

Nodeflux Internship Project

Bayu Aditya

Artificial Intelligence Engineer

2019

# Kata Pengantar

Assalamu’alaikum warahmatullahi wabarakatuh

Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga penulis dapat menyelesaikan makalah ini. Tanpa pertolongan-Nya tentunya penulis tidak akan sanggup untuk menyelesaikan makalah ini dengan baik. Shalawat serta salam semoga terlimpah curahkan kepada baginda tercinta kita yaitu Nabi Muhammad SAW yang kita nanti-natikan syafa’atnya di akhirat nanti.

Penulis mengucapkan syukur kepada Allah SWT atas limpahan nikmat sehat-Nya, baik itu berupa sehat fisik maupun akal pikiran, sehingga penulis mampu untuk menyelesaikan pembuatan makalah sebagai tugas akhir dari program magang di Nodeflux sebagai *AI Engineer*.

Penulis tentu menyadari bahwa makalah ini masih jauh dari kata sempurna dan masih banyak terdapat kesalahan serta kekurangan di dalamnya. Untuk itu, penulis mengharapkan kritik serta saran dari pembaca untuk makalah ini, supaya makalah ini nantinya dapat menjadi makalah yang lebih baik lagi. Demikian, dan apabila terdapat banyak kesalahan pada makalah ini penulis mohon maaf yang sebesar-besarnya.

Penulis juga mengucapkan terima kasih kepada semua pihak khususnya kepada team *Artificial Intellegence* Nodeflux yang telah membimbing penulis dalam pembuatan makalah ini.

Demikian, semoga makalah ini dapat bermanfaat. Terima kasih.

Jakarta, Maret 2019

Penyusun

# Daftar isi

[Kata Pengantar 2](#_Toc4064155)

[Daftar isi 3](#_Toc4064156)

[Pendahuluan 5](#_Toc4064157)

[1.1. Latar Belakang 5](#_Toc4064158)

[1.2. Tujuan 6](#_Toc4064159)

[Landasan Teori 7](#_Toc4064160)

[2.1. Imbalance Dataset 7](#_Toc4064162)

[2.2. Augmentasi Gambar 7](#_Toc4064163)

[2.3. Bounding box 8](#_Toc4064164)

[2.4. Metode dalam pencocokan gambar *(Image Similarity)* 8](#_Toc4064165)

[2.4.1. Hashing 8](#_Toc4064166)

[2.4.2. SIFT (Scale-Invariant Feature Transform) 8](#_Toc4064167)

[2.4.3. SURF (Speeded-UP Robust Features) 10](#_Toc4064168)

[2.5. Metode dalam pencocokkan teks *(Text Similarity)* 11](#_Toc4064169)

[2.5.1. Levensthein Distance 11](#_Toc4064170)

[2.5.2. Sequence Matcher 12](#_Toc4064171)

[2.6. Benchmarking 12](#_Toc4064172)

[Pembahasan 13](#_Toc4064173)

[3.1. Metode Penelitian 13](#_Toc4064175)

[3.2. Augmentasi Gambar 13](#_Toc4064176)

[3.2.1. Proses 13](#_Toc4064177)

[3.2.2. Hasil Augmentasi 15](#_Toc4064178)

[3.2.2.1. Augmentasi hanya pada gambar 15](#_Toc4064179)

[3.2.2.2. Augmentasi pada gambar dan *bounding box* 15](#_Toc4064180)

[3.2.3. Cara Penggunaan 16](#_Toc4064181)

[3.2.3.1. Augmentasi Gambar 16](#_Toc4064182)

[*3.2.3.2.* Augmentasi pada Gambar dan *Bounding Box* 17](#_Toc4064183)

[3.3. Pencocokkan Gambar (*Image Similarity*) 19](#_Toc4064184)

[3.3.1. Metode Image Similarity 19](#_Toc4064185)

[3.3.2. Skenario uji penelitian 20](#_Toc4064186)

[3.3.3. Hasil 20](#_Toc4064187)

[3.3.3.1. Metode Hashing Similarity 20](#_Toc4064188)

[3.3.3.2. SIFT 21](#_Toc4064189)

[3.3.3.3. SURF 22](#_Toc4064190)

[3.3.4. Analisis 23](#_Toc4064191)

[3.4. Pencocokkan Teks *(Text Similarity)* 24](#_Toc4064192)

[3.4.1. Metode Text Similarity 24](#_Toc4064193)

[3.4.2. Skenario uji penelitian 24](#_Toc4064194)

[3.4.3. Hasil 25](#_Toc4064195)

[3.4.3.1. Metode Levensthein Distance 25](#_Toc4064196)

[3.4.3.2. Metode Sequence Matcher 28](#_Toc4064197)

[3.4.4. Analisis 31](#_Toc4064198)

[Penutup 33](#_Toc4064199)

[4.1. Kesimpulan 33](#_Toc4064201)

[4.2. Saran 33](#_Toc4064202)

[Daftar Pustaka 34](#_Toc4064203)

# Pendahuluan

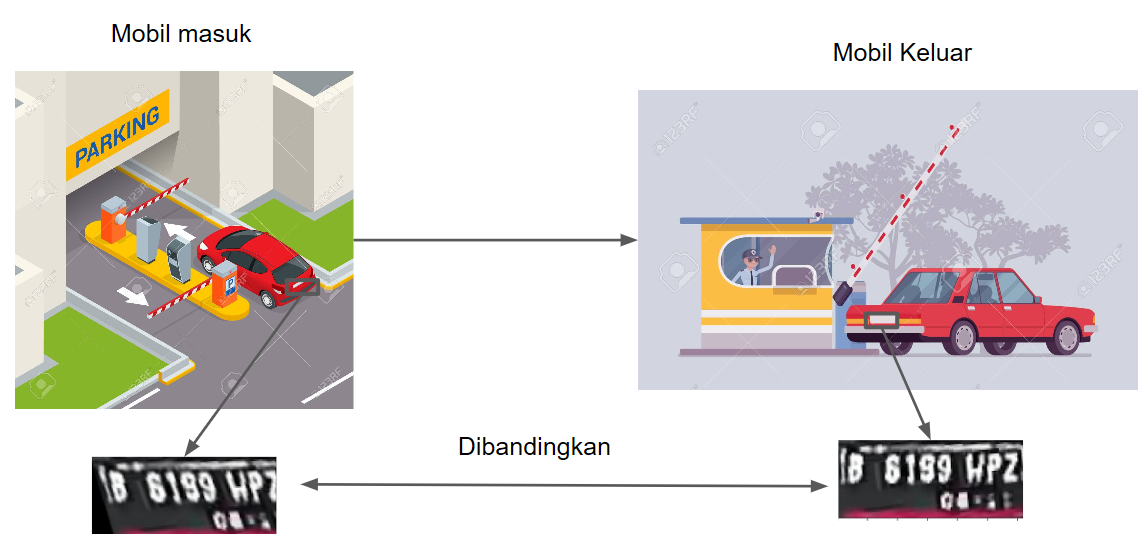
## Latar Belakang

Dengan semakin canggihnya teknologi dan semakin kuatnya kemampuan komputasi dari komputer, banyak aplikasi dan algoritma yang sebelumnya tidak memungkinkan untuk diterapkan di laptop atau telepon genggam sekarang bisa diimplementasikan dengan mudah. Salah satu teknik yang menjadi populer adalah kecerdasan buatan atau *Artiﬁcial Intelligence* (AI) menjadi terkenal karena banyak aplikasi jenis ini yang sekarang bisa ditemui sehari hari seperti siri pada iPhone dan speech recognition atau pengenal suara pada telepon seluler Android. Seiring dengan munculnya istilah AI, Pembelajaran Mesin atau *Machine Learning* (ML), Jaringan Syaraf tiruan *Artiﬁcial Neural Network* (NN), dan *Deep Learning* (DL), merupakan sebuah algoritma kompleks yang dapat mempelajari suatu hal dari data-data yang diberikan.

Algoritma kompleks dari *Deep Learning* dapat diimplementasikan di dalam kehidupan nyata, salah satunya yaitu dalam penentuan pengklasifikasian jenis gambar. Tetapi, masalah yang seringkali timbul di dalam kasus nyata adalah adanya ketidaksetimbangan data, atau disebut *imbalance dataset*. Kasus tersebut dapat ditangani dengan menggunakan augmentasi pada gambar yang memiliki jenis yang minoritas.

Atau dalam hal lainnya seperti plat nomor kendaraan di jalanan. Data yang di dapat di lapangan selalu tidak sebagus dibandingkan dengan data yang di *train*. Sehingga hal tersebut akan membuat performa dari model *Deep Learning* menjadi kurang baik. Kasus tersebut juga dapat ditangani dengan menggunakan augmentasi pada gambar plat nomor kendaraan dengan arah orientasi yang menyerupai data di lapangan.

Selain itu, terdapat juga beberapa algoritma untuk mencocokkan gambar maupun teks dari sebuah gambar, salah satunya adalah dari plat nomor kendaraan bermotor. Aplikasi dari mencocokkan plat nomor tersebut dapat diilustrasikan pada gambar di bawah ini.



Berdasarkan gambar diatas dapat dilihat bahwa apabila terdapat mobil yang masuk ke parkiran ataupun jalan tol, maka plat nomor kendaraan tersebut akan direkam dan disimpan. Pada saat mobil tersebut keluar parkiran atau jalan tol, maka plat nomor tersebut akan direkam ulang dan dibandingkan dengan semua plat nomor yang telah disimpan. Permasalahan yang sering timbul adalah perbedaan gambar plat nomor yang terjadi saat kendaraan masuk dan keluar.

Lalu, dari algoritma tersebut, yang manakah algoritma yang memiliki hasil yang terbaik? Hasil tersebut dapat ditentukan dengan membandingkan metode-metode tersebut yang mana berada di dalam makalah ini.

## Tujuan

1. Menampilkan augmentasi pada gambar plat nomor kendaraan bermotor
2. Menampilkan augmentasi pada gambar beserta *bounding box* di dalam plat nomor kendaraan bermotor
3. Mengetahui performa dari berbagai metode pencocokkan gambar pada plat nomor kendaraan bermotor
4. Mengetahui performa dari berbagai metode pencocokkan teks pada plat nomor kendaraan bermotor

# Landasan Teori



## Imbalance Dataset

Di dalam dunia pembelajaran mesin (*Machine Learning*), komputer dapat membuat suatu prediksi atau keputusan berdasarkan data yang telah dipelajarinya. Data tersebut dinamakan sebagai *dataset*. Sebuah dataset memiliki sebuah label yang mana label tersebut merupakan keputusan yang akan ditentukan oleh komputer nantinya. Misal terdapat suatu kumpulan gambar mobil dan motor, mobil dan motor tersebut adalah label dari dataset gambar tersebut.

Di dalam kebanyakan kasus, dataset gambar tersebut sering terjadi ketidaksetimbangan label. Seperti terdapat 1000 gambar mobil, 10.000 gambar motor, dan 10 gambar truk. Fenomena tersebut sering dinamakan sebagai *imbalance dataset*. Hal tersebut merupakan suatu ancaman besar untuk sebuah model pembelajaran mesin karena perhitungan probabilitas pada suatu label akan menjadi terlalu dominan, misal label motor dalam kasus tersebut. Untuk menangani hal tersebut, diperlukan augmentasi gambar di dataset tersebut.

## Augmentasi Gambar

Augmentasi gambar merupakan suatu proses untuk memperbanyak suatu jumlah gambar dengan cara memodifikasi atau mengubah orientasi gambar tanpa mengubah makna dari gambar tersebut.

Misalkan terdapat sebuah gambar mobil, apabila gambar tersebut diputar, maka gambar tersebut tetap dapat dianalisis oleh manusia sebagai label mobil, tetapi belum tentu dapat dianalisis oleh komputer sebagai label mobil. Hal tersebut terjadi karena komputer memproses data berdasarkan piksel per piksel.

Maka dari itu, agar komputer dapat memiliki kemampuan analisis yang hampir sama seperti manusia, maka segala kemungkinan orientasi gambar harus ditambahkan sebagai dataset. Sehingga apabila data yang diaugmentasi berasal dari data yang memiliki jenis label terkecil, maka data tersebut menjadi semakin banyak dan dapat mengimbangi jumlah data jenis label lainnya, dan augmentasi menjadi solusi yang tepat untuk masalah *imbalance dataset*.

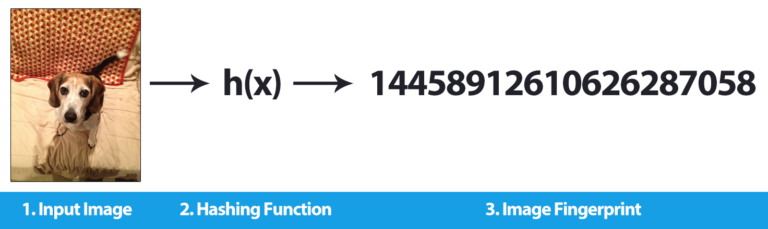
## Bounding box

Apabila terdapat suatu gambar mobil di jalan raya, maka kita dapat melihat dimana letak mobil, pohon maupun orang. Tetapi tidak dengan komputer, karena komputer hanya dapat memprediksi satu jenis label baik itu mobil, pohon, atau orang dalam satu gambar. Solusi agar komputer agar dapat mengenali ketiga label tersebut dalam satu gambar adalah dengan memotong atau melokalisasi gambar tersebut menjadi gambar hanya mobil, pohon, dan orang. Daerah lokalisasi tersebut ditandai dengan suatu garis persegi panjang yang dinamakan *bounding box*.

## Metode dalam pencocokan gambar *(Image Similarity)*

### Hashing

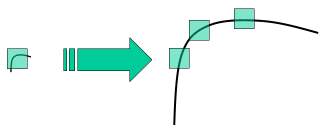
Sebuah gambar dapat direpresentasikan hanya sebagai suatu angka tertentu apabila dimasukkan ke dalam suatu fungsi, fungsi tersebut dinamakan *hashing function*. Biasanya fungsi hashing digunakan untuk kriptografi. Ilustrasi dari hal tersebut dapat digambarkan di dalam gambar di bawah ini.



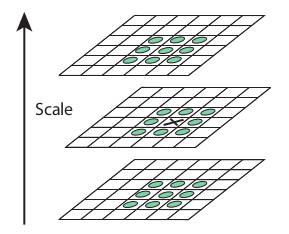
Pada ilustrasi tersebut dapat dilihat bahwa gambar seekor anjing berubah menjadi angka yang disebut *image fingerprint*. Dari *image fingerprint* inilah dapat dimanfaatkan sebagai suatu metode pencocokan gambar *(image similarity)* hanya dengan mencocokkan angka dari *fingerprint*.

### SIFT (Scale-Invariant Feature Transform)

SIFT merupakan sebuah singkatan dari *Scale-Invariant Feature Transform* yang merupakan suatu metode yang dapat digunakan untuk mencocokkan suatu gambar (*Image Similarity)*. SIFT bekerja dengan mencari titik-titik unik dari setiap foto yang biasa disebut sebagai *keypoint*.



Berdasarkan gambar di atas dapat dilihat terdapat dua gambar yang sama tetapi dengan ukuran yang berbeda. Apabila kita melihat gambar tersebut dengan rentang di daerah kotak hijau, maka dapat dilihat sebuah garis lengkung di dalam gambar kiri, sedangkan untuk gambar kanan maka hanya akan dilihat sebuah garis yang hampir lurus. SIFT menggunakan dasar ide ini untuk membuat titik mana saja yang terbaik agar dapat dijadikan *keypoints*.

****

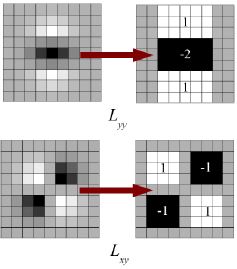
Langkah pertama adalah gambar di buat blur dengan *Gaussian Blur Kernel* dengan besarnya \sigma yang bervariasi. Setelah itu dicari titik yang memiliki intensitas maksimum di daerah lokal. Setelah didapat, maka setiap titik tersebut dibandingkan nilainya dengan tetangga terdekatnya dan titik di lokasi itu dengan besarnya \sigma yang lain. Apabila nilai pada titik tersebut masih dominan, maka titik tersebut berpotensi untuk dijadikan *keypoints*. Contoh *keypoints* pada suatu gambar dapat dilihat pada gambar di bawah ini.



Setelah didapat daerah pada titik *keypoint*, maka metode ini bisa digunakan untuk mencocokkan suatu gambar dengan cara mencari *keypoint* pada gambar yang lain. Setelah didapat *keypoint* pada gambar yang lain, maka *keypoint* dari kedua gambar dibandingkan.

### SURF (Speeded-UP Robust Features)

SURF merupakan sebuah singkatan dari *Speeded-Up Robust Features* yang mana merupakan suatu metode lain untuk mencocokkan suatu gambar dengan gambar yang lain *(Image Similarity)*. Apabila SIFT bekerja dengan cara membuat gambar menjadi blur dengan skala yang berbeda dan dicari titik *keypoint-*nya. maka SURF bekerja dengan cara menjumlahkan nilai-nilai dari suatu daerah lokal, atau biasa disebut sebagai *integral images*, ilustrasi gambarannya dapat dilihat pada gambar dibawah ini.



Berdasarkan gambar diatas, dapat dilihat bahwa kalkulasi pada metode SURF terlihat menjadi lebih kasar. Karena hal itulah biasanya metode SURF lebih cepat durasi kalkulasinya dibandingkan dengan SIFT. Pada gambar di bawah ini merupakan titik-titik unik (*keypoint)* yang didapat dari metode SURF.

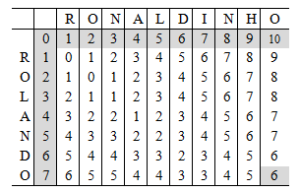


Setelah didapat daerah pada titik *keypoint*, maka metode ini bisa digunakan untuk mencocokkan suatu gambar dengan cara mencari *keypoint* pada gambar yang lain. Setelah didapat *keypoint* pada gambar yang lain, maka *keypoint* dari kedua gambar dibandingkan.

## Metode dalam pencocokkan teks *(Text Similarity)*

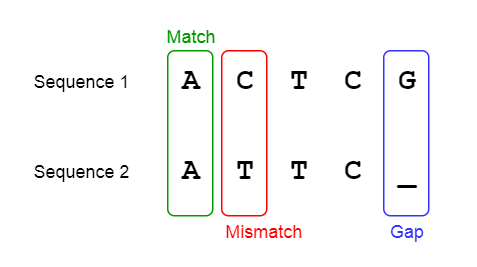
### Levensthein Distance

*Levenshtein distance* merupakan salah satu metode yang digunakan untuk mengukur nilai kesamaan atau kemiripan antara dua buah kata (string). Pengukuran *Levensthein distance* biasanya dapat dengan mudah direpresentasikan dalam bentuk matriks seperti gambar di bawah ini. Pada gambar di bawah ini menunjukkan hasil perbandingan dari kata “Ronaldinho” dan kata “Rolando”. Skor dari perbandingan kata tersebut berada di ujung kanan matriks. Semakin kecil skor yang didapat, maka kedua teks tersebut semakin mirip.

****

### Sequence Matcher

Sequence Matcher merupakan metode lain yang dapat digunakan untuk mencocokkan suatu teks. Metode tersebut mengambil kemiripan dengan mencari huruf yang sama dengan label dan membaginya dengan jumlah huruf keseluruhan



Nilai kemiripan tersebut mempunyai rentang penilaian dari 0 sampai 1. Nilai tersebut menunjukan apabila nilai kemiripan mendekati nilai 1 maka hasil testing semakin mirip dengan data label, begitu sebaliknya.

## Benchmarking

Apabila kita dihadapkan dengan beberapa metode algoritma yang memiliki tujuan yang sama, maka kita harus memilih salah satu dari metode algoritma tersebut. Metode yang dipilih harus merupakan metode yang memiliki nilai keakurasian yang tinggi dengan durasi yang sesingkat-singkatnya. Proses untuk dapat melihat performa dari masing-masing metode tersebut biasa dinamakan dengan istilah *benchmarking.*

# Pembahasan



## Metode Penelitian

Metode penelitian yang dilakukan pada kesempatan kali ini adalah metode penelitian dan pengembangan (*Research and Development*). Hal yang dikembangkan adalah Augmentasi Gambar pada plat nomor kendaraan bermotor, sedangkan hal yang diteliti adalah kemampuan performa dari kemiripan gambar (*Image Similarity*) pada plat nomor kendaraan bermotor.

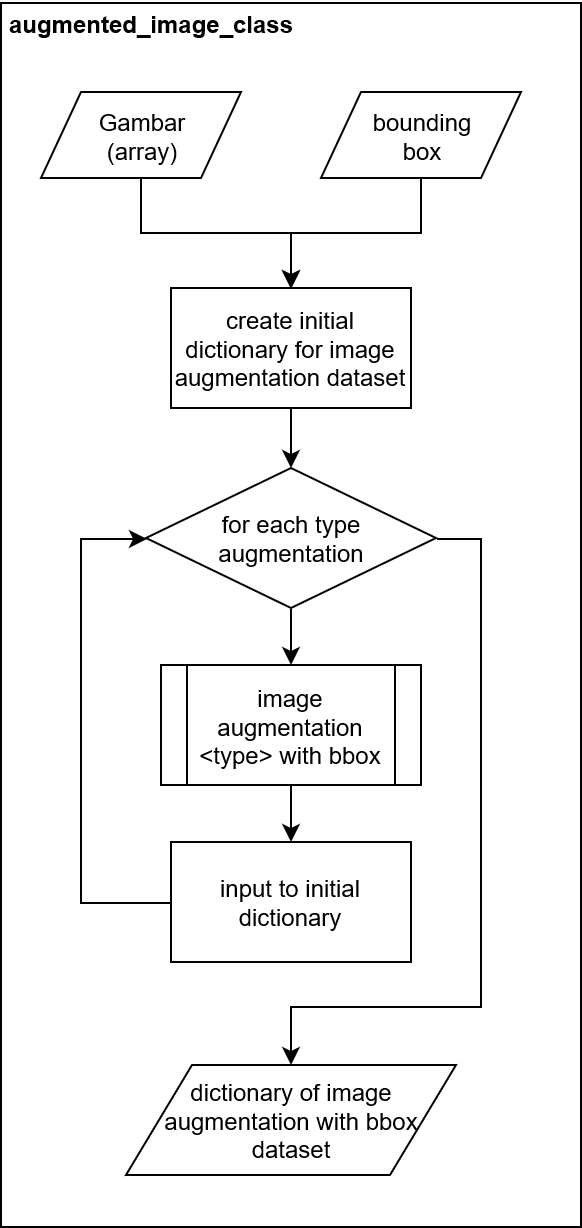
## Augmentasi Gambar

### Proses

Augmentasi gambar pada proyek yang telah dikembangkan ini dilakukan pada gambar plat nomor kendaraan bermotor. Proses augmentasi dilakukan dengan bahasa pemrograman Python 3 dan *library* OpenCV. Augmentasi gambar terdiri dari dua bagian, yaitu augmentasi hanya pada gambar, dan augmentasi pada gambar beserta *bounding box*.

Untuk augmentasi hanya pada gambar, semua codingan berada di dalam folder 1.Image\_Augmentation/Image\_Augmentation. Folder tersebut dapat langsung dipanggil di Python dengan cara mengetik perintah from Image\_Augmentation import image\_augmentation. Hasil dari perintah tersebut merupakan *class* yang mana hanya dengan memasukkan argumen variabel gambar berupa matriks (*Array*) dengan tipe data integer 8 bit (uint8). Hasil output tersebut berupa sebuah data dengan tipe *dictionary* yang mana di dalam data tersebut terdapat beberapa gambar dengan jenis augmentasi yang berbeda-beda..

Untuk augmentasi pada gambar beserta *bounding box*, semua codingan berada di dalam folder 1.Image\_Augmentation/Image\_Augmentation\_bbox. Folder tersebut dapat langsung dipanggil di Python dengan cara mengetik perintah from Image\_Augmentation\_bbox import augmented\_image\_class. Hasil dari perintah tersebut merupakan *class* yang mana hanya dengan memasukkan argumen variabel gambar berupa matriks (*Array*) dengan tipe data integer 8 bit (uint8), dan sebuah data dari letak lokasi titik *bounding box* bertipe dictionary dengan konvensi bentuk yang akan dijelaskan di bagian cara penggunaan. Hasil output tersebut berupa sebuah data dengan tipe *dictionary* yang mana di dalam data tersebut terdapat beberapa gambar dengan jenis augmentasi yang berbeda-beda, dan di masing-masing gambar tersebut terdapat lokasi *bounding box*-nya dengan bentuk *dictionary*. Berikut ini merupakan alur proses dari augmentasi pada gambar beserta *bounding box* :

A screenshot of a social media post

Description automatically generated

### Hasil Augmentasi

Tabel di bawah ini merupakan hasil dari augmentasi pada gambar plat nomor kendaraan bermotor. Terdapat dua tabel yang mana satu tabel untuk augmentasi hanya pada gambar, dan yang satu lagi augmentasi untuk gambar beserta *bounding box*.

#### Augmentasi hanya pada gambar

|  |  |  |  |
| --- | --- | --- | --- |
| Shear : left | Shear : right | Intensity : glow | Intensity : dark |
| Scale : small | Scale : big | Rotate : left | Rotate : right |
| Translate : up | Translate : down | Translate : right | Translate : left |
| Blur : very low | Blur : low | Blur : med | Blur : high |
| Crop : very low | Crop : low | Crop : med | Crop : high |
| Noise : gaussian | Noise : salt & pepper | Noise : speckle |  |

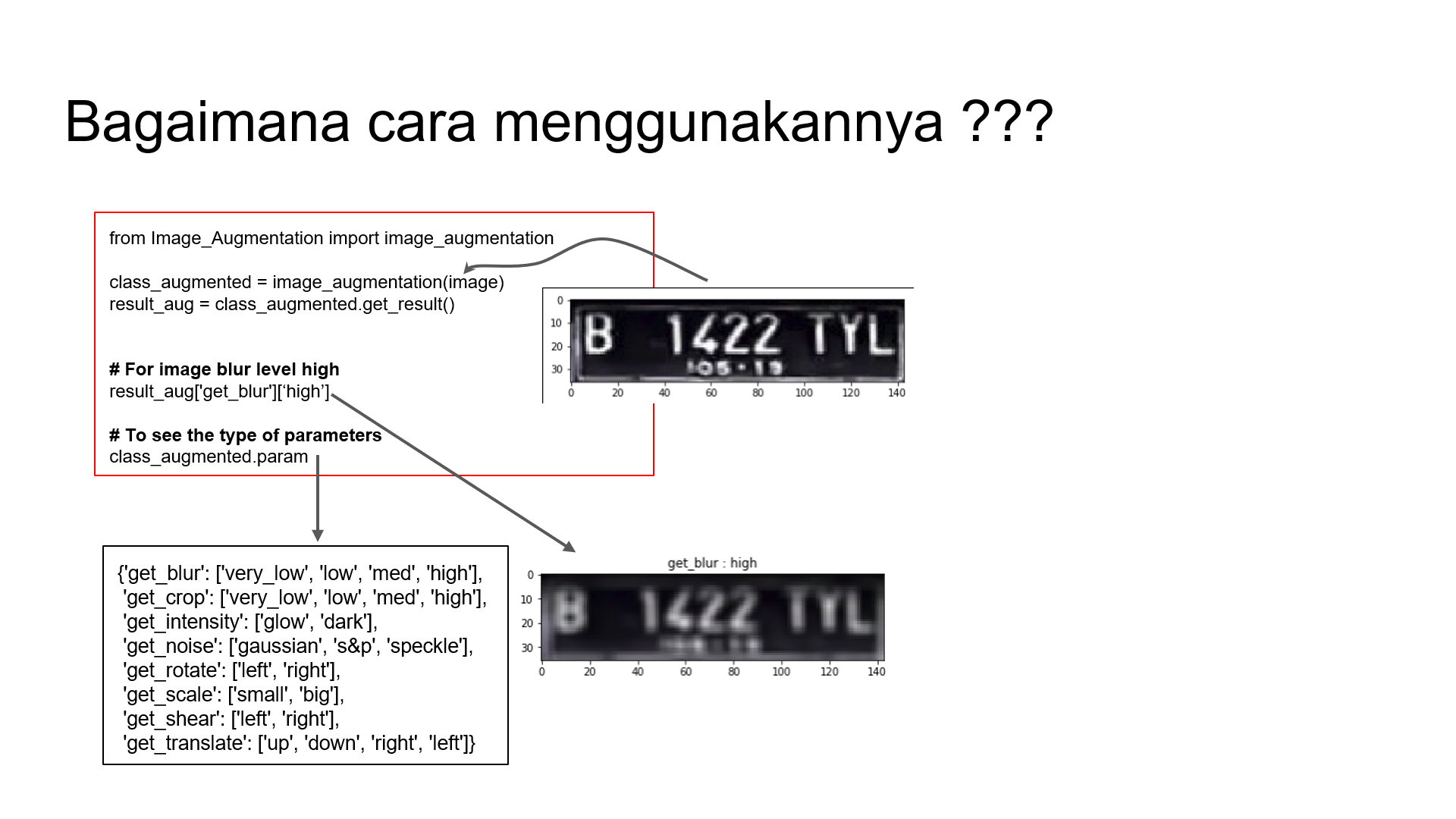
#### Augmentasi pada gambar dan *bounding box*

|  |  |  |  |
| --- | --- | --- | --- |
| Intensity : dark | Intensity : glow | Scale : big | Scale : small |
| Blur : very low | Blur : low | Blur : med | Blur : high |
| Translate : up | Translate : left | Translate : down | Translate : right |
| Noise : salt & pepper | Noise : gaussian | Noise : speckle | Shear : left |
| Shear : right | Rotate Shear : left | Rotate Shear : right | Rotate : left |
| Rotate : right |  |  |  |

### Cara Penggunaan

#### Augmentasi Gambar

*Package* yang telah dibuat untuk augmentasi pada gambar terdapat di dalam folder 1.Image\_Augmentation/Image\_Augmentation. Hanya dengan memasukkan gambar yang berbentuk array, maka akan dihasilkan beberapa gambar yang sudah secara otomatis dihasilkan. Berikut ini merupakan contoh program untuk menggunakan *package* tersebut.



Berdasarkan bagan diatas dapat dilihat bahwa fungsi *image\_augmentation* hanya membutuhkan input berupa gambar yang berbentuk array (image). Setelah itu, untuk menghasilkan gambar yang di augmentasi, *class* yang telah dibuat tersebut diberi subclass *get\_result* yang akan mengeluarkan output berupa data *dictionary*. Data *dictionary* yang dihasilkan tersebut dapat apabila kita menginginkan gambar augmentasi yang diblur secara ekstrim, maka dapat memilih **get\_blur** dengan mode **high**, maka akan dihasilkan gambar berbentuk array yang telah diblur.

Untuk melihat jenis-jenis augmentasi apa saja yang tersedia, maka *class* augmentasi tersebut ditambahkan dengan subclass *param* sehingga akan muncul tampilan seperti gambar di atas.

#### Augmentasi pada Gambar dan *Bounding Box*

*Package* yang telah dibuat untuk augmentasi pada gambar beserta *bounding box* terdapat di dalam folder 1.Image\_Augmentation/Image\_Augmentation\_bbox. Hanya dengan memasukkan gambar yang berbentuk array, maka akan dihasilkan beberapa gambar yang sudah secara otomatis dihasilkan. Berikut ini merupakan contoh program untuk menggunakan *package* tersebut.

A screenshot of text

Description automatically generated

Berdasarkan bagan diatas dapat dilihat bahwa fungsi *augmented\_image\_class* hanya membutuhkan input berupa gambar yang berbentuk array (image) dan *bounding box* yang berbentuk *dictionary* (dict\_bbox). Setelah itu, untuk menghasilkan gambar dan *bounding box* yang di augmentasi, *class* yang telah dibuat tersebut diberi subclass *get\_result* yang akan mengeluarkan output berupa data *dictionary*. Data *dictionary* yang dihasilkan tersebut dapat dilihat pada gambar dibawah ini.

A screenshot of a cell phone

Description automatically generated

Pada gambar diatas dapat dilihat bahwa apabila kita menginginkan gambar augmentasi yang skalanya diperkecil, maka dapat memilih **scale** dengan mode **small** dan jenis **images**. Maka akan dihasilkan gambar berbentuk array yang telah diperkecil. Apabila ingin menampilkan *bounding box* dari augmentasi tersebut, maka pemilihan sama seperti sebelumnya tetapi dengan jenis **bbox\_char** sehingga akan ditampilkan hasil seperti gambar diatas.

Untuk melihat jenis-jenis augmentasi apa saja yang tersedia, maka *class* augmentasi tersebut ditambahkan dengan subclass *parameter\_mode* sehingga akan muncul tampilan seperti gambar di atas.

## Pencocokkan Gambar (*Image Similarity*)

### Metode Image Similarity

*Image Similarity* atau proses untuk mencocokan gambar yang sama pada kesempatan kali ini telah dikembangkan untuk gambar plat nomor kendaraan bermotor. Proses pencocokan gambar dilakukan dengan bahasa pemrograman Python 3 dan beberapa metode untuk pencocokan gambar. Metode yang digunakan adalah *Hashing Image, SIFT,* dan *SURF*. Untuk metode *SIFT* dan *SURF* sudah terdapat di dalam *library* OpenCV.

### Skenario uji penelitian

Untuk membandingkan performa pada metode pencocokan gambar (*Image Similarity*), diambil 10 sampel gambar secara acak dari *database*, yang mana gambar yang berada di *database* sebesar 1169 gambar plat nomor kendaraan. Setelah itu, masing-masing dari kesepuluh gambar tersebut diaugmentasi dengan jenis augmentasi ditentukan secara acak. Selanjutnya, masing-masing kesepuluh gambar tersebut dicocokan dengan semua gambar yang berada di dalam *database* dan menampilkan 3 gambar termirip berdasarkan skor kemiripan yang tertinggi, dan ditampilkan plot kurva distribusi persentase skor kemiripan gambar dari hasil proses tersebut.

Skor penilaian untuk akurasi ditentukan berdasarkan urutan gambar asli dari gambar yang di augmentasi. Apabila gambar yang asli berada di posisi pertama maka skor akan bernilai 3, apabila posisi kedua maka skor akan bernilai 2, apabila posisi ketiga maka skor akan bernilai 1, dan apabila berada di posisi seterusnya maka skor akan bernilai 0. Skor tersebut kemudian dijumlahkan dan dibagi berdasarkan banyaknya sampel yang diteliti. Kemudian dijadikan persentase.

Selain itu, selama proses pencocokan gambar, juga dicatat lamanya durasi dari proses pencocokan gambar 1 terhadap gambar yang berada di *database*.

Selanjutnya adalah melakukan hal yang sama dari proses tersebut untuk metode pencocokan gambar lainnya. Hasil dari proses pencocokan gambar pada ketiga metode tersebut ditampilkan pada bagian hasil.

### Hasil

#### Metode Hashing Similarity

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Duration similarity image per image : *0.001039 sec*

Accuracy : 53.3333 %

#### SIFT

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Duration similarity image per image : *0.012375 sec*

Accuracy : 100.0000 %

#### SURF

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Duration similarity image per image : *0.003389 sec*

Accuracy : 86.6667 %

### Analisis

Besarnya performa berupa akurasi dan durasi pencocokan antara 2 gambar dari setiap metode dapat ditulis dalam tabel di bawah ini.

|  |  |  |
| --- | --- | --- |
|  | **Akurasi** | **Durasi pencocokan antara 2 gambar** |
| **Hashing Similarity** | 53.3333 % | 0.001039 detik |
| **SIFT Similarity** | 100.0000 % | 0.012375 detik |
| **SURF Similarity** | 86.6667 % | 0.003389 detik |

Berdasarkan tabel di atas, dapat dilihat bahwa metode SIFT memiliki nilai akurasi yang paling baik dibandingkan dengan metode lainnya. Tetapi metode SIFT juga merupakan metode yang membutuhkan durasi paling lama dibandingkan dengan metode lainnya. Selanjutnya, untuk metode SURF memiliki akurasi yang juga cukup baik dibandingkan dengan *hashing similarity*, dan durasi dengan metode SURF juga lebih cepat dibandingkan dengan metode SIFT. Dan yang terakhir, untuk metode *hashing similarity* memiliki waktu yang paling singkat dibandingkan dengan metode lainnya, tetapi akurasi yang dihasilkan sangat buruk, yaitu hampir hanya separuhnya saja.

## Pencocokkan Teks *(Text Similarity)*

### Metode Text Similarity

*Text Similarity* atau proses untuk mencocokan teks yang sama pada kesempatan kali ini telah dikembangkan untuk teks pada plat nomor kendaraan bermotor. Proses pencocokan teks dilakukan dengan bahasa pemrograman Python 3 dan beberapa metode untuk pencocokan teks. Metode yang digunakan adalah Levensthein Distance dan Sequence Matcher.

Untuk metode Levensthein Distance berada di dalam file levensthein.py, sementara untuk metode Sequence Matcher telah tersedia di dalam library python bernama difflib.

### Skenario uji penelitian

Untuk membandingkan performa pada metode pencocokan teks (*Text SImilarity*), diambil 1 sampel teks plat nomor yang paling unik dari *database*, yang mana banyaknya jenis teks di dalam database sebanyak 1169 jenis. Setelah itu, sampel teks tersebut divariasikan berupa pengurangan satu persatu kata dari sisi kanan dan kiri hingga sebanyak 3 huruf, dan pengacakan satu persatu kata dari sisi kanan dan kiri hingga sebanyak 3 huruf. Selanjutnya, masing-masing hasil variasi tersebut dicocokan dengan semua gambar yang berada di dalam *database* dan menampilkan 4 jenis teks termirip berdasarkan skor kemiripan yang tertinggi.

Skor penilaian untuk akurasi ditentukan berdasarkan urutan teks asli dari teks yang di variasikan. Apabila teks yang asli berada di posisi pertama maka skor akan bernilai 3, apabila posisi kedua maka skor akan bernilai 2, apabila posisi ketiga maka skor akan bernilai 1, dan apabila berada di posisi seterusnya maka skor akan bernilai 0. Skor tersebut kemudian dijumlahkan dan dibagi berdasarkan banyaknya sampel yang diteliti. Kemudian dijadikan persentase.

Selain itu, selama proses pencocokan teks, juga dicatat lamanya durasi dari proses pencocokan teks dari masing-masing variasi teks terhadap semua jenis teks yang berada di *database*.

Selanjutnya adalah melakukan hal yang sama dari proses tersebut untuk metoda pencocokan teks lainnya. Hasil dari proses pencocokan teks pada kedua metode tersebut ditampilkan pada bagian hasil.

### Hasil

#### Metode Levensthein Distance

|  |
| --- |
| **Pengurangan kata (dari sisi kanan)**  REDUCTION 1 WORD (from right side)  ==================================================  key : B4838SC  Original key : B4838SCN  Duration : 0:00:00.629836  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 57.1429 % ID : 2 plat : B4913BPC  score : 57.1429 % ID : 26 plat : B1483SLP  score : 57.1429 % ID : 89 plat : B389BSP  ==================== Finish ====================  REDUCTION 2 WORD (from right side)  ==================================================  key : B4838S  Original key : B4838SCN  Duration : 0:00:00.551917  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 83.3333 % ID : 430 plat : B6231S  score : 83.3333 % ID : 875 plat : B8381VS  score : 66.6667 % ID : 26 plat : B1483SLP  ==================== Finish ====================  REDUCTION 3 WORD (from right side)  ==================================================  key : B4838  Original key : B4838SCN  Duration : 0:00:00.451274  ==================== Result ====================  score : 100.0000 % ID : 111 plat : B8GD  score : 100.0000 % ID : 427 plat : B7031  score : 100.0000 % ID : 463 plat : B4186T  score : 100.0000 % ID : 899 plat : B8KN  ==================== Finish ==================== |
| **Pengurangan kata (dari sisi kiri)**  REDUCTION 1 WORD (from left side)  ==================================================  key : 4838SCN  Original key : B4838SCN  Duration : 0:00:00.712576  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 57.1429 % ID : 477 plat : B1386CB  score : 57.1429 % ID : 611 plat : B8180CY  score : 57.1429 % ID : 875 plat : B8381VS  ==================== Finish ====================  REDUCTION 2 WORD (from left side)  ==================================================  key : 838SCN  Original key : B4838SCN  Duration : 0:00:00.605252  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 66.6667 % ID : 477 plat : B1386CB  score : 66.6667 % ID : 611 plat : B8180CY  score : 66.6667 % ID : 875 plat : B8381VS  ==================== Finish ====================  REDUCTION 3 WORD (from left side)  ==================================================  key : 38SCN  Original key : B4838SCN  Duration : 0:00:00.474401  ==================== Result ====================  score : 100.0000 % ID : 899 plat : B8KN  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 80.0000 % ID : 111 plat : B8GD  score : 80.0000 % ID : 179 plat : 3135R  ==================== Finish ==================== |
| **Pengacakan kata (dari sisi kanan)**  RANDOM 1 WORD (from right side)  ==================================================  key : B4838SCT  Original key : B4838SCN  Duration : 0:00:00.809448  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 57.1429 % ID : 26 plat : B1483SLP  score : 57.1429 % ID : 51 plat : B6825SHT  score : 57.1429 % ID : 106 plat : B4936SEM  ==================== Finish ====================  RANDOM 2 WORD (from right side)  ==================================================  key : B4838SSS  Original key : B4838SCN  Duration : 0:00:00.754603  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 83.3333 % ID : 875 plat : B8381VS  score : 66.6667 % ID : 26 plat : B1483SLP  score : 66.6667 % ID : 106 plat : B4936SEM  ==================== Finish ====================  RANDOM 3 WORD (from right side)  ==================================================  key : B4838VSM  Original key : B4838SCN  Duration : 0:00:00.904185  ==================== Result ====================  score : 100.0000 % ID : 875 plat : B8381VS  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 80.0000 % ID : 99 plat : B4383TME  score : 80.0000 % ID : 102 plat : B8085PSM  ==================== Finish ==================== |
| **Pengacakan kata (dari sisi kiri)**  RANDOM 1 WORD (from left side)  ==================================================  key : P4838SCN  Original key : B4838SCN  Duration : 0:00:00.746149  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 42.8571 % ID : 26 plat : B1483SLP  score : 42.8571 % ID : 40 plat : B2937SON  score : 42.8571 % ID : 106 plat : B4936SEM  ==================== Finish ====================  RANDOM 2 WORD (from left side)  ==================================================  key : EZ838SCN  Original key : B4838SCN  Duration : 0:00:00.735755  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 50.0000 % ID : 40 plat : B2937SON  score : 50.0000 % ID : 143 plat : B688RCH  score : 50.0000 % ID : 412 plat : B1838WUA  ==================== Finish ====================  RANDOM 3 WORD (from left side)  ==================================================  key : AH238SCN  Original key : B4838SCN  Duration : 0:00:00.728360  ==================== Result ====================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 60.0000 % ID : 40 plat : B2937SON  score : 60.0000 % ID : 146 plat : B2238UQ  score : 60.0000 % ID : 154 plat : B2238FE  ==================== Finish ==================== |

#### Metode Sequence Matcher

|  |
| --- |
| **Pengurangan kata (dari sisi kanan)**  REDUCTION 1 WORD (from right side)  ==================================================  key : B4838SC  Original key : B4838SCN  Duration : 0:00:00.037645  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 76.5306 % ID : 875 plat : B8381VS  score : 71.4286 % ID : 26 plat : B1483SLP  score : 71.4286 % ID : 135 plat : B3598SYC  ====================== Finish ======================  REDUCTION 2 WORD (from right side)  ==================================================  key : B4838S  Original key : B4838SCN  Duration : 0:00:00.031159  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 89.7436 % ID : 875 plat : B8381VS  score : 83.3333 % ID : 26 plat : B1483SLP  score : 83.3333 % ID : 798 plat : B4368SGJ  ====================== Finish ======================  REDUCTION 3 WORD (from right side)  ==================================================  key : B4838  Original key : B4838SCN  Duration : 0:00:00.031960  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 86.6667 % ID : 198 plat : B8358RG  score : 86.6667 % ID : 694 plat : B8328HQ  score : 86.6667 % ID : 875 plat : B8381VS  ====================== Finish ====================== |
| **Pengurangan kata (dari sisi kiri)**  REDUCTION 1 WORD (from left side)  ==================================================  key : 4838SCN  Original key : B4838SCN  Duration : 0:00:00.032850  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 61.2245 % ID : 875 plat : B8381VS  score : 57.1429 % ID : 26 plat : B1483SLP  score : 57.1429 % ID : 135 plat : B3598SYC  ====================== Finish ======================  REDUCTION 2 WORD (from left side)  ==================================================  key : 838SCN  Original key : B4838SCN  Duration : 0:00:00.029397  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 71.7949 % ID : 875 plat : B8381VS  score : 66.6667 % ID : 135 plat : B3598SYC  score : 53.8462 % ID : 89 plat : B389BSP  ====================== Finish ======================  REDUCTION 3 WORD (from left side)  ==================================================  key : 38SCN  Original key : B4838SCN  Duration : 0:00:00.023243  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 80.0000 % ID : 135 plat : B3598SYC  score : 65.0000 % ID : 89 plat : B389BSP  score : 65.0000 % ID : 315 plat : B308NFI  ====================== Finish ====================== |
| **Pengacakan kata (dari sisi kanan)**  RANDOM 1 WORD (from right side)  ==================================================  key : B4838SCT  Original key : B4838SCN  Duration : 0:00:00.040803  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 76.1905 % ID : 875 plat : B8381VS  score : 71.4286 % ID : 26 plat : B1483SLP  score : 71.4286 % ID : 99 plat : B4383TME  ====================== Finish ======================  RANDOM 2 WORD (from right side)  ==================================================  key : B4838SSS  Original key : B4838SCN  Duration : 0:00:00.041161  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 88.8889 % ID : 875 plat : B8381VS  score : 83.3333 % ID : 26 plat : B1483SLP  score : 83.3333 % ID : 798 plat : B4368SGJ  ====================== Finish ======================  RANDOM 3 WORD (from right side)  ==================================================  key : B4838VSM  Original key : B4838SCN  Duration : 0:00:00.044383  ====================== Result ======================  score : 100.0000 % ID : 875 plat : B8381VS  score : 93.7500 % ID : 1000 plat : B4838SCN  score : 78.1250 % ID : 26 plat : B1483SLP  score : 78.1250 % ID : 99 plat : B4383TME  ====================== Finish ====================== |
| **Pengacakan kata (dari sisi kiri)**  RANDOM 1 WORD (from left side)  ==================================================  key : P4838SCN  Original key : B4838SCN  Duration : 0:00:00.037677  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 60.9524 % ID : 875 plat : B8381VS  score : 57.1429 % ID : 26 plat : B1483SLP  score : 57.1429 % ID : 135 plat : B3598SYC  ====================== Finish ======================  RANDOM 2 WORD (from left side)  ==================================================  key : EZ838SCN  Original key : B4838SCN  Duration : 0:00:00.030872  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 71.1111 % ID : 875 plat : B8381VS  score : 66.6667 % ID : 135 plat : B3598SYC  score : 53.3333 % ID : 89 plat : B389BSP  ====================== Finish ======================  RANDOM 3 WORD (from left side)  ==================================================  key : AH238SCN  Original key : B4838SCN  Duration : 0:00:00.028817  ====================== Result ======================  score : 100.0000 % ID : 1000 plat : B4838SCN  score : 80.0000 % ID : 40 plat : B2937SON  score : 80.0000 % ID : 135 plat : B3598SYC  score : 80.0000 % ID : 575 plat : B2780SON  ====================== Finish ====================== |

### Analisis

Besarnya performa berupa posisi teks asli dan durasi pencocokan untuk masing-masing variasi teks dapat dilihat pada tabel dibawah ini :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type Text** | **Scale** | **Levensthein** | | **Sequence Matcher** | |
| Reduction (Right Side) | 1 | 1-st | 0.629836 sec | 1-st | 0.037645 sec |
| 2 | 1-st | 0.551917 sec | 1-st | 0.031159 sec |
| 3 | Out of range | 0.451274 sec | 1-st | 0.031960 sec |
| Reduction (Left Side) | 1 | 1-st | 0.712576 sec | 1-st | 0.032850 sec |
| 2 | 1-st | 0.605252 sec | 1-st | 0.029397 sec |
| 3 | 2-nd | 0.474401 sec | 1-st | 0.023243 sec |
| Random (Right Side) | 1 | 1-st | 0.809448 sec | 1-st | 0.040803 sec |
| 2 | 1-st | 0.754603 sec | 1-st | 0.041161 sec |
| 3 | 2-nd | 0.904185 sec | 2-nd | 0.044383 sec |
| Random (Left Side) | 1 | 1-st | 0.746149 sec | 1-st | 0.037677 sec |
| 2 | 1-st | 0.735755 sec | 1-st | 0.030872 sec |
| 3 | 1-st | 0.728360 sec | 1-st | 0.028817 sec |

Sehingga berdasarkan tabel tersebut, dapat ditarik kesimpulan menjadi :

|  |  |  |
| --- | --- | --- |
|  | **Accuracy** | **Speed** |
| **Levensthein Similarity** | 86.1111 % | Fast |
| **Sequence Matcher Similarity** | 97.2222 % | Very Fast |

Berdasarkan tabel di atas, dapat dilihat bahwa metode Levensthein distance memiliki akurasi yang masih tergolong baik yakni 86.1111 % dengan durasi yang cepat. Tetapi, metode Sequence Matcher memiliki akurasi yang sangat baik yakni 97.2222 % dengan durasi yang sangat cepat dibandingkan dengan Levensthein Distance, hampir 10 kali lipat dari Levensthein Distance.

# Penutup



## Kesimpulan

Berdasarkan hasil dari pembahasan pada bab sebelumnya yang sudah dilakukan dalam penelitian ini, maka didapat kesimpulan bahwa :

* Augmentasi pada gambar dapat digunakan untuk mengatasi data *imbalance* maupun perbedaan perspektif yang mungkin terjadi saat terjadi dilapangan, seperti kasus pada plat nomor kendaraan bermotor.
* SIFT merupakan metode pencocokan gambar pada plat nomor kendaraan yang memiliki akurasi terbaik, namun akan sedikit lama memangkas waktu komputasi.
* Sequence Matcher merupakan metode pencocokan teks yang memiliki akurasi dan waktu komputasi yang terbaik pada plat nomor kendaraan.

## Saran

Setelah didapatkan sebelumnya bahwa penelitian ini masih memiliki banyak kekurangan. Seperti metode yang dibandingkan masih sedikit, keseluruhan metode belum ada yang menggunakan prinsip *Deep Learning* dikarenakan arsitektur *Deep Convolutional Autoencoder* yang penulis bikin belum optimal, dan belum dapat disatukan skor dari *image similarity* dan *text similarity* karena penulis tidak memiliki algoritma OCR dimana kita dapat mendapatkan teks pada gambar plat nomor kendaraan bermotor. Mungkin apabila kekurangan diatas telah terpenuhi, hasil dari *benchmark* akan semakin akurat untuk kedepannya.

# Daftar Pustaka

* 723636837799177. (2017, September 02). Pairwise Sequence Alignment using Biopython. Retrieved from https://towardsdatascience.com/pairwise-sequence-alignment-using-biopython-d1a9d0ba861f
* Image hashing with OpenCV and Python. (2017, November 28). Retrieved from https://www.pyimagesearch.com/2017/11/27/image-hashing-opencv-python/
* Introduction to SIFT (Scale-Invariant Feature Transform). (n.d.). Retrieved from https://opencv-python-tutroals.readthedocs.io/en/latest/py\_tutorials/py\_feature2d/py\_sift\_intro/py\_sift\_intro.html
* Introduction to SURF (Speeded-Up Robust Features). (n.d.). Retrieved from <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html>
* Levenshtein Distance. (2011, May 30). Retrieved from https://saulimbong.wordpress.com/2011/05/17/levenshtein-distance/