

# Implementasi Augmentasi Citra pada Suatu *Dataset*

Muhammad Fadli Gunardi (13519130)  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): 13519130@std.stei.itb.ac.id

**Abstrak**—Pada saat ini penggunaan CNN sering kali digunakan dalam mengenali suatu objek. Namun, untuk membuat suatu model CNN yang memiliki nilai akurasi yang tinggi tidaklah mudah. Berbagai macam faktor dapat mempengaruhi kinerja dari model CNN tersebut, diantaranya adalah arsitektur CNN yang digunakan, hyperparameter pada training data, jumlah data training, hingga kualitas citra dalam suatu dataset. Makalah ini akan membahas solusi dalam mengatasi permasalahan kurangnya jumlah citra pada suatu dataset dengan melakukan metode augmentasi citra dengan pendekatan proses *filtering* dan transformasi geometri

**kata kunci**—*augmentasi; dataset; CNN;*

## I. PENDAHULUAN

Pada saat ini, teknologi *computer vision* sudah banyak dikembangkan dalam berbagai bidang, contohnya pada bidang otomotif, pengenalan wajah, kesehatan, manufaktur, pertambangan, dll. Misal, pada bidang otomotif *computer vision* dapat diterapkan dalam pendeteksian jarak mobil depan dan pendeteksian batas lajur yang berguna untuk mendukung penerapan kendaraan *semi autonomous*. Selanjutnya, pada bidang pengenalan wajah *computer vision* dapat diterapkan untuk mengetahui jenis kelamin seseorang, usia, nama, dll. Dengan demikian, *computer vision* sangat membantu dan meringankan beban pekerjaan manusia pada kehidupan sehari-hari di berbagai bidang. Oleh karena itu, dengan keunggulan yang ditawarkan teknologi *computer vision* sangatlah menarik untuk dikembangkan.

Dalam pengaplikasian *computer vision* pada berbagai bidang kerap kali membutuhkan suatu proses pembelajaran berupa deep learning yang berguna untuk mendapatkan hasil yang sebanding dengan manusia atau bahkan melampauinya. Salah satu model *deep learning* yang cocok untuk menyelesaikan masalah berhubungan dengan citra adalah CNN. CNN atau *Convolutional Neural Network* adalah salah satu kelas *neural network* yang menerapkan variasi *perceptron multilayer* yang berisi satu atau lebih lapisan konvolusi yang dapat terhubung seluruhnya atau digabungkan. Hal tersebut memungkinkan CNN untuk dapat mempelajari filter atau kernel secara otomatis tanpa membuatnya secara eksplisit. Filter atau kernel ini dapat membantu mengekstraksi fitur yang tepat dan relevan dari data *input*. Dengan demikian, CNN sangat cocok dalam tugas-tugas pengenalan objek

Namun, kinerja dari CNN itu tersendiri sangat bergantung dengan jumlah data *training*. Hal ini dibuktikan dengan CNN yang memiliki kecenderungan *overfitting* apabila jumlah

training data pada model tergolong sedikit. *Overfitting* ini terjadi karena CNN menghafal fitur detail dari data *training* yang tidak dapat digeneralisasi. Sebaliknya, dengan menambahkan jumlah data *training* pada suatu model CNN dapat meningkatkan akurasi dari model tersebut.

Untuk meningkatkan nilai akurasi pada suatu model dibutuhkan data *training* yang berjumlah banyak. Dalam *deep learning* kumpulan data *training* tergabung pada suatu *dataset*. Dengan demikian, dibutuhkan suatu dataset yang memiliki ratusan ribu atau bahkan jutaan citra. Akan tetapi, untuk mendapatkan *dataset* citra yang memiliki jumlah data yang banyak tidak mudah, mayoritas berbayar. Oleh karena itu, diperlukan suatu metode yang dapat memperbanyak citra dalam pada suatu *dataset* tanpa menghilangkan karakteristik dari citra tersebut.

Metode augmentasi adalah metode yang paling tepat untuk menyelesaikan masalah tersebut karena dapat menciptakan citra artifisial melalui berbagai cara pemrosesan atau kombinasi beberapa pemrosesan, seperti rotasi acak, pergeseran, peregangan kontras, penajaman, dll tanpa mengubah label dari citra tersebut. Berdasarkan hasil eksperimen yang dilakukan (Daniel, 2018) proses augmentasi terutama melalui metode *filtering*, seperti penajaman, penambahan blur, penambahan *noise* dapat meningkatkan akurasi model CNN dari 54,9% menjadi 99,6% dari 2.801 *test data*. Peningkatan nilai akurasi tersebut terjadi karena model CNN dapat mengenali lebih banyak objek dari beragam kondisi yang terjadi pada citra tersebut. Misal, sebelum dilakukan proses augmentasi suatu model CNN tidak dapat mengklasifikasikan citra yang mengalami blur, memiliki *noise*, atau kurang kontras, maka setelah proses augmentasi suatu model CNN dapat mengklasifikasikan citra tersebut.

Dengan demikian, pada makalah ini akan diimplementasikan proses augmentasi pada suatu *dataset* melalui pendekatan proses *filtering* dan transformasi geometri untuk dapat menyelesaikan masalah keterbatasan data pada suatu *dataset*. Dengan menerapkan proses augmentasi ini diharapkan dapat meningkatkan nilai akurasi dari model CNN yang selanjutnya dapat digunakan untuk mengenali suatu objek.

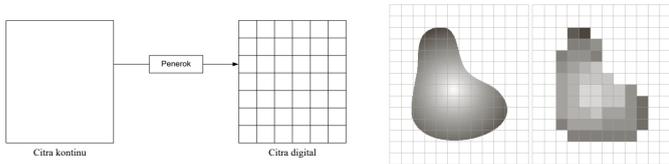
## II. LANDASAN TEORI

### A. Citra digital

Secara matematis citra adalah fungsi intensitas cahaya pada bidang dwimatra disimbolkan dengan  $f(x, y)$ , yang dalam hal

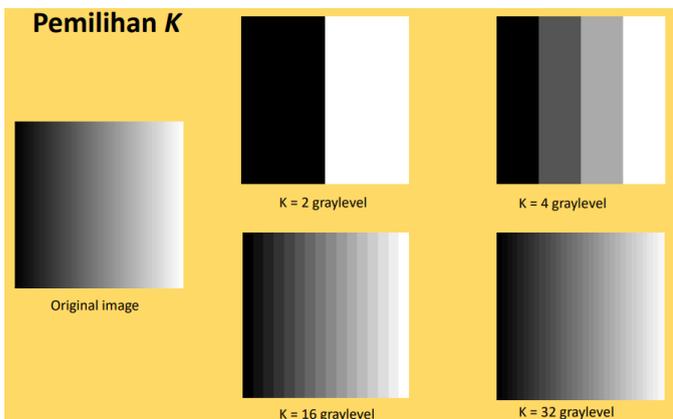
ini  $(x, y)$  adalah koordinat pada bidang dwimatra dan  $f(x, y)$  adalah intensitas cahaya (*brightness*) pada titik  $(x, y)$ . Nilai intensitas cahaya dari suatu citra merupakan hasil perkalian dari jumlah cahaya yang berasal dari sumbernya (*illumination*) dengan derajat kemampuan objek memantulkan cahaya (*reflection*) yang nilainya antara 0 dan 1. Intensitas cahaya bernilai antara 0 sampai tidak berhingga.

Dalam pemrosesan digital, citra akan diubah ke dalam bentuk citra digital melalui proses penerokan atau *sampling* yang membuat domain citra menjadi digital per sumbu  $x$  dan sumbu  $y$  (dwimatra) dan kuantisasi yang memberikan *integer* untuk nilai intensitas yang dicerminkan pada koordinat terkait. Penerokan bertujuan untuk menentukan seberapa banyak pixel yang diperlukan untuk merepresentasikan citra kontinu, dan bagaimana pengaturannya. Pada gambar dibawah adalah ilustrasi proses penerokan dimana citra kontinu akan diterok menjadi *grid-grid* yang berbentuk bujursangkar. Jumlah terokan biasanya diasumsikan perpangkatan dari dua. Misal  $256 \times 256$ ,  $128 \times 256$ ,  $512 \times 1024$ , dll.



Gambar II.1 Ilustrasi penerokan

Berbeda dengan metode penerokan, kuantisasi adalah metode yang berkaitan dengan diskritisasi nilai intensitas cahaya pada koordinat  $(x, y)$ . Tujuan proses kuantisasi ini adalah memetakan nilai dari sinyal kontinyu menjadi  $K$  buah nilai diskrit ( $K$  buah level). Kuantisasi dapat dilakukan dengan beberapa teknik, diantaranya adalah *uniform mapping*, *logarithmic mapping*, dll. Dalam teknik *uniform mapping* kuantisasi yang membagi citra ke dalam rentang nilai keabuan  $[0, M]$  menjadi  $K$  buah upa-selang yang berjarak sama. Kemudian masing-masing koordinat pixel diisi nilai integer dalam selang  $[0, K-1]$ . Sedangkan *Logarithmic Mapping* yang menyatakan sinyal dalam fungsi logaritma. Berikut merupakan ilustrasi dari proses kuantisasi dengan pemilihan nilai  $K$  bernilai 2, 4, 16, dst.



Gambar II.2 Ilustrasi Kuantisasi

## B. Perbaikan Kualitas citra

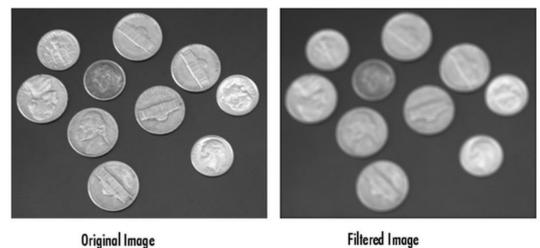
*Image Enhancement* atau perbaikan kualitas biasa dilakukan pada citra sebelum dilakukan operasi-operasi lebih lanjut seperti operasi pengenalan objek. Hal tersebut dilakukan karena citra input biasanya memiliki gangguan berupa blur, mengandung derau (*noise*), kontras terlalu rendah, terlalu terang atau terlalu gelap, dll. Proses perbaikan kualitas ini dapat dilakukan di ranah spasial atau ranah frekuensi sesuai dengan jenis gangguan yang dialami citra. Pada makalah ini akan dibahas perbaikan beberapa metode perbaikan citra yang berhubungan dengan proses augmentasi, diantaranya adalah *image smoothing*, *image sharpening*, *contrast stretching*, dan *image brightening*.

### 1. Image Smoothing

*Image smoothing* atau pelembutan citra dapat dilakukan untuk menghasilkan efek blur dan untuk mengurangi derau (*noise*) pada citra. Pelembutan citra dilakukan untuk menekan komponen yang berfrekuensi tinggi dan meloloskan komponen yang berfrekuensi rendah. Terdapat berbagai macam metode yang dapat digunakan dalam pelembutan citra, di antara lain adalah *mean filter*, *gaussian filter*, atau dengan non-linier filter, seperti *minimum filter*, *maximum filter* ataupun median filter.

Metode *mean filter* digunakan dengan cara menandai nilai masing-masing elemen matriks sama, yaitu  $1/n$  dengan  $n$  adalah ukuran *mask*. *Mean filter* dapat digunakan untuk menghilangkan derau pada citra. Sedangkan metode *gaussian filter* bekerja dengan melakukan operasi konvolusi citra input dengan penapis *gaussian* dengan aturan sebagai berikut.

- Jumlah semua koefisien penapis harus bernilai 1.
- Setiap koefisien penapis harus lebih besar dari 0 (positif).



Gambar II.3 perbandingan citra gambar sebelum dan sesudah dilakukan pelembutan citra dengan penapis rata-rata

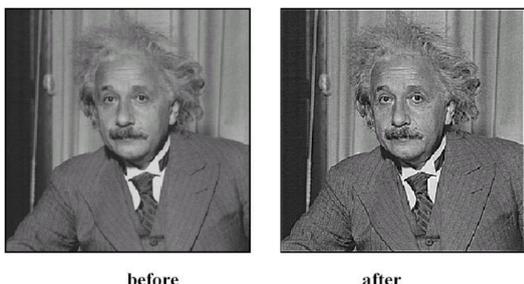
### 2. Image Sharpening

*Image sharpening* atau penajaman citra adalah proses proses peningkatan intensitas dari bagian tepi dari suatu citra. Operasi ini dilakukan dengan menggunakan penapis lolos tinggi atau *high-pass filter*. Penapis lolos-tinggi akan meloloskan (sekalius memperkuat) komponen yang berfrekuensi tinggi (misalnya tepi atau pinggiran objek) dan akan

menurunkan komponen berfrekuensi rendah. Akibatnya, pinggiran objek terlihat lebih tajam dibandingkan sekitarnya. Penapis lolos tinggi (*high-pass filter*) digunakan dengan aturan berikut.

- Bobot di dalam penapis boleh positif, negatif, atau nol
- Jumlah semua bobot adalah 0 atau 1

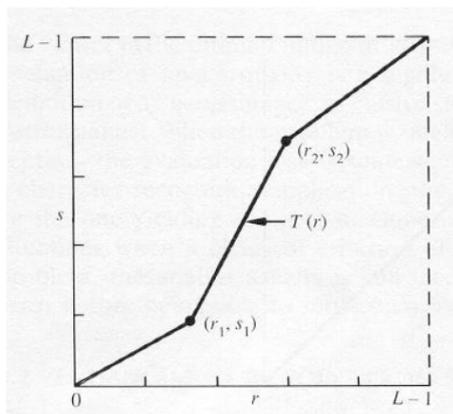
Berikut merupakan contoh perbandingan dari citra yang telah dilakukan penajaman.



Gambar II.4 perbandingan citra gambar sebelum dan sesudah dilakukan penajaman citra

### 3. Contrast Stretching

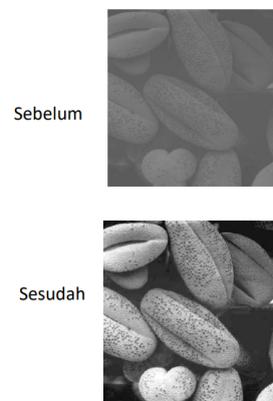
*Contrast Stretching* atau peregangan kontras adalah suatu metode sederhana untuk memperbaiki citra yang memiliki kontras rendah. Tujuan dari peregangan kontras adalah untuk meningkatkan rentang nilai-nilai keabuan untuk citra kontras-rendah (terentang dari nilai  $r_1$  sampai  $r_2$  pada citra dengan nilai keabuan 0 sampai  $L - 1$ ).



Gambar II.5 Ilustrasi peregangan kontras

Berdasarkan gambar di atas nilai-nilai pixel antara  $r_1$  sampai  $r_2$  akan dipetakan menjadi nilai antara  $s_1$  sampai  $s$ . Salah satu pendekatan dalam melakukan peregangan kontras adalah dengan memindai histogram citra, atau pindai pixel-pixel di dalam citra, lalu cari pixel bernilai *minimum*, misalkan  $r_{min}$ . Selanjutnya, cari pixel bernilai *maksimum*, misalkan  $r_{max}$  - pixel-pixel di bawah  $r_{min}$  diset 0, lalu pixel-pixel di atas  $r_{max}$  di-set  $L - 1 - r_1 = r_{min}$ ,  $r_2 = r_{max}$ . Selanjutnya, tentukan persamaan garis yang menghubungkan titik  $(r_{min}, 0)$  dan  $(r_{max}, L - 1)$

dengan persamaan umum garis:  $(y - y_1)/(y_2 - y_1) = (x - x_1)/(x_2 - x_1)$ . Terakhir, petakan nilai keabuan yang lain di antara  $(r_{min}, 0)$  dan  $(r_{max}, L - 1)$  dengan menggunakan persamaan tersebut. Berikut merupakan contoh perbandingan dari citra yang telah dilakukan peregangan kontras.



Gambar II.6 perbandingan citra sebelum dan sesudah dilakukan peregangan kontras

### 4. Image Brightening

*Image brightening* atau pencerahan citra dilakukan untuk menerangkan piksel-piksel yang gelap yang menyebabkan suatu citra tampak gelap. Proses ini bekerja dengan menambahkan/mengurangkan sebuah konstanta kepada (atau dari) setiap pixel, atau mengalikan sebuah konstanta ke setiap pixel. Berikut adalah rumus dari pencerahan citra:

$$s = r + b$$

$r$  adalah nilai kecerahan piksel pada saat ini dan  $b$  adalah input dari user yang dapat bernilai positif atau negatif. Jika  $b$  positif, maka citra akan bertambah terang, sedangkan jika  $b$  bernilai negatif, maka akan semakin gelap. Operasi *clipping* dilakukan jika nilai  $r + b$  berada di bawah nilai intensitas minimum atau di atas nilai intensitas maksimum. Misal, apabila  $r + b > 255$ , maka  $s$  akan bernilai 255, sedangkan apabila  $r + b < 0$ , maka  $s$  akan bernilai 0. Rumus lain dalam melakukan *image brightening* adalah mengalikan nilai piksel saat ini dengan sebuah konstanta dengan rumus sebagai berikut.

$$s = ar + b$$

Berikut merupakan contoh perbandingan dari citra yang telah dilakukan operasi pencerahan.



Gambar II.7 perbandingan citra sebelum dan sesudah dilakukan penerangan citra

### C. Transformasi Citra

Transformasi citra merupakan operasi yang diterapkan pada suatu citra dengan tujuan untuk menggeser, merotasi, melakukan perbesaran atau pengecilan skala pada sebuah citra. Hal tersebut dilakukan untuk melakukan penambahan citra pada suatu dataset dengan cara menerapkan augmentasi geometri. Pada makalah ini akan dibahas 3 jenis transformasi, yaitu *translation*, *rotation*, dan *scaling*, berikut merupakan penjelasan secara detailnya.

#### 1. Translation

*Translation* atau translasi adalah operasi transformasi pada citra yang berguna untuk menggeser citra pada sumbu x atau y. Berikut adalah fungsi dari translasi pada citra.

$$\begin{cases} x = u + t_x \\ y = v + t_y \end{cases} \quad \text{and} \quad \begin{cases} u = x - t_x \\ v = y - t_y \end{cases}$$

Gambar II.8 Fungsi translasi

#### 2. Rotation

*Rotation* atau rotasi adalah operasi transformasi pada citra yang berguna untuk menggeser citra pada suatu sudut tertentu. Berikut adalah fungsi dari rotasi pada citra.

$$\begin{cases} x = u \cos \theta - v \sin \theta \\ y = u \sin \theta + v \cos \theta \end{cases} \quad \text{and} \quad \begin{cases} u = x \cos \theta + y \sin \theta \\ v = -x \sin \theta + y \cos \theta \end{cases}$$

Gambar II.9 Fungsi rotasi

#### 3. Scaling

*Scaling* atau penskalaan adalah operasi transformasi pada citra yang berguna untuk memperbesar atau memperkecil tampilan dari suatu citra. Berikut adalah fungsi dari Scaling pada citra.

$$\begin{cases} x = s_x u \\ y = s_y v \end{cases} \quad \text{and} \quad \begin{cases} u = x / s_x \\ v = y / s_y \end{cases}$$

Gambar II.10 Fungsi *scaling*

## III. IMPLEMENTASI SOLUSI

Implementasi augmentasi akan diimplementasikan dengan Python. Proses augmentasi yang dilakukan berupa metode *filtering* dan augmentasi geometri. Untuk proses augmentasi *filtering* akan dilakukan metode *random image brightening* dan *random contrast stretching*, sedangkan untuk proses augmentasi geometri akan dilakukan proses *rotation* sebesar 180 derajat terhadap sumbu x dan y. Pemilihan metode augmentasi ini didasarkan oleh kasus-kasus yang biasa terjadi di *dataset*, yaitu kurang terangnya citra, kurang kontrasnya citra, atau citra yang terbalik.

Proses augmentasi dimulai dengan melakukan pembagian citra pada dataset ke data *training*, data *testing*. Selanjutnya akan dilanjutkan dengan pembacaan citra dengan melakukan iterasi pada *dataset flower*. Berikut gambaran dari algoritma tersebut.

```
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
import tensorflow_datasets as tfds

(train_ds, val_ds, test_ds), metadata = tfds.load(
    'tf_flowers',
    split=['train[:80%]', 'train[80%:90%]', 'train[90%:]'],
    with_info=True,
    as_supervised=True,
)

image, label = next(iter(train_ds))
```

Gambar III.1 Proses pemisahan *training* data dengan *testing* data pada *dataset*

Selanjutnya untuk menerapkan proses augmentasi digunakan *class image* pada tensorflow untuk mempermudah algoritma yang digunakan. Pada kelas *image* tersebut terdapat beberapa fungsi yang dapat melakukan metode *image enhancement*, seperti *image brightening*, *contrast stretching*, atau *increase saturation* sehingga dapat menyederhanakan proses yang dibutuhkan. Setelah proses augmentasi dilakukan, dilanjutkan dengan proses menampilkan hasil citra yang telah di augmentasi menggunakan metode subplot pada koordinat yang telah ditentukan. Berikut merupakan gambaran dari algoritma tersebut.

```
[35] fig = plt.figure()
      fig.suptitle('Augmented image with random brightness', fontsize=20)

      for i in range(9):
          seed = (i, 0)
          stateless_random_brightness = tf.image.stateless_random_brightness(
              image, max_delta=0.5, seed=seed)
          plt.subplot(3, 3, i + 1)
          plt.imshow(stateless_random_brightness)
          plt.axis('off')
```

Gambar III.2 Proses augmentasi pada *dataset*

Dalam mengimplementasi *random contrast* dilakukan pemanggilan fungsi *stateless\_random\_contrast* dengan parameter berupa *lower* dan *upper* yang berguna untuk menentukan batas bawah dan atas pada faktor kontras yang akan dipanggil secara random. Berikut merupakan gambaran dari fungsi tersebut.

```
tf.image.stateless_random_contrast(image, lower=0.5, upper=2.4, seed=seed)
```

Gambar III.3 Fungsi *random* kontras

Selanjutnya, untuk *random saturation* dilakukan pemanggilan fungsi *stateless\_random\_saturation* dengan parameter berupa batas bawah dan atas pada faktor *saturation* yang akan dipanggil secara *random*. Berikut merupakan gambaran dari fungsi tersebut.

```
tf.image.stateless_random_saturation(image, 0.5, 1, seed)
```

Gambar III.4 Fungsi *random saturation*

Dalam mengimplementasi *random flip* dilakukan pemanggilan fungsi *stateless\_random\_flip\_left\_right* sebagai berikut.

```
tf.image.stateless_random_flip_left_right(image, seed)
```

Gambar III.5 Fungsi *random flip*

Terakhir, proses yang dilakukan setelah melakukan augmentasi data pada *dataset* adalah menggabungkannya kembali dengan *training data* pada *dataset* sebelumnya dengan mempertahankan labelnya berdasarkan *unique value* citra tersebut.

```
def augment(image_label, seed):
    image, label = image_label
    # Random brightness.
    image = tf.image.stateless_random_brightness(image, max_delta=0.5, seed=seed)
    # Random contrast.
    image = tf.image.stateless_random_contrast(image, lower=0.5, upper=2.4, seed=seed)
    # Random saturation.
    image = tf.image.stateless_random_saturation(image, 0.5, 1, seed)
    # Random flip.
    image = tf.image.stateless_random_flip_left_right(image, seed=seed)
    image = tf.image.stateless_random_flip_up_down(image, seed=seed)
    return image, label

# Retain each image with unique label.
counter = tf.data.experimental.Counter()
train_ds = tf.data.Dataset.zip((train_datasets, (counter, counter)))

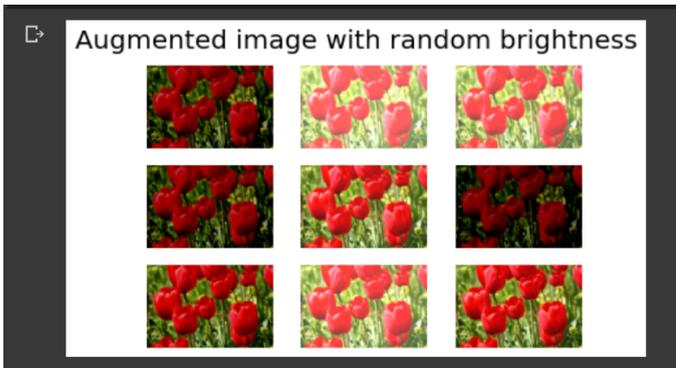
train_ds = (
    train_ds
    .shuffle(1000)
    .map(augment, num_parallel_calls=AUTOTUNE)
    .batch(batch_size)
    .prefetch(AUTOTUNE)
)

print(len(train_ds))
```

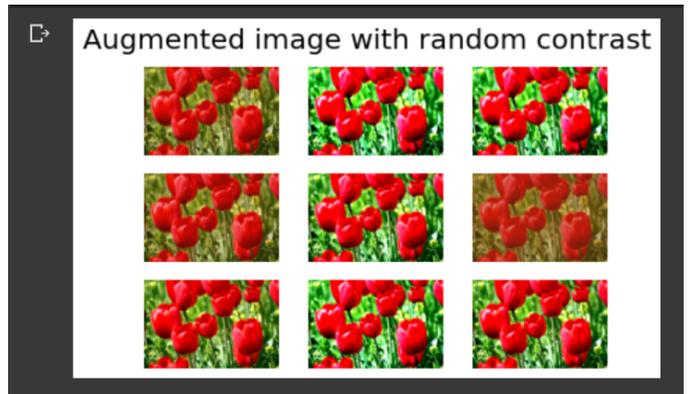
Gambar III.6 Integrasi hasil augmentasi ke dalam *dataset*

#### IV. HASIL EKSPERIMEN

Berikut merupakan hasil dari solusi yang telah diimplementasikan pada bab sebelumnya.



Gambar IV.1 Hasil augmentasi citra berdasarkan *random brightness*



Gambar IV.2 Hasil augmentasi citra berdasarkan *random contrast*



Gambar IV.3 Hasil augmentasi citra berdasarkan *random saturation*



Gambar IV.4 Hasil augmentasi citra berdasarkan *random flip* pada sumbu y

Berdasarkan hasil eksperimen yang telah dilakukan didapatkan bahwa telah berhasil memperbanyak suatu citra. Akan tetapi hanya terbatas pada metode yang random sehingga perlu pengembangan metode augmentasi ini untuk dapat mencakup parameter yang tidak random.

#### KESIMPULAN

Proses augmentasi citra pada suatu *dataset* dapat dilakukan dengan berbagai macam metode, diantaranya adalah *filtering* dan transformasi geometri. Dalam pengimplementasiannya masih banyak metode dari augmentasi

*filtering* yang belum dicakup pada makalah ini, seperti *image smoothing* dan *image brightening*. Terlepas dari hal tersebut, proses augmentasi yang telah diterapkan berhasil memperbanyak jumlah citra pada suatu *dataset* dengan mempertahankan label citranya.

Namun, proses augmentasi berupa ini perlu diteliti lagi dampaknya pada suatu model CNN. Apakah dengan menambahkan citra ini dapat meningkatkan nilai akurasi dari suatu model CNN atau tidak. Lalu, apabila didapatkan bahwa proses augmentasi dapat meningkatkan nilai akurasi dari suatu model CNN, diperlukan penelitian lebih lanjut berkaitan dengan metode-metode augmentasi mana yang dapat meningkatkan nilai akurasi dari suatu model CNN.

LINK YOUTUBE : <https://youtu.be/W30KiGXrpo4>

#### UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Allah SWT dengan izinnya penulis dapat menuntaskan tugas makalah ini. Ucapan terima kasih juga ingin disampaikan oleh penulis kepada dosen mata kuliah Interpretasi dan Pengolahan Citra, IF 4073 terutama kepada Bapak Rinaldi Munir selaku dosen pengajar, atas ilmunya yang telah disampaikan dapat berguna bagi penulis untuk menerapkannya dalam makalah ini.

#### REFERENSI

- [1] Daniel, V. (2018). Pengenalan Wajah Menggunakan Convolutional Neural Network. Tesis, Institut Teknologi Sepuluh Noverber, Surabaya: Departemen Informatika.
- [2] Munir, R. 2022. Pembentukan Citra dan Digitalisasi Citra. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/03-Pem>

[bentukan-Citra-dan-Digitalisasi-Citra-2022.pdf](#). diakses pada 19 Desember.

- [3] Munir, R. 2022. Image Enhancement Bagian 1. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/08-Image-Enhancement-Bagian1-2022.pdf>. diakses pada 19 Desember.
- [4] Munir, R. 2022. Image Enhancement Bagian 3. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/10-Image-Enhancement-Bagian3-2022.pdf>. diakses pada 19 Desember.
- [5] Munir, R. 2022. Image Warping dan Image Morphing. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/27-Image-Warping-Morphing-2021.pdf>. diakses pada 19 Desember.

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2022



Muhammad Fadli Gunardi  
13519130