dikondisikan pada beberapa informasi tambahan y . y dapat berupa informasi tambahan apa pun, seperti label kelas atau data dari modalitas lain. Lalu dilakukan pengkondisian dengan memasukkan y ke diskriminator dan generator sebagai lapisan input tambahan.

Dalam generator, noise input sebelumnya $p_z(z)$, dan y digabungkan dalam representasi tersembunyi bersama, dan kerangka kerja pelatihan memungkinkan fleksibilitas yang cukup besar dalam bagaimana representasi tersembunyi ini disusun.

Dalam diskriminator x dan y disajikan sebagai input dan fungsi diskriminatif (diwujudkan lagi oleh MLP dalam kasus ini).

Fungsi tujuan dari permainan *minimax* dua pemain adalah sebagai Equation 2

$$\min_G \max_D V(D, G) = \underset{G}{\text{Ex} \sim p_{\text{data}}(x)} [\log D(x|y)] + \underset{D}{E_{z \sim p_z(z)} [\log (1 - D(G(z|y)))]} \quad (2)$$

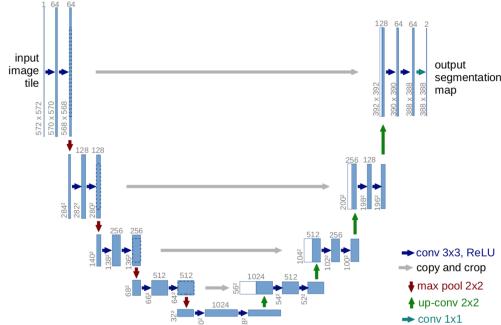


Fig. 1. Arsitektur U-Net

C. Pix2Pix GAN

Pix2Pix GAN sendiri berarti *pixel to pixel* yang berarti dalam sebuah gambar dibutuhkan satu *pixel*, lalu mengubahnya menjadi *pixel* lain. Tujuan dari model ini adalah untuk mengkonversi dari satu gambar ke gambar lain atau bisa juga dianggap tujuannya untuk mempelajari pemetaan dari gambar input hingga ke gambar output.

D. U-Net

U-Net pertama kali diperkenalkan oleh [6] untuk menjawab permasalahan dalam melokalisasi dan membedakan batas pada citra dengan melakukan klasifikasi pada setiap piksel, sehingga input dan output berbagi ukuran yang sama. Arsitektur *U-Net* diilustrasikan pada Figure 1 terdiri dari,

- *Contracting Path* (sisi kiri dari Figure 1) yang terdiri dari lapisan konvolusi yang menurunkan sampel data saat mengekstrak informasi.
- *Expansive path* (sisi kanan dari Figure 1) yang terbuat dari lapisan konvolusi transpos yang meningkatkan sampel informasi.

Contracting Path terdiri dari aplikasi berulang dari dua konvolusi 3×3 (konvolusi tanpa *padding*), masing-masing diikuti oleh fungsi aktivasi *ReLU* dan operasi *max-pooling* 2×2 dengan *stride* 2 untuk *downsampling*. Pada setiap langkah *downsampling*, jumlah fitur digandakan dan setiap langkah di *expansive path* terdiri dari *upsampling* peta fitur diikuti oleh konvolusi 2×2 (“konvolusi-naik”) yang membagi dua jumlah saluran fitur, penggabungan dengan peta fitur yang dipangkas sesuai dari *contracting path*, dan dua 2×2 konvolusi, masing-masing diikuti oleh *ReLU*. Pemangkasannya diperlukan karena hilangnya piksel batas di setiap konvolusi. Pada lapisan terakhir, konvolusi 1×1 digunakan untuk memetakan setiap vektor fitur 64-komponen ke jumlah kelas yang diinginkan. Secara total jaringan memiliki 23 lapisan convolutional. Untuk memungkinkan hasil segmentasi yang mulus dari peta segmentasi output (lihat Figure 2), penting untuk memilih ukuran *kernel* input sedemikian rupa sehingga semua operasi *max-pooling* 2×2 diterapkan ke lapisan dengan ukuran (x, y) yang genap.

E. PatchGAN Discriminator

Pada PatchGAN, alih-alih memprediksi seluruh gambar sebagai palsu atau nyata pada diskriminator, model mengambil

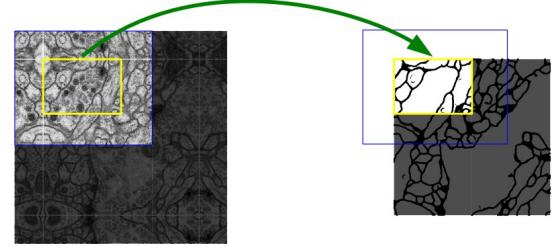


Fig. 2. Hasil Segmentasi Citra menggunakan *U-Net*

$N \times N$ gambar dan memprediksi setiap *pixel* pada *patch* itu sebagai asli atau palsu. PatchGAN memiliki lebih sedikit parameter dan berjalan lebih cepat daripada mengklasifikasikan seluruh gambar. [4]

F. Loss Function

Adapun *loss function* dari *generator network* sebagai berikut

$$\mathcal{L}_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x,z}[\log(1 - D(x, G(x, z)))] \quad (3)$$

Persamaan di atas memiliki dua komponen: satu untuk diskriminator dan yang lainnya untuk generator. Dalam setiap GAN diskriminator dilatih terlebih dahulu di setiap iterasi sehingga dapat mengenali data asli dan palsu sehingga dapat membedakan atau mengklasifikasikan di antara mereka. Di antaranya,

$$D(x, y) = 1 \text{ adalah asli} \quad (4)$$

$$D(x, G(z)) = 0 \text{ adalah palsu} \quad (5)$$

Perlu dicatat bahwa $G(z)$ juga akan menghasilkan sampel palsu dan dengan demikian nilainya akan mendekati nol. Secara teori, diskriminator harus selalu mengklasifikasikan $G(z)$ sebagai nol saja. Jadi pembeda harus menjaga jarak maksimum antara asli dan palsu, dengan kata lain diskriminator harus memaksimalkan *loss function*.

Setelah diskriminator, generator akan di *train*. Generator yaitu $G(z)$ harus belajar untuk menghasilkan sampel yang lebih dekat dengan sampel sebenarnya. Untuk mempelajari distribusi asli dibutuhkan bantuan dari diskriminator yaitu $D(x, G(z)) = 0$, dan diubah menjadi $D(x, G(z)) = 1$.

Dengan perubahan pelabelan, generator sekarang mengoptimalkan parameternya sehubungan dengan parameter milik diskriminator dengan label *ground truth*. Tahap ini memastikan bahwa generator sekarang dapat menghasilkan sampel yang mendekati data nyata yaitu 1.

loss function juga dicampur dengan *L1 loss* sehingga generator tidak hanya menipu diskriminator tetapi juga menghasilkan gambar yang mendekati *ground truth*, yang pada intinya *loss function* memiliki tambahan *L1 loss* untuk generator [1].

$$\mathcal{L}_{L1}(G) = E_{x,y,z}[\|y - G(x, z)\|_1]. \quad (6)$$

dan hasil final dari *loss function* sebagai berikut,

$$G^* = \arg \min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \quad (7)$$

IV. METODE PENELITIAN

A. Dataset

Pada penelitian ini, dataset yang digunakan adalah *dataset maps*. Setiap gambar pada dataset ini berukuran 600x1200 pixel dan merupakan citra RGB. Setiap gambar ini tersusun atas citra *satelit* dan citra satelit dalam bentuk peta yang merepresentasikan citra satelit sebelumnya yang masing-masing berukuran 600x600 pixel. Citra peta akan menjadi *input* bagi model yang dibangun. Hasil dari model ini akan berupa gambar sintesis dari citra *satelit* yang asli.

B. Model

Pada metode ini, terdapat 2 arsitektur yang digunakan yakni *generator* yang menggunakan arsitektur U-Net *discriminator* yang menggunakan *Patch GAN*. U-Net pada *generator* terdiri dari sebuah *encoder* dan *decoder*. Setiap *block* pada *encoder* terdiri dari convolution, *batch normalization* dan menggunakan fungsi aktivasi *Leaky ReLU*. Pada blok *decoder*, terdapat *layer transposed convolution*, *Batch Normalization* dan *Dropout* yang diaplikasikan pada 3 blok pertama. Di antara *encoder* dan *decoder* terdapat *skip connections* seperti pada U-Net.

PatchGAN classifier pada *discriminator* berfungsi untuk mengklasifikasikan setiap *patch* pada gambar apakah asli atau palsu. Setiap blok pada *discriminator* terdiri dari *layer konvolusi*, *batch normalization* dan fungsi aktivasi berupa *Leaky ReLU*. *Discriminator* akan menerima 2 input:

- *input image* dan *target image* yang harus diklasifikasikan sebagai gambar asli
- *input image* dan *generated image* (dihasilkan oleh *generator*) yang harus diklasifikasikan sebagai gambar palsu

V. HASIL DAN PEMBAHASAN

Setelah melakukan pembangunan model dan melakukan *train* terhadap *dataset latih*, maka selanjutnya dilakukan proses pengujian terhadap *dataset uji*. Figure 3 menunjukkan hasil pengaplikasian model pix2pix ke 5 citra peta berbeda. Secara sekilas, model pix2pix yang dibangun mampu menciptakan gambar *aerial* sintetis yang mirip dengan aslinya. Pada *input image* yang pertama, model menghasilkan citra satelit yang sedikit mirip dengan citra aslinya. Namun hasil yang diperoleh menunjukkan citra satelit yang dihasilkan lebih menyerupai citra satelit pada daerah pedesaan dibandingkan dengan citra satelit yang asli yang lebih menyerupai daerah perkotaan. Pada *input image* yang kedua, keempat dan kelima, citra satelit sintetis yang dihasilkan lebih menyerupai citra satelit aslinya. Begitu pula pada *input image* yang ketiga, citra yang dihasilkan sangat mirip dengan citra satelit aslinya.

Implementasi dari model ini juga dilakukan untuk menciptakan citra peta sintetis dari citra satelit. Hasil dari implementasi ini dapat dilihat pada Figure 4. Secara umum hasil yang diperoleh juga cukup mendekati gambar peta yang asli. Apabila dilihat dengan lebih detail hasil yang diperoleh disetiap gambar memiliki gambar jalan yang tidak lurus seperti pada gambar peta yang asli.



Fig. 3. Hasil transformasi gambar peta menjadi citra satelit

VI. KESIMPULAN

Setelah melakukan penelitian terkait model Pix2Pix, dapat disimpulkan bahwa model Pix2Pix dapat memberikan hasil yang baik. Model tersebut mampu menghasilkan citra sintesis yang sangat mirip dengan citra yang asli khususnya dalam implementasinya pada *dataset maps*. Selain itu, model ini juga bersifat universal dalam artian dapat diterapkan dalam berbagai kasus *image to image translation*.

APPENDIX

A. Implementasi Model

Model pix2pix yang dibangun diimplementasikan melalui bahasa pemrograman Python. *Source code* dari model ini tersedia pada tautan <https://github.com/yukiao/Pengantar-Deep-Learning/tree/main/Final%20Term>

Setiap konvolusi yang dilakukan akan menggunakan *filter* berukuran 4x4 dengan *stride* 2. Proses konvolusi pada *encoder* dan *discriminator*, akan mengalami *downsample* dengan faktor 2, sedangkan pada *decoder* akan mengalami *upsample* dengan faktor 2.

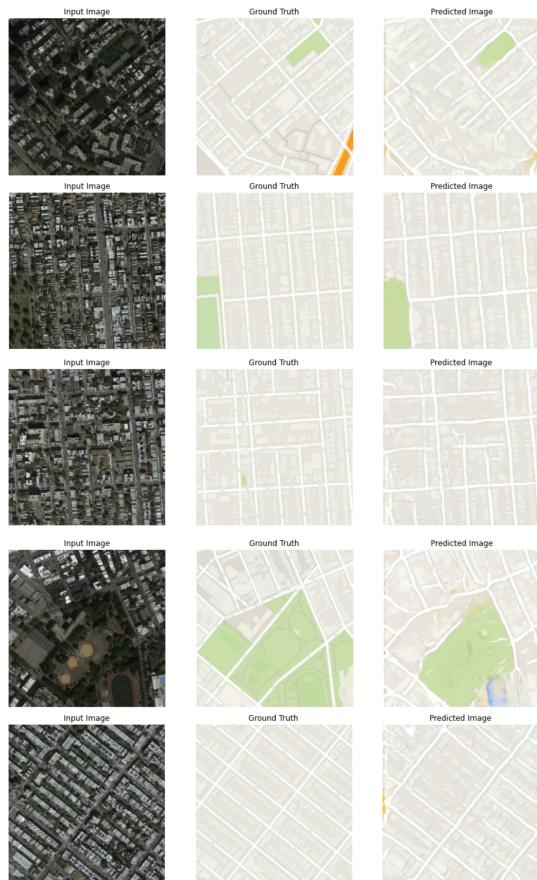


Fig. 4. Hasil transformasi citra satelit menjadi gambar peta

B. Detail Pelatihan

Setiap gambar input akan diresize menjadi ukuran 256x256. Setelah itu akan diaplikasikan *random jitter* yakni mengubah ukuran gambar menjadi 286x286 dan secara acak akan dipotong menjadi berukuran 256x256. Parameter weights diinisialisasikan dengan menggunakan fungsi distribusi Gaussian dengan mean 0 dan standar deviasi 0.02.

Maps ↔ Aerial. Terdapat 1096 gambar pada *dataset* latih yang dapat diunduh dari <http://efrosgans.eecs.berkeley.edu/pix2pix/datasets/maps.tar.gz>. Data dilatih dengan *batch size* 1 dengan *loss function* yang digunakan adalah Adam.

REFERENCES

- [1] Isola, P., Zhu, J. Y., Zhou, T., Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
- [2] Liu, M. Y., Breuel, T., Kautz, J. (2017). Unsupervised image-to-image translation networks. Advances in neural information processing systems, 30.
- [3] Brownlee, J. (2019, July 29). MEMAHAMI PIX2PIX GAN - BLOG. Machine Learning Mastery. Retrieved June 21, 2022, from <https://id.quish.tv/understanding-pix2pix-gan>
- [4] Gopalani, P. (2021, December 27). Understanding Pix2Pix GAN - Artificial Intelligence in Plain English. Medium. Retrieved June 21, 2022, from <https://ai.plainenglish.io/understanding-pix2pix-gan-e21c2bedd213>
- [5] Abbasi, N. (2022, June 20). What is a Conditional GAN (cGAN)? Eduative: Interactive Courses for Software Developers. Retrieved June 21, 2022, from <https://www.educative.io/answers/what-is-a-conditional-gan-cgan>
- [6] Ronneberger, O., Fischer, P., Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
- [7] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.