

CERDAS MENGUASAI GIT

CERDAS MENGUASAI GIT

Dalam 24 Jam

Rolly M. Awangga
Informatics Research Center



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.’
Imam Syafi’i*

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
5 Chapter 5	9
6 Chapter 6	11
7 Chapter 7	13
8 Chapter 8	15
9 Chapter 9	17
10 Chapter 10	27
11 Chapter 11	29
12 Chapter 12	31
13 Chapter 13	33
14 Chapter 14	35

DAFTAR ISI

Foreword	xi
Kata Pengantar	xiii
Acknowledgments	xv
Acronyms	xvii
Glossary	xix
List of Symbols	xxi
Introduction	xxiii
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Chapter 1	1
2 Chapter 2	3
3 Chapter 3	5
4 Chapter 4	7
	ix

5	Chapter 5	9
6	Chapter 6	11
7	Chapter 7	13
8	Chapter 8	15
9	Chapter 9	17
9.1	1174006 - Kadek Diva Krishna Murti	17
9.1.1	Teori	17
9.1.2	Praktek	21
9.1.3	Penanganan Error	26
10	Chapter 10	27
11	Chapter 11	29
12	Chapter 12	31
13	Chapter 13	33
14	Chapter 14	35
Daftar Pustaka		37

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

CHAPTER 1

BAB 2

CHAPTER 2

BAB 3

CHAPTER 3

BAB 4

CHAPTER 4

BAB 5

CHAPTER 5

BAB 6

CHAPTER 6

BAB 7

CHAPTER 7

BAB 8

CHAPTER 8

BAB 9

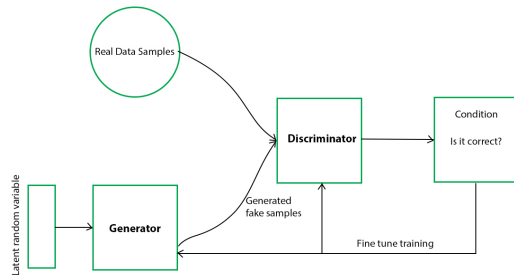
CHAPTER 9

9.1 1174006 - Kadek Diva Krishna Murti

9.1.1 Teori

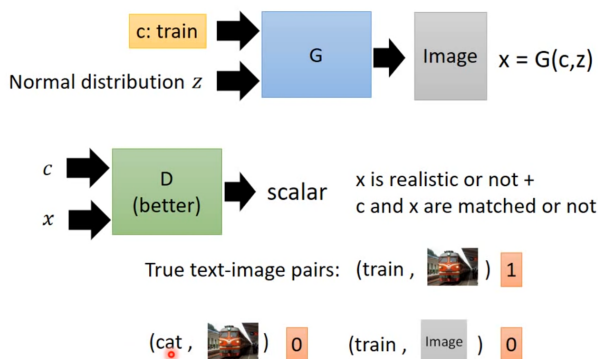
1. Jelaskan dengan ilustrasi gambar sendiri apa perbedaan antara vanilla GAN dan cGAN.

Vanilla GAN merupakan tipe GAN yang paling sederhana. Ia memiliki dua network diantaranya generator network dan discriminator network. Kedua network tersebut akan dilatih dan diadu satu sama lain. Generator network dilatih untuk membodohi discriminator network dengan membuat data baru dari data asli. Discriminator dilatih untuk tidak dibodohi oleh generator network



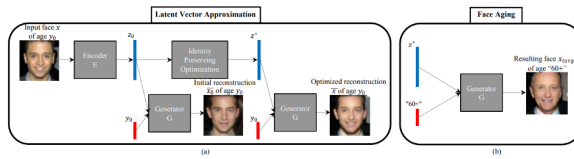
CGAN merupakan tipe GAN yang menerapkan metode deep learning dimana memiliki beberapa parameter kondisional pada generator dan discriminator. Parameter tersebut bisa berupa informasi label untuk membantu membedakan data asli dan data palsu.

Conditional GAN



2. Jelaskan dengan ilustrasi gambar sendiri arsitektur dari Age-cGAN.
Age-cGAN terdiri dari empat network, diantaranya:

- Encoder, network ini memetakan data input ke dalam bentuk latent vector.
- FaceNet, network ini belajar mengenali perbedaan antara data input dengan data hasil generate
- Generator Network, network ini mengenerate data baru dari inputan yang ada.
- Discriminator Network, network ini mencoba membedakan antara data asli dan data palsu.



3. Jelaskan dengan ilustrasi gambar sendiri arsitektur encoder network dari Agec-GAN.

Encoder network memetakan data input ke dalam bentuk latent vector.

4. Jelaskan dengan ilustrasi gambar sendiri arsitektur generator network dari Agec-GAN.

Generator network mengenerate data baru dari inputan yang ada.

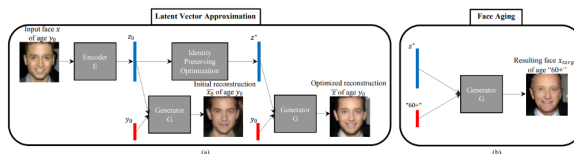
5. Jelaskan dengan ilustrasi gambar sendiri arsitektur discriminator network dari Agec-GAN.

Discriminator network mencoba membedakan antara data asli dan data palsu.

6. Jelaskan dengan ilustrasi gambar apa itu pretrained Inception-ResNet-2 Model. Pretrained Inception-ResNet-2 Model adalah sebuah model terlatih berdasar jutaan gambar dari ImageNet

7. Jelaskan dengan ilustrasi gambar sendiri arsitektur Face recognition network Agec-GAN.

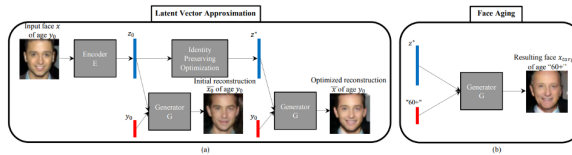
- Encoder, network ini memetakan data input ke dalam bentuk latent vector.
- FaceNet, network ini belajar mengenali perbedaan antara data input dengan data hasil generate
- Generator Network, network ini mengenerate data baru dari inputan yang ada.
- Discriminator Network, network ini mencoba membedakan antara data asli dan data palsu.



8. Sebutkan dan jelaskan serta di sertai contoh-contoh tahapan dari Agec-GAN

- Encoder, network ini memetakan data input ke dalam bentuk latent vector.
- FaceNet, network ini belajar mengenali perbedaan antara data input dengan data hasil generate

- Generator Network, network ini mengenerate data baru dari inputan yang ada.
- Discriminator Network, network ini mencoba membedakan antara data asli dan data palsu.



9. Berikan contoh perhitungan fungsi training objektif

Rumus yang dipakai untuk menerapkan fungsi training objektif adalah sebagai berikut:

$$\min_{\theta_G} \max_{\theta_D} v(\theta_G, \theta_D) = \mathbf{E}_{x, y \sim p_{data}} [\log D(x, y)] \\ + \mathbf{E}_{z \sim p_z(z), \tilde{y} \sim p_y} [\log (1 - D(G(z, \tilde{y}), \tilde{y}))]$$

Penjelasan:

- $\log D(x, y)$ adalah kerugian pada model Diskriminator.
 - $\log (1 - D(G(x, y'), y'))$ adalah kerugian pada model Generator.
 - $P(data)$ adalah distribusi dari semua gambar yang memungkinkan.
10. Berikan contoh dengan ilustrasi penjelasan dari Initial latent vector approximation
- Initial latent vector approximation merupakan metode yang digunakan untuk memperkirakan vektor latent untuk mengoptimalkan rekonstruksi gambar wajah.
 - Encoder adalah jaringan saraf yang mendekati vektor latent.
 - Encoder network dilatih pada gambar yang dihasilkan dan pada gambar asli.
 - Setelah dilatih, jaringan encoder akan mulai menghasilkan vektor laten dari distribusi yang dipelajari.
 - Fungsi objektif pelatihan dari pelatihan jaringan network adalah Euclidean distance loss.
11. Berikan contoh perhitungan latent vector optimization
- Latent vector optimization merupakan proses untuk meningkatkan kinerja dari encoder dan generator network secara sekaligus. Rumus yang dipakai untuk

menerapkan latent vector optimization adalah sebagai berikut:

$$z^*_{IP} = \underset{z}{\operatorname{argmin}} ||FR(x) - FR(\bar{x})||_{L_2}$$

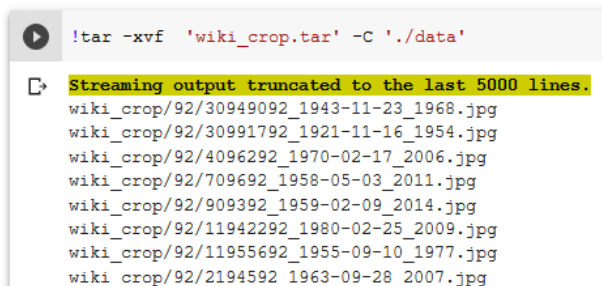
Penjelasan:

- FR adalah face recognition network untuk mengenali identitas orang berdasar inputan gambar wajah
- Persamaan jarak Euclidean antara gambar asli dan gambar rekontruksi
- Meminimalkan jarak Euclidean ini harus meningkatkan pemeliharaan identitas pada rekonstruksi gambar

9.1.2 Praktek

1. Jelaskan bagaimana cara ekstrak file dataset Age-cGAN menggunakan google colab

```
1 !tar -xvf 'wiki_crop.tar' -C './data'
```



```
!tar -xvf 'wiki_crop.tar' -C './data'
```

Streaming output truncated to the last 5000 lines.

```
wiki_crop/92/30949092_1943-11-23_1968.jpg
wiki_crop/92/30991792_1921-11-16_1954.jpg
wiki_crop/92/4096292_1970-02-17_2006.jpg
wiki_crop/92/709692_1958-05-03_2011.jpg
wiki_crop/92/909392_1959-02-09_2014.jpg
wiki_crop/92/11942292_1980-02-25_2009.jpg
wiki_crop/92/11955692_1955-09-10_1977.jpg
wiki_crop/92/2194592_1963-09-28_2007.jpg
```

2. Jelaskan bagaimana kode program bekerja untuk melakukan load terhadap dataset yang sudah di ekstrak, termasuk bagaimana penjelasan kode program perhitungan usia

Cara load dataset, adalah sebagai berikut:

```
1 def load_data(wiki_dir, dataset='wiki'):
2     # Load the wiki.mat file
3     meta = loadmat(os.path.join(wiki_dir, "{}.mat".format
4                               (dataset)))
5
6     # Load the list of all files
7     full_path = meta[dataset][0, 0]["full_path"][0]
8
9     # List of Matlab serial date numbers
```

```

9         dob = meta[dataset][0, 0]["dob"][0]
10
11         # List of years when photo was taken
12         photo_taken = meta[dataset][0, 0]["photo_taken"][0]
13
14         # year
15
16         # Calculate age for all dobs
17         age = [calculate_age(photo_taken[i], dob[i]) for i in
18                 range(len(dob))]
19
20         # Create a list of tuples containing a pair of an
21         image path and age
22         images = []
23         age_list = []
24         for index, image_path in enumerate(full_path):
25             images.append(image_path[0])
26             age_list.append(age[index])
27
28         # Return a list of all images and respective age
29         return images, age_list

```

Penjelasan kode program perhitungan usia, adalah sebagai berikut:

```

1     def calculate_age(taken, dob):
2         birth = datetime.fromordinal(max(int(dob) - 366, 1))
3
4         if birth.month < 7:
5             return taken - birth.year
6         else:
7             return taken - birth.year - 1
8

```

3. Jelaskan bagaimana kode program The Encoder Network bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana

```

1     def build_encoder():
2         """
3         Encoder Network
4         """
5         input_layer = Input(shape=(64, 64, 3))
6
7         # 1st Convolutional Block
8         enc = Conv2D(filters=32, kernel_size=5, strides=2,
9           padding='same')(input_layer)
10        # enc = BatchNormalization()(enc)
11        enc = LeakyReLU(alpha=0.2)(enc)
12
13        # 2nd Convolutional Block
14        enc = Conv2D(filters=64, kernel_size=5, strides=2,
15           padding='same')(enc)
16        enc = BatchNormalization()(enc)
17        enc = LeakyReLU(alpha=0.2)(enc)
18
19        # 3rd Convolutional Block
20        enc = Conv2D(filters=128, kernel_size=5, strides=2,
21           padding='same')(enc)

```

```

19     enc = BatchNormalization()(enc)
20     enc = LeakyReLU(alpha=0.2)(enc)
21
22     # 4th Convolutional Block
23     enc = Conv2D(filters=256, kernel_size=5, strides=2,
padding='same')(enc)
24     enc = BatchNormalization()(enc)
25     enc = LeakyReLU(alpha=0.2)(enc)
26
27     # Flatten layer
28     enc = Flatten()(enc)
29
30     # 1st Fully Connected Layer
31     enc = Dense(4096)(enc)
32     enc = BatchNormalization()(enc)
33     enc = LeakyReLU(alpha=0.2)(enc)
34
35     # Second Fully Connected Layer
36     enc = Dense(100)(enc)
37
38     # Create a model
39     model = Model(inputs=[input_layer], outputs=[enc])
40     return model
41

```

Encoder, network ini memetakan data input ke dalam bentuk latent vector.

4. Jelaskan bagaimana kode program The Generator Network bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana

```

1     def build_generator():
2         """
3         Create a Generator Model with hyperparameters values
4         defined as follows
5         """
6         latent_dims = 100
7         num_classes = 6
8
9         input_z_noise = Input(shape=(latent_dims,))
10        input_label = Input(shape=(num_classes,))
11
12        x = concatenate([input_z_noise, input_label])
13
14        x = Dense(2048, input_dim=latent_dims + num_classes)(
x)
15        x = LeakyReLU(alpha=0.2)(x)
16        x = Dropout(0.2)(x)
17
18        x = Dense(256 * 8 * 8)(x)
19        x = BatchNormalization()(x)
20        x = LeakyReLU(alpha=0.2)(x)
21        x = Dropout(0.2)(x)
22
23        x = Reshape((8, 8, 256))(x)
24
25        x = UpSampling2D(size=(2, 2))(x)

```



```

25         x = Conv2D(filters=128, kernel_size=5, padding='same',
26         )(x)
27         x = BatchNormalization(momentum=0.8)(x)
28         x = LeakyReLU(alpha=0.2)(x)
29
30         x = UpSampling2D(size=(2, 2))(x)
31         x = Conv2D(filters=64, kernel_size=5, padding='same')(
32         x)
33         x = BatchNormalization(momentum=0.8)(x)
34         x = LeakyReLU(alpha=0.2)(x)
35
36         x = UpSampling2D(size=(2, 2))(x)
37         x = Conv2D(filters=3, kernel_size=5, padding='same')(
38         x)
39         x = Activation('tanh')(x)
40
41         model = Model(inputs=[input_z_noise, input_label],
42         outputs=[x])
43         return model

```

Generator Network, network ini mengenerate data baru dari inputan yang ada.

5. Jelaskan bagaimana kode program The Discriminator Network bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana

```

1     def build_discriminator():
2         """
3         Create a Discriminator Model with hyperparameters
4         values defined as follows
5         """
6         input_shape = (64, 64, 3)
7         label_shape = (6,)
8         image_input = Input(shape=input_shape)
9         label_input = Input(shape=label_shape)
10
11         x = Conv2D(64, kernel_size=3, strides=2, padding='
12         same')(image_input)
13         x = LeakyReLU(alpha=0.2)(x)
14
15         label_input1 = Lambda(expand_label_input)(label_input
16         )
17         x = concatenate([x, label_input1], axis=3)
18
19         x = Conv2D(128, kernel_size=3, strides=2, padding='
20         same')(x)
21         x = BatchNormalization()(x)
22         x = LeakyReLU(alpha=0.2)(x)
23
24         x = Conv2D(256, kernel_size=3, strides=2, padding='
25         same')(x)
26         x = BatchNormalization()(x)
27         x = LeakyReLU(alpha=0.2)(x)
28
29         x = Conv2D(512, kernel_size=3, strides=2, padding='
30         same')(x)

```

```

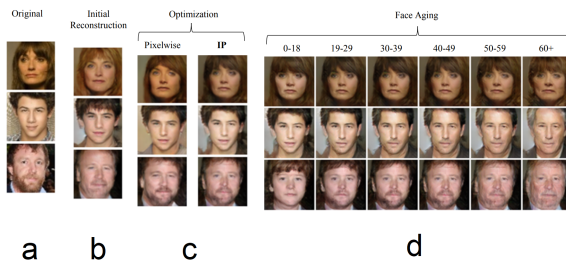
25     x = BatchNormalization()(x)
26     x = LeakyReLU(alpha=0.2)(x)
27
28     x = Flatten()(x)
29     x = Dense(1, activation='sigmoid')(x)
30
31     model = Model(inputs=[image_input, label_input],
32                   outputs=[x])
33     return model

```

Discriminator Network, network ini mencoba membedakan antara data asli dan data palsu.

6. Jelaskan bagaimana kode program Training cGAN bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana Cara kerja program Training cGAN, adalah sebagai berikut:

- Gambar asli yang akan diproses.
- Rekonstruksi gambar yang telah digenerate menggunakan initial latent approximations z_0 .
- Rekonstruksi gambar yang telah digenerate menggunakan "Pixelwise" and "Identity-Preserving" lalu dilakukan optimasi menggunakan latent approximations: $z * \text{pixel}$ and $z * \text{IP}$.
- penentuan umur berdasar gambar generate yang telah direkonstruksi menggunakan identity-preserving $z * \text{IP}$ latent approximations and kondisi dari berbagai label umur y (one per column).



7. Jelaskan bagaimana kode program Initial dan latent vector approximation bekerja dijelaskan dengan bahasa awam dengan ilustrasi sederhana

- Initial latent vector approximation merupakan metode yang digunakan untuk memperkirakan vektor latent untuk mengoptimalkan rekonstruksi gambar wajah.
- Encoder adalah jaringan saraf yang mendekati vektor latent.
- Encoder network dilatih pada gambar yang dihasilkan dan pada gambar asli.

- Setelah dilatih, jaringan encoder akan mulai menghasilkan vektor laten dari distribusi yang dipelajari.
- Fungsi objektif pelatihan dari pelatihan jaringan network adalah Euclidean distance loss.

9.1.3 Penanganan Error

1. Skrinsut error
2. Tuliskan kode eror dan jenis errornya
3. Solusi pemecahan masalah error tersebut

BAB 10

CHAPTER 10

BAB 11

CHAPTER 11

BAB 12

CHAPTER 12

BAB 13

CHAPTER 13

BAB 14

CHAPTER 14

DAFTAR PUSTAKA

- [1] R. Awangga, "Sampeu: Servicing web map tile service over web map service to increase computation performance," in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.

