



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

Pengertian-pengertian dan teori dari berbagai hal yang digunakan akan dimasukkan di bab ini.

2.1 Gambar

Menurut KBBI, gambar merupakan sebuah tiruan atas barang (orang, binatang, tumbuhan, dan sebagainya) yang dibuat dengan coretan pensil dan sebagainya pada kertas dan sebagainya (Setiawan, 2019) (Kemendikbud, 2019)

Gambar Digital, merupakan representasi visual atas barang atau objek (orang, binatang, tumbuhan, dan sebagainya) yang di simpan dalam bentuk digital.

2.2 Video

Video merupakan kumpulan gambar yang bergerak atau rekaman gambar hidup yang dapat menampilkan representasi visual yang bergerak atas barang atau objek (orang, binatang, tumbuhan, dan sebagainya) yang di simpan di dalam bentuk digital yang dapat disertakan dengan adanya suara. (Kemendikbud, 2019)

2.3 *Machine Learning*

Machine learning adalah sebuah kategori algoritma yang membuat aplikasi perangkat lunak menjadi lebih akurat dalam memprediksi hasil tanpa harus diprogram secara eksplisit. Pada dasarnya *machine learning* adalah membangun algoritma yang dapat menerima *input* dan menggunakan analisis statistika untuk dapat memprediksi *output* (Burn, 2019)

2.4 *Neural Network*

Neural network adalah kumpulan algoritma yang berusaha mengenali relasi antar data melalui proses yang menyerupai otak manusia. *Neural network* dapat beradaptasi pada perubahan *input* sehingga tidak perlu membuat ulang kriteria *output*. *Neural network* bekerja menyerupai otak manusia. “neuron” di dalam *neural network* merupakan fungsi matematik yang mengumpulkan dan melakukan klasifikasi informasi berdasarkan arsitektur tertentu. (Chen, 2019)

Pada umumnya *neural network* memiliki 3 *layer* yakni *input layer*, *output layer*, dan *hidden layer*. Di mana *input layer* merupakan data yang kita masukkan kedalam *neural network*. *output layer* adalah data yang diharapkan. Sedangkan *hidden layer* merupakan fitur-fitur pembobotan yang dapat menentukan *output*. *Hidden layer* tersebut isinya bermacam-macam tergantung dari arsitektur yang digunakan.

2.5 Perangkat Lunak

Perangkat lunak atau peranti lunak adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer.

Pembuatan perangkat lunak itu sendiri memerlukan “bahas pemrograman” yang ditulis oleh seorang pemrogram untuk selanjutnya di kompilasi dengan aplikasi kompilasi sehingga menjadi kode yang bisa dikenali oleh mesin komputer.

Beberapa perangkat lunak yang biasa dikenal:

1. Perangkat lunak aplikasi
2. Sistem operasi
3. Perkakas pengembang perangkat lunak
4. Pengendali perangkat keras

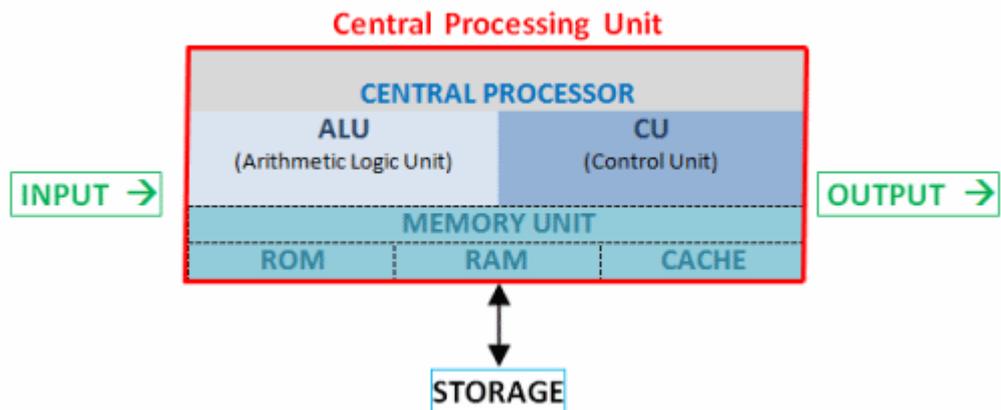
2.6 Perangkat Keras

Perangkat keras adalah semua bagian fisik komputer, dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, dan dibedakan dengan perangkat lunak yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya.

Perangkat keras yang biasa dikenal adalah papan *system*/papan induk, *RAM*, *CPU*, pencatu daya, *hard disk*, *keyboard*, *mouse*, *web cam*, dan *monitor*

2.7 CPU

CPU (*central processing unit*) atau disebut prosesor adalah otak dari komputer, di mana *CPU*. Merupakan bagian terpenting di dalam sistem komputer. *CPU* menerima instruksi dan melakukan kalkulasi. Perintah di ambil dari *RAM* dan di terjemahkan sebelum di jalankan dan mengembalikan hasil bagi program yang mengirimkannya (Martindale, 2019).



Gambar 2.1 Arsitektur Komputer

[https://scholr-](https://scholr-asset.s3.amazonaws.com/media/forum_answer_images/hztrsyywrbms.png)

[asset.s3.amazonaws.com/media/forum_answer_images/hztrsyywrbms.png](https://scholr-asset.s3.amazonaws.com/media/forum_answer_images/hztrsyywrbms.png)

Pada umumnya, seperti pada Gambar 2.1, terdapat dua buah bagian dalam sebuah *CPU*, yaitu *Aritmatic Logic Unit*, dan *Control Unit*. *Aritmatic Logic Unit* berfungsi sebagai komponen yang melakukan operasi aritmetika dan logika, sedangkan *Control Unit* berfungsi sebagai komponen yang melakukan instruksi (Beal, 2019)

2.8 Python

Python merupakan bahasa pemrograman yang dikembangkan oleh Guido van Rossum pada tahun 1990 dan dikomersialkan ke publik dalam bentuk *Open Source* sehingga Python berjalan di banyak platform sistem operasi seperti Windows, Linux/Unix, Mac OS X, OS/2, Amiga dan telepon genggam Nokia yang terdahulu (Python Software Foundations, 2019)



Gambar 2.2 Python Logo

<https://cdn.worldvectorlogo.com/logos/python-3.svg>

Python berjalan di banyak platform sistem operasi, *Python* juga mendukung bentuk paradigma pemrograman seperti struktural, *functional*, *object oriented programming*, prosedural. Seperti bahasa pemrograman lainnya, *Python* sering digunakan sebagai bahasa *scripting*, tetapi bisa juga digunakan dalam berbagai konteks non-*scripting* dengan menggunakan pihak ketiga sebagai pihak program *executable* seperti *py2* atau *PyInstaller Interpreter Python* untuk berbagai sistem operasi.

2.9 *OpenCV*

OpenCV adalah sebuah *library open-source* yang dirilis dengan lisensi *BSD*

3. *OpenCV* dapat digunakan secara gratis untuk keperluan komersial, teroptimasi untuk penggunaan *real-time*, dan juga bisa digunakan di banyak *platform* dan bahasa pemrograman.



Gambar 2.3 Logo OpenCV

https://upload.wikimedia.org/wikipedia/commons/thumb/3/32/OpenCV_Logo_with_text_svg_version.svg/487px-OpenCV_Logo_with_text_svg_version.svg.png

OpenCV digunakan untuk mengubah ukuran gambar, memberikan distorsi, serta melakukan pemetaan dan penyuntingan gambar yang akan di proses.

2.10 Tensor Flow

TensorFlow dibuat oleh *Google Brain Team* adalah sebuah *library* untuk melakukan komputasi numerik dan melakukan *machine learning* secara besar. *TensorFlow* juga di desain untuk dapat menjalankan *deep neural networks* untuk melakukan klasifikasi *handwritten digit*, pengenalan gambar atau objek dalam sebuah gambar, dan lain sebagainya. (Yegulalp, 2019)

Tensorflow merupakan *tools* yang digunakan dalam penelitian ini untuk melakukan *training data*. Yakni gambar yang mengandung manusia dan gambar yang tidak mengandung manusia.

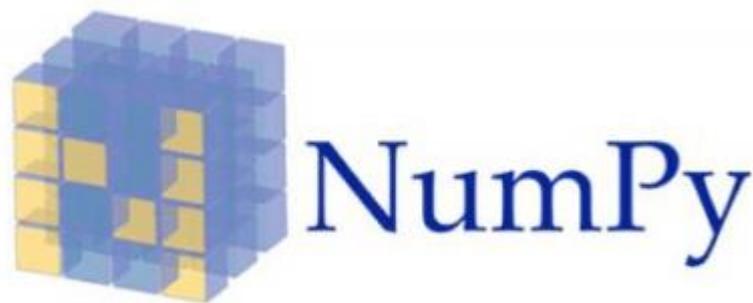


Gambar 2.4 Logo TensorFlow

<https://mpng.pngfly.com/20180805/ipw/kisspng-tensorflow-artificial-intelligence-brand-logo-lear-rapidswitch-nvidia-powered-servers-5b6715663b0830.0909969615334823422418.jpg>

2.11 NumPy

Numpy adalah sebuah *package* yang sangat penting untuk melakukan komputasi ilmiah dengan *Python*.



Gambar 2.5 Logo NumPy

https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwiHu9j--MXmAhWBe30KHRttCykQjRx6BAgBEAQ&url=https%3A%2F%2Fcommons.wikimedia.org%2Fwiki%2Ffile%3ANumpy_logo.svg&psig=AOvVaw0Rzj2Pb-OTrL9C-bfwBFh&ust=1576990346078377

Di dalamnya terdapat hal-hal yang penting, di antaranya adalah:

1. N-dimensional array object
2. program untuk melakukan integrasi c/c++ dan fortran
3. Aljabar linear, *Fourier transform* (mengubah signal menjadi bentuk representasi yang lain, seperti dalam bentuk *sine*, dan *cosine*), dan *random number capabilities*

2.12 Keras

Keras merupakan sebuah *library open source* yang dibuat menggunakan *Python*. *Keras* dapat berjalan menggunakan *TensorFlow*, *CNTK*, *Theano*, dan lain sebagainya. Logo dari *Keras* dapat dilihat pada Gambar 2.6



Gambar 2.6 Logo Keras

<https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjGuN37-cXmAhWRcn0KHTIICGIQjRx6BAgBEAQ&url=https%3A%2F%2Fmlai.ipitek.web.id%2F2018%2F07%2F01%2Fsimple-image-classification-with-keras%2F&psig=AOvVaw0gIALcnAPjYgalv8OAb1fj&ust=1576990594788703>

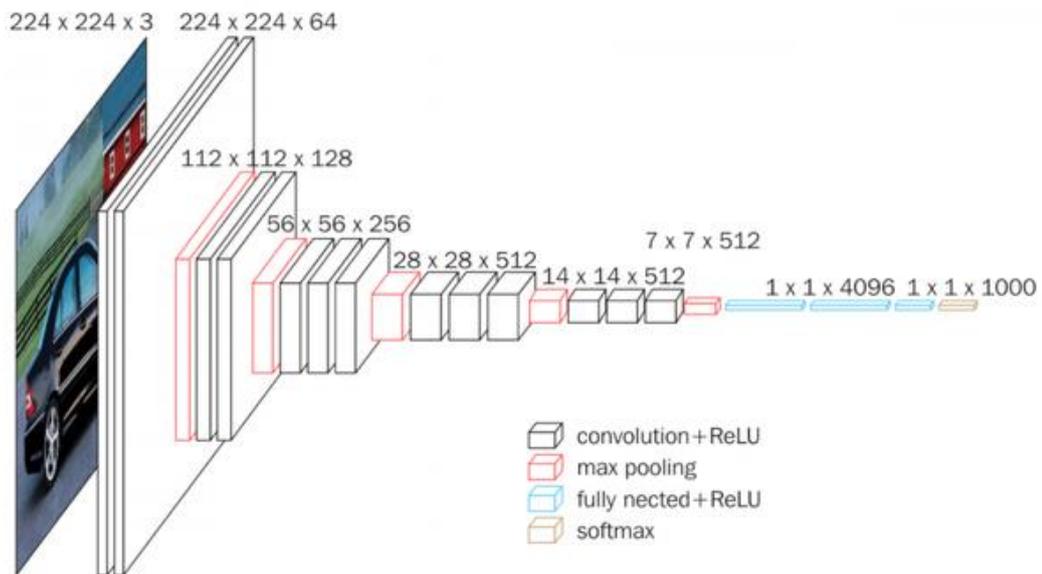
2.13 VGG

VGG (Visual Geometry Group) merupakan sebuah *convolutional neural network* yang dikembangkan oleh Oxford untuk melakukan deteksi terhadap objek. Lapisan yang dimiliki oleh *VGG* dalam melakukan analisa gambar dapat dilihat

pada

Gambar

2.7

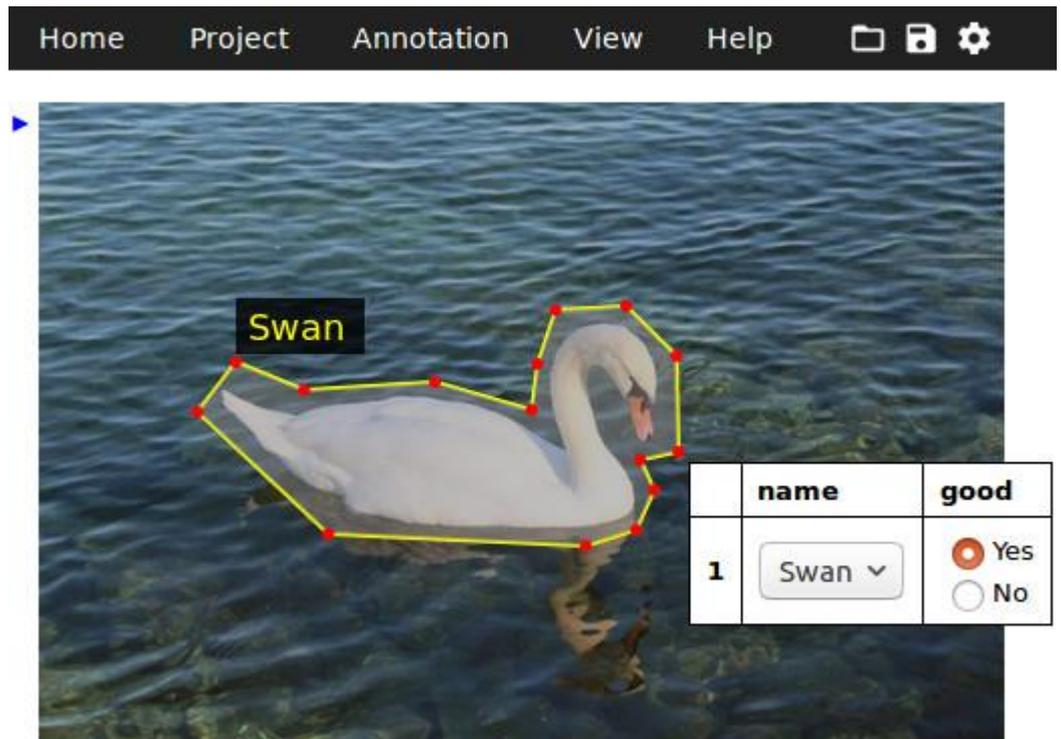


Gambar 2.7 Layer VGG-16

https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjf_NWR-sXmAhXKZSsKHawOD9wQjRx6BAgBEAQ&url=https%3A%2F%2Fneurohive.io%2Fen%2Fpopular-networks%2Fvgg16%2F&psig=AOvVaw2jG2m6dNQgjkZ_Pg3APJT&ust=1576990655391676

2.14 VGG Image Annotations (VIA)

VGG Image Annotations (VIA) merupakan sebuah program pendukung untuk melakukan anotasi pada gambar, ataupun video. VGG Image Annotations (VIA) dapat digunakan hanya dengan menggunakan web browser untuk membuat anotasi pada gambar, dan memberikan output data dengan format JSON. VIA dapat dilihat lebih jelas pada Gambar 2.8



Gambar 2.8 Screenshot VIA

https://www.robots.ox.ac.uk/~vgg/software/via/images/via_demo_screenshot2_via-2.0.2.jpg

2.15 CNN

CNN (Convolutional Neural Network) merupakan sebuah model yang dirancang khusus untuk melakukan proses data matrix, seperti data 1 dimensi yang berhubungan dengan waktu, atau matrix 2 dimensi, seperti pada gambar yang memiliki 2 dimensi, yaitu *width* dan *height*. *Layer* yang terdapat pada *CNN* terdiri atas, *Convolution Layer*, *Activation Layer*, *Pooling Layer*, dan *FCN (Fully Connected Layer)*.

Pada tahap *Convolution Layer*, data yang di berikan akan di proses dan di lakukan penerapan filter seperti pada Gambar 2.9, dimana filter tersebut akan

diterapkan di atas data asli yang disediakan, sehingga menghasilkan output seperti pada Gambar 2.10, atau dapat disebut sebagai *Feature Map*

1	0	1
0	1	0
1	0	1

3 x 3
filter

Gambar 2.9 Filter Convolutional Layer

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

Image

4	3	4
2	4	3
2	3	4

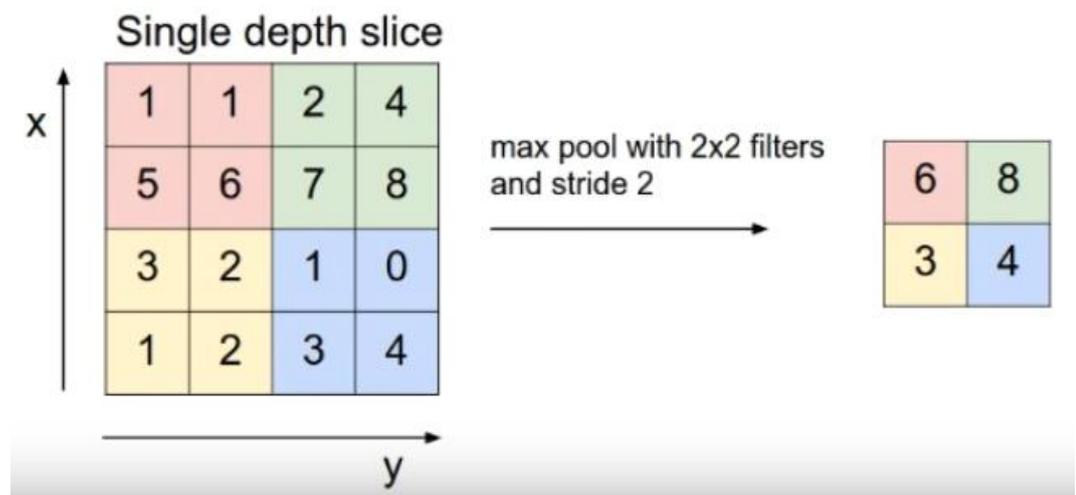
Convolved
Feature

Gambar 2.10 Output Convolutional Layer

sumber: https://miro.medium.com/max/526/1*GcI7G-JLAQiEoCON7xFbhg.gif

Selanjutnya, output yang dihasilkan akan melewati *Activation Layer*, dimana *layer* ini berfungsi untuk mengubah output dari *Convolutional Layer* menjadi output final yang telah melewati fungsi non linear.

Pooling layer, menerima input dari layer sebelumnya, dan melakukan *down sampling* atau pengurangan data yang tidak diperlukan. Contoh *Pooling* dapat di lihat pada Gambar 2.11



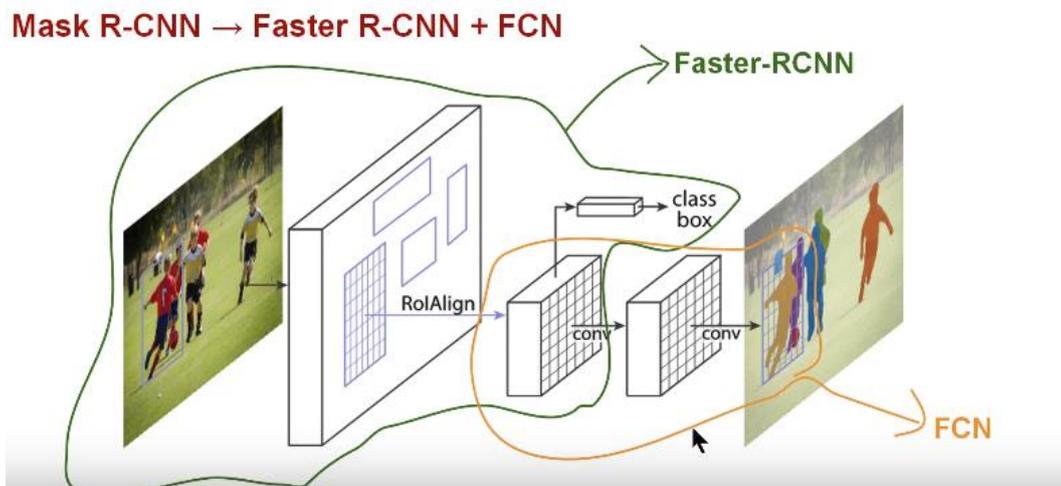
Gambar 2.11 Pooling Layer
 sumber: <https://qph.fs.quoracdn.net/main-qimg-8afedfb2f82f279781bfefa269bc6a90>

Langkah terakhir, adalah *Fully Connected Layer (FCN)*, dimana data yang dihasilkan akan diproses untuk menghasilkan beberapa kombinasi dari *feature* yang sudah didapatkan, sehingga dapat dihasilkan sebuah output probabilitas kesamaan dengan sebuah objek.

2.16 Mask R-CNN

Mask R-CNN merupakan adaptasi dari *Faster R-CNN*, dimana *Mask R-CNN* menambahkan fitur untuk melakukan prediksi *mask* dari sebuah objek bersamaan

dengan prediksi *bounding box* dari sebuah objek, sehingga dapat menghasilkan segmentasi dengan presisi yang tinggi. Secara garis besar, alur kerja *Mask R-CNN* dapat dilihat pada Gambar 2.12



Gambar 2.12 Framework Mask R-CNN

[https://cdn-images-](https://cdn-images-1.medium.com/max/800/1*6MHxZVujW2W5khpQKCCDUw.png)

[1.medium.com/max/800/1*6MHxZVujW2W5khpQKCCDUw.png](https://cdn-images-1.medium.com/max/800/1*6MHxZVujW2W5khpQKCCDUw.png)

2.17 Epoch

Epoch adalah sebuah ukuran dimana, seluruh dataset, melewati seluruh layer yang ada di dalam sebuah *neural network*. *Epoch* ada, dikarenakan, program tidak dapat langsung melakukan *training* keseluruhan dataset yang ada, karena faktor seperti limitasi memori yang dimiliki oleh komputer, atau limitasi lain.

Di dalam sebuah *epoch*, terdapat *batch* atau sekumpulan data yang akan di *input* kedalam *layer* dari sebuah *neural network*.

Sehingga, untuk memberikan gambaran yang lebih jelas mengenai hal ini, sebagai contoh, jika kita memiliki 5000 data dalam sebuah dataset, maka jika memiliki *batch* dengan besaran 250 untuk setiap batch, dan mengharapkan untuk

melakukan 5 *epoch*, maka pada setiap *epoch*, akan didapatkan 20 *batch*. Setiap *batch* akan melewati *layer* yang ada, sehingga akan menghasilkan 20 iterasi untuk menyelesaikan 1 *epoch*.

Dan karena ingin untuk melakukan 5 *epoch*, maka total iterasi yang harus dilakukan untuk menyelesaikan sebuah *training* adalah 100 ($20 * 5$) iterasi untuk menyelesaikan sebuah keseluruhan *training*, yaitu 5 buah *epoch*.

2.18 *Loss Value*

Loss Value merupakan sebuah nilai yang dapat merepresentasikan seberapa baik atau buruk sebuah *model* dalam melakukan evaluasi terhadap data yang diberikan. Idealnya, semakin banyak iterasi *training* yang dilakukan, semakin kecil nilai dari *Loss Value* yang dihasilkan oleh sebuah *model*.

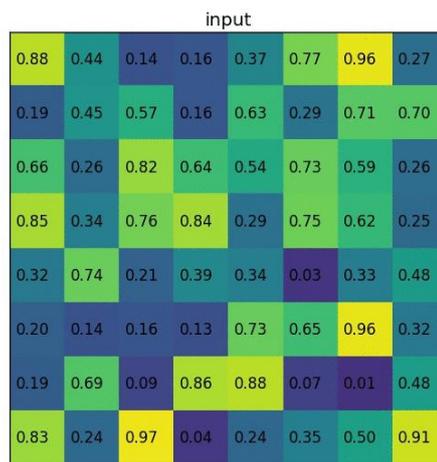
2.19 *Train Time*

Train Time merupakan lamanya waktu yang diperlukan untuk melakukan satu *Epoch*. Semakin banyak *Epoch* yang diperlukan, dan semakin banyak iterasi yang dipeflukan, maka semakin lama pula waktu yang dibutuhkan untuk melakukan *training* dari sebuah *model*.

2.20 *ROI Pooling*

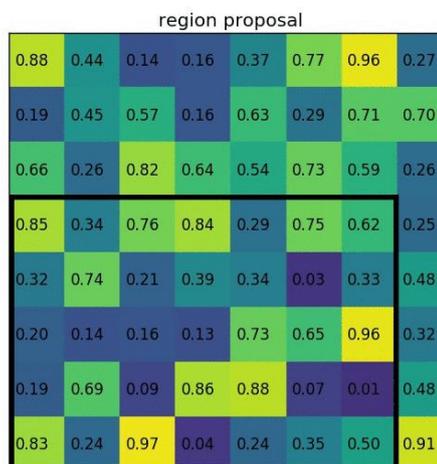
ROI Pooling (*Region of Interest Pooling*) merupakan sebuah metode yang digunakan oleh *Faster R-CNN* (landasan bagi *Mask R-CNN*) untuk melakukan pengurangan waktu *training* yang dibutuhkan oleh sebuah *model*, dimana setiap *ROI* yang dihasilkan oleh *layer* sebelumnya, di ambil, dan di cocokan dengan *feature map* yang juga di hasilkan oleh hasilkan oleh *layer* sebelumnya, dan

melakukan *scaling* dengan besaran yang sudah di tentukan sebelumnya. (7x7px atau 4x4px). *Scaling* dilakukan dengan cara membagi dimensi dari *ROI* menjadi bagian yang sama, kemudian melakukan pengambilan nilai terbesar dari setiap bagian, sehingga menjadi sebuah kotak, dengan nilai terbesar dari setiap daerah yang telah di bagi. Penjelasan dapat dilihat pada Gambar 2.13



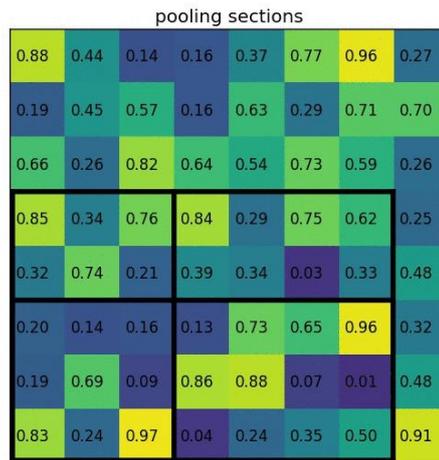
Gambar 2.13 Input image

sumber: <https://cdn-sv1.deepsense.ai/wp-content/uploads/2017/02/1.jpg>



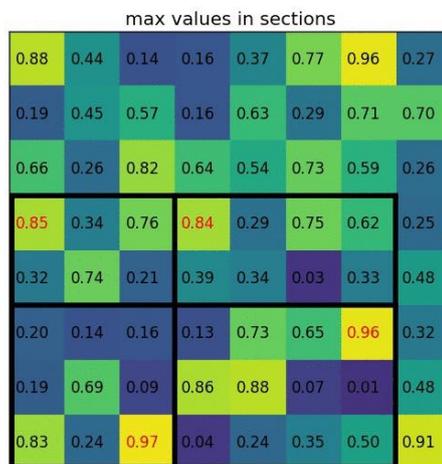
Gambar 2.14 Region Proposal

sumber: <https://cdn-sv1.deepsense.ai/wp-content/uploads/2017/02/2.jpg>



Gambar 2.15 Pooling Sections

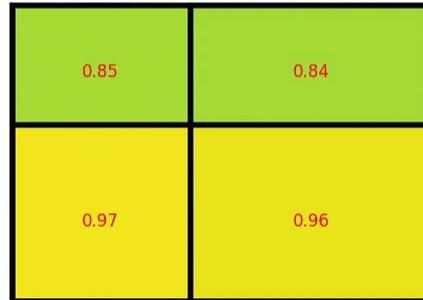
sumber: <https://cdn-sv1.deepsense.ai/wp-content/uploads/2017/02/3.jpg>



Gambar 2.16 Max Valuees in Sections

sumber: <https://cdn-sv1.deepsense.ai/wp-content/uploads/2017/02/4.jpg>

output



Gambar 2.17 Output

sumber: <https://cdn-sv1.deepsense.ai/wp-content/uploads/2017/02/output.jpg>

2.21 ROIAlign

ROIAlign merupakan modifikasi dari *ROI Pooling*, dimana menggunakan *bi-linear interpolation* sebuah metode untuk melakukan estimasi nilai dari 2 buah titik sehingga dapat memberikan hasil yang lebih akurat, dimana tidak dilakukan oleh *ROI Pooling*. Hal ini dilakukan karena presisi lebih di perlukan oleh *Mask R-CNN* dalam melakukan segmentasi pada pixel.

2.22 Pengumpulan Data

Teknik pengumpulan data merupakan sebuah langkah yang sangat penting dalam melakukan sebuah penelitian. Data yang dikumpulkan sangat menentukan hasil akhir yang akan dikeluarkan dalam sebuah penelitian. Untuk penelitian ini, gambar akan diambil dari *InriaHumanPose* (Hinduja, 2016). Gambar diambil dari sumber tersebut, dikarenakan sudah tersedia dalam kualitas yang baik, dan berfokus kepada objek manusia.

2.23 Flowchart

Flowchart merupakan sebuah gambaran grafik atas langkah-langkah yang digunakan sesuai dengan urutan prosedur yang digunakan dalam sebuah program ataupun dalam melakukan sebuah aktifitas. (Subrata, 2019)

2.24 Penelitian Sebelumnya

Tabel 2.1 Tabel Penelitian Terdahulu

1	Nama Penulis	Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick
	Tahun	2018
	Judul	<i>Mask R-CNN</i>
	Metode	Merupakan perpanjangan dari <i>Faster R-CNN</i> , di mana <i>Mask R-CNN</i> menambahkan metode untuk melakukan prediksi sebuah objek bersamaan dengan dilakukannya deteksi dengan melakukan <i>bounding box</i>
Hasil	<i>Mask R-CNN</i> mampu melakukan deteksi terhadap objek yang telah di latih sebelumnya	
2	Nama Penulis	Igor Vasiljevic, Ayan Chakrabarti, Gregory Shakhnarovich
	Tahun	2017
	Judul	<i>Examining the Impact of Blur on Recognition by Convolutional Networks</i>

	Metode	Dengan menggabungkan <i>training</i> antara gambar yang memiliki kualitas gambar rendah, yaitu <i>blurred image</i> dengan gambar dengan kualitas baik atau <i>sharp image</i> .
	Hasil	Ditemukan bahwa dengan melakukan <i>training</i> campuran terhadap gambar dengan kualitas yang baik (<i>Sharp Images</i>) dengan gambar yang memiliki kualitas yang rendah (<i>Blured Images</i>) dapat meningkatkan hasil deteksi pada program konvensional
3	Nama Penulis	Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, Andrew Zisserman
	Tahun	2010
	Judul	The PASCAL Visual Object Classes (VOC) Challenge
	Metode	Membandingkan hasil antara deteksi yang dilakukan oleh program dengan <i>ground truth</i> atau data yang benar yang telah di tandai, dengan melakukan perhitungan besaran area yang tumpang tindih antara kedua buah objek. Deteksi di katakan benar (<i>true positif</i>) besaran area yang tumpang tindih antara deteksi dan <i>ground truth</i> harus lebih besar dari 50%. Rumus yang digunakan adalah
	Hasil	Berdasarkan metode tersebut, akurasi dan deteksi dapat di hitung dan melakukan evaluasi sehingga dapat meningkatkan tingkat akurasi dari model yang digunakan

Penelitian ini akan didasarkan kepada penelitian dari jurnal Metode yang digunakan untuk meningkatkan kemampuan deteksi komputer terhadap objek manusia, yaitu dengan melakukan *masking* terhadap objek yang ingin di deteksi (He, Gkioxari, Dollar, & Girshick, 2017), lalu melakukan *training* pada gambar dengan kualitas rendah, dan kualitas tinggi (Vasiljevic , Chakrabarti, & Shakhnarovich, 2017).

Penelitian akan melakukan adopsi dari metode-metode yang dijelaskan oleh jurnal pertama (Hinduja, 2016) dan jurnal kedua (Vasiljevic , Chakrabarti, & Shakhnarovich, 2017).