AI Mastery Course

orbit FUTURE ACADEMY | Skills For Future Jobs

Module 4
Computer Vision

Section
Learning Schedule

01 Learning Objectives

Learning Objectives

Total Pertemuan	12 pertemuan
Total Pertemuan per Minggu	2 kali (Selasa Pagi & Kamis Pagi)
Total Jam per sesi	3.5 jam

Setiap sesi akan terdiri dari dua pengajaran

- *Theory-based learning*
- *Empiric-based learning*

Learning Objectives

Pretest	Pertemuan ke 1
Materi	Pertemuan ke 2 hingga ke 10
Pembuatan dan Diskusi Mini Projek	Pertemuan ke 11 dan ke 12
Posttest dan Presentasi Mini Projek	Pertemuan ke 12

Mini Projek

- Kelompok projek berjumlah maksimum 10 orang (harus berasal dari kelas asal yang sama)
- Pada pertemuan ke 12, setiap kelompok akan melakukan presentasi dengan durasi 15 menit (presentasi + tanya jawab)

Deskripsi Mini Projek

- https://www.kaggle.com/c/cassava-leaf-disease-classification

02

Learning Outcome

Learning Outcome

L.O 1 **Introduction**	Memahami secara menyeluruh tentang definisi, sejarah, dan aplikasi computer vision
	Memahami trend penggunaan computer vision
L.O 2 **Basic Computer Vision**	Memahami dasar-dasar pengolahan citra dengan pustaka OpenCV seperti load image, write image, dan lain-lain
	Memahami representasi matrix dan ruang warna citra
	Memahami perbaikan citra seperti contrast , brightness, histogram equalization.

Learning Outcome

L.O 3 Advanced Computer Vision (1)	Memahami konsep konvolusi
	Memahami penggunaan deep learning pada computer vision
	Memhami cara kerja *Convolutional Neural Network*
	Memahami cara kerja *object detection* dan *localization*
	Memahami cara kerja *image segmentation*
	Memahami cara kerja *object tracking*

Learning Outcome

L.O 4 Advanced Computer Vision (2)	Memahami cara kerja *transfer learning* menggunakan VGG, AlexNet, dll
	Memahami cara kerja *one-shot recognition* menggunakan *Siamese neural network*
	Memhami cara kerja *image denoising* menggunakan *auto-encoder*

Orbit FUTURE ACADEMY | Skills For Future Jobs

03 Learning Timeline

Learning Timeline

Week 5	**Selasa** • Kontrak pengajaran, Pretest, Pengenalan *Computer Vision* (L.O 1)
	Kamis • *Image reading & writing* (L.O 2), Image enhancement (L.O 2)
Week 6	**Selasa** • Deep Learning & CNN (L.O 3)
	Kamis • Object Detection (L.O 3), Object Localization (L.O 3)
Week 7	**Selasa** • Image Segmentation (L.O 3)
	Kamis • Object Tracking (L.O 3)
Week 8	**Selasa** • Transfer Learning (vgg, alexnet, etc) (L.O 3)
	Kamis • One-shot recognition (L.O 3), Siamese neural network (L.O 3)

Learning Timeline

Week 9	**Selasa** • Image denoising (L.O 3), Autoencoder (L.O 4)
	Kamis • Review dan Focus Group Discussion, **Special Topic**
Week 10	**Selasa** • Pembuatan projek
	Kamis • Presentasi projek, Post Test

TERIMA KASIH

Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media

Orbit Future Academy
@OrbitFutureAcademyIn1
OrbitFutureAcademy
Orbit Future Academy

AI Mastery Course

Module 4
Computer Vision

Section
Introduction of Computer Vision

Profil Coach

YOAN PURBOLINGGA

Coach AI

Enthusiasm in the field of AI, Has an Electrical Engineering Background. research in the field of explorer robot optimization and searching robot. Currently focusing on being a AI Coach at Orbit Future Academy

yoan@orbitfutureacademy.sch.id

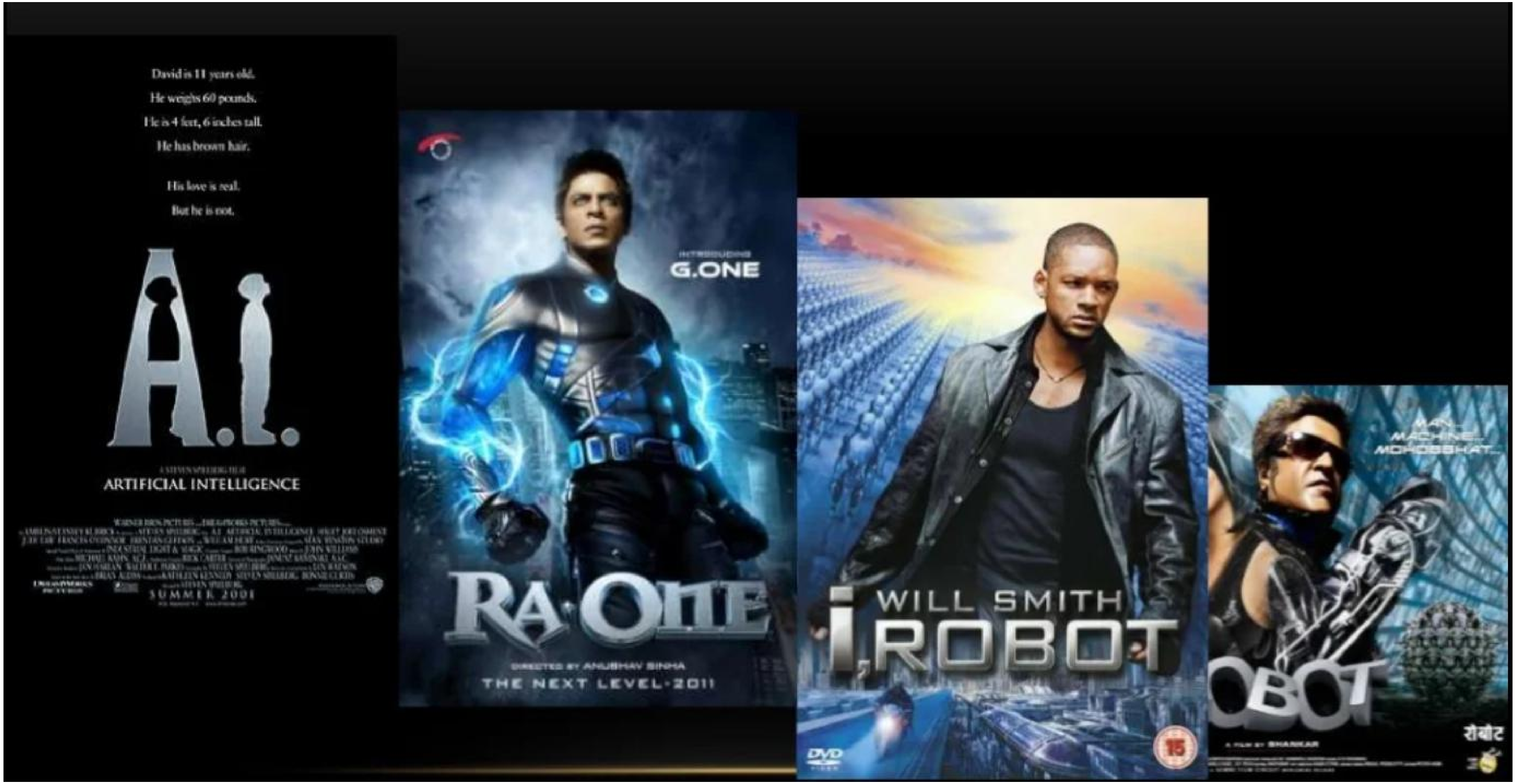
082388669073

1

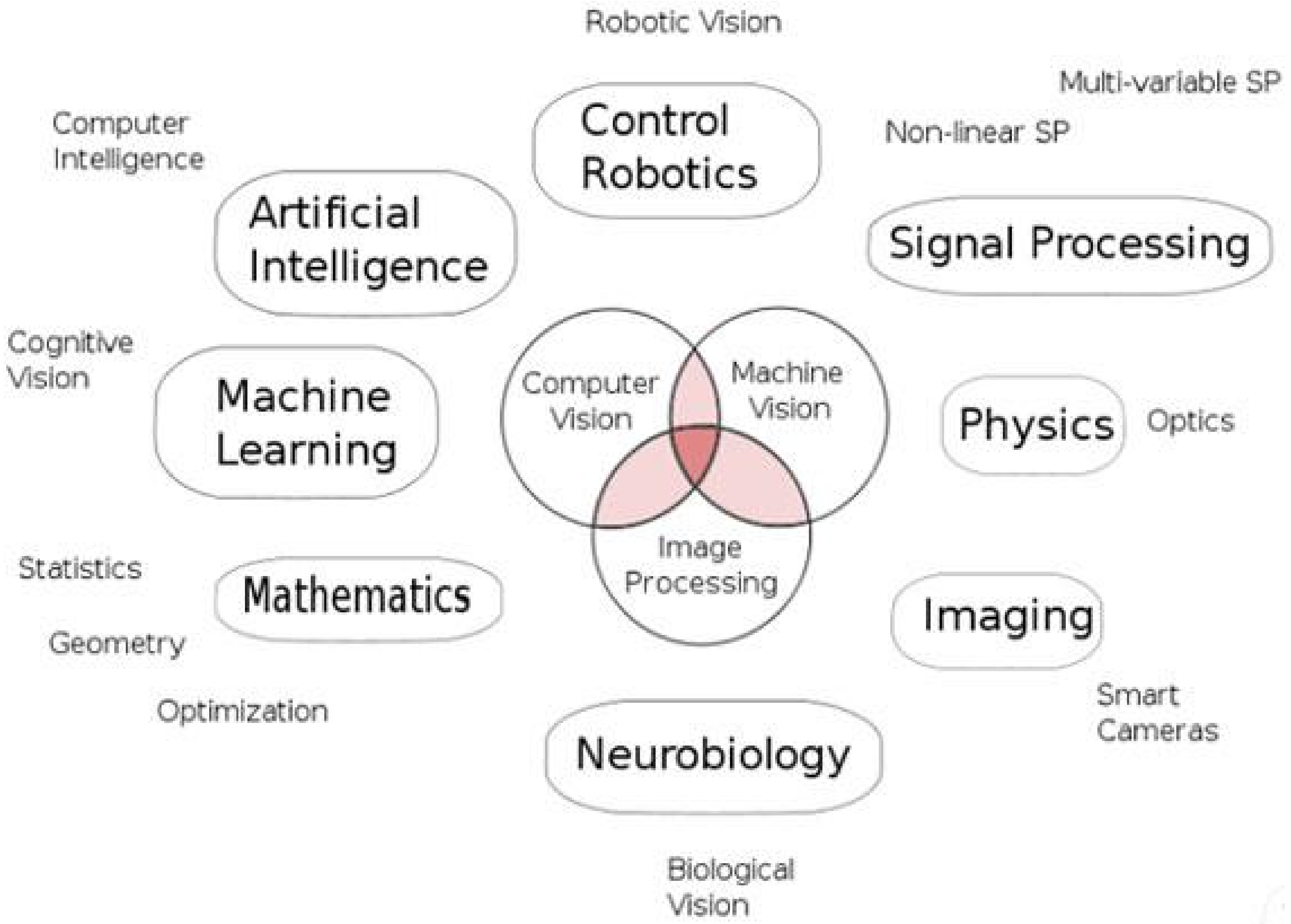
Apa itu Computer Vision ?

Apa itu Computer Vision

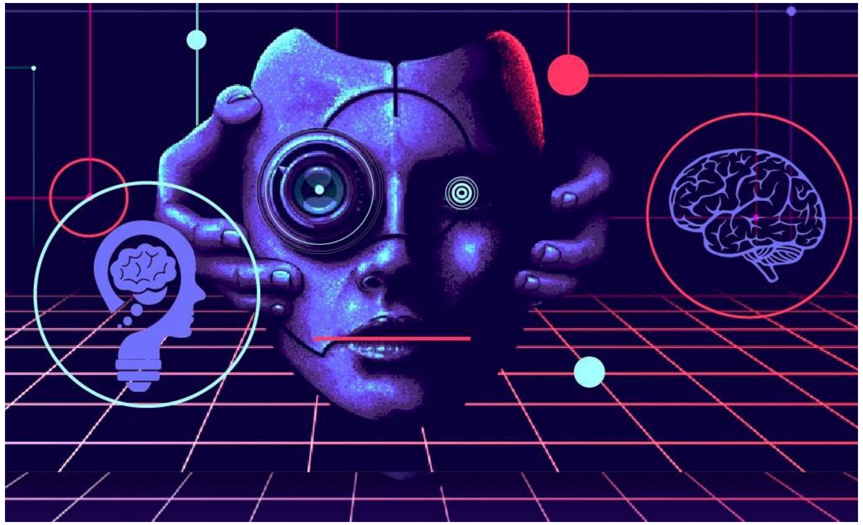
orbit FUTURE ACADEMY | Skills For Future Jobs



Computer Vision



Apa itu Computer Vision



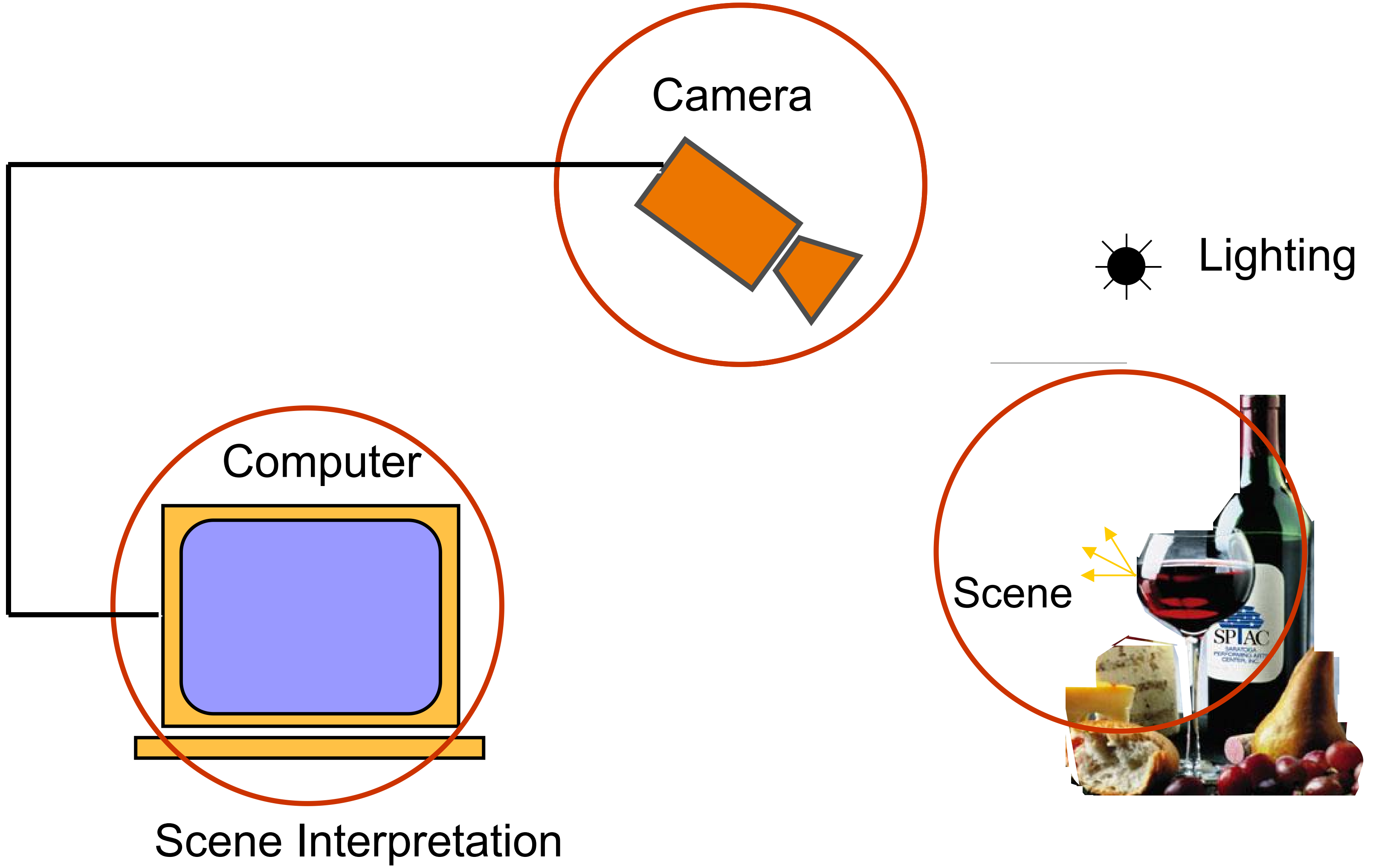
- Computer vision adalah bidang A.I yang digunakan untuk memproses, menganalisa, dan memahami citra (images).
- Duplikat atau tiruan dari *human vision* (penglihatan manusia).
- Data citra berasal dari banyak sumber seperti *video, depth image, medical scanner, satellite sensor,* dan lain-lain.

Computer Vision vs Human Vision

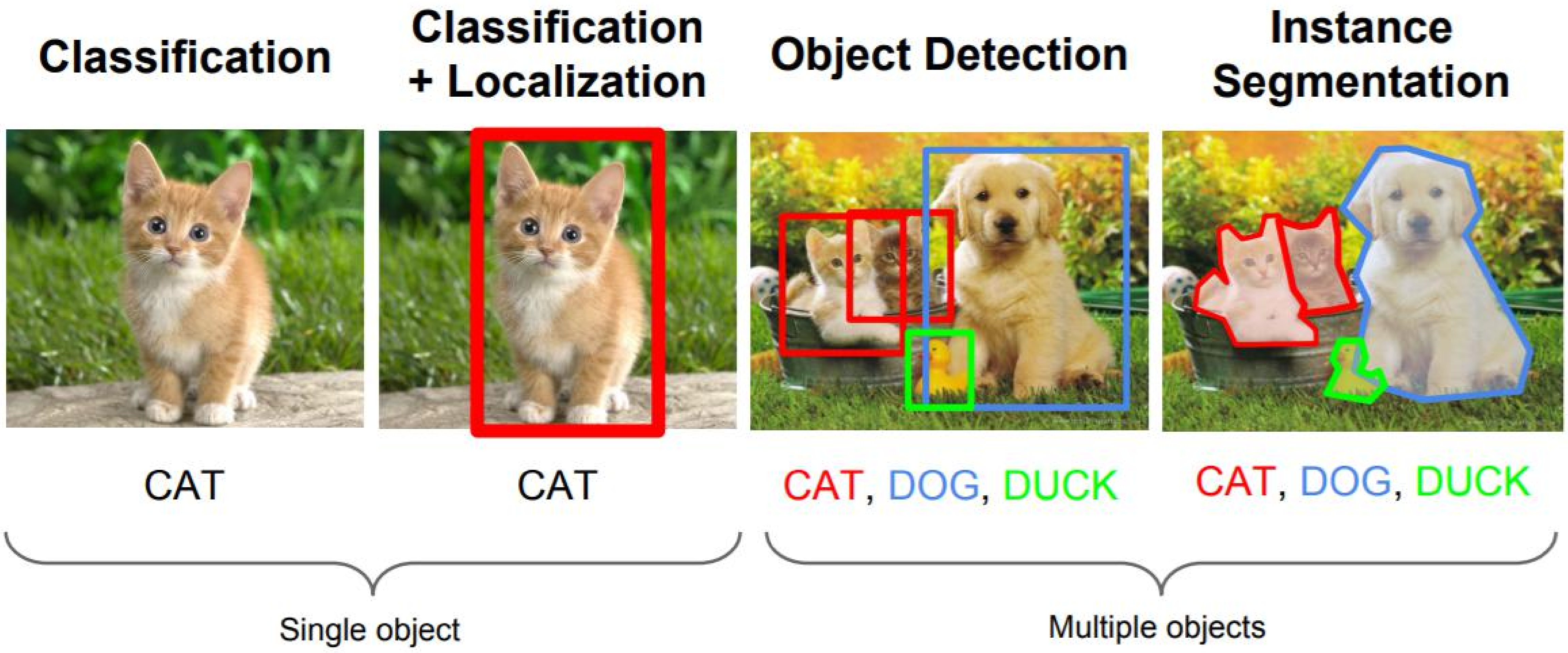
What We See

What Computers See

Components of Computer Vision



Jenis tugas yang dapat dilakukan oleh Computer Vision



A little story about Computer Vision

Marvin Minsky

American cognitive and Computer Scientist

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to "spend the summer linking a camera to a computer and getting the computer to describe what it saw". We now know that the problem is slightly more difficult than that.

A little story about Computer Vision

- **1966** : Minsky assigns computer vision as an undergrad summer project
- **1960's** : Interception of synthetic world
- **1970's** : Some progress of interpreting selected images.
- **1980's** : ANN come and go;
- **1990's** : face recognition, statistical analysis in vogue
- **2000's** : border recognition, video processing start, internet vision
- **2010- now** : Deep Learning era

Computer Vision methods

- Image acquisition
- Pre-processing
- Feature extraction
- Detection / segmentation
- Recognition an interpretation

Mathematic in CV

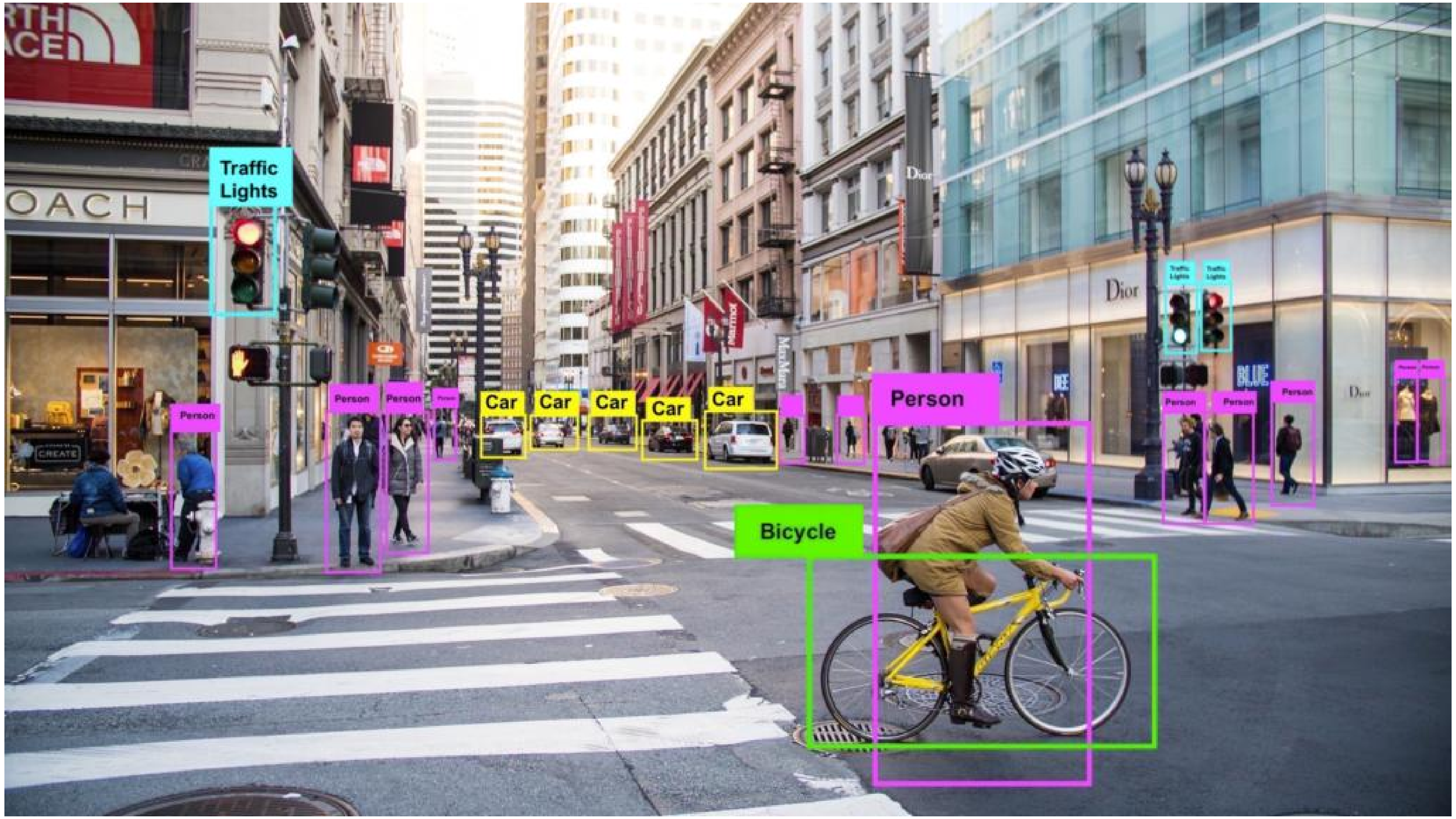
- Kalkulus
- Aljabar Linear
- Probabilitas dan Statistik
- Sinyal Prosessing
- Projektif Geometri
- Komputasional Geometri
- Teori optimasisasi
- Teori kendali

2

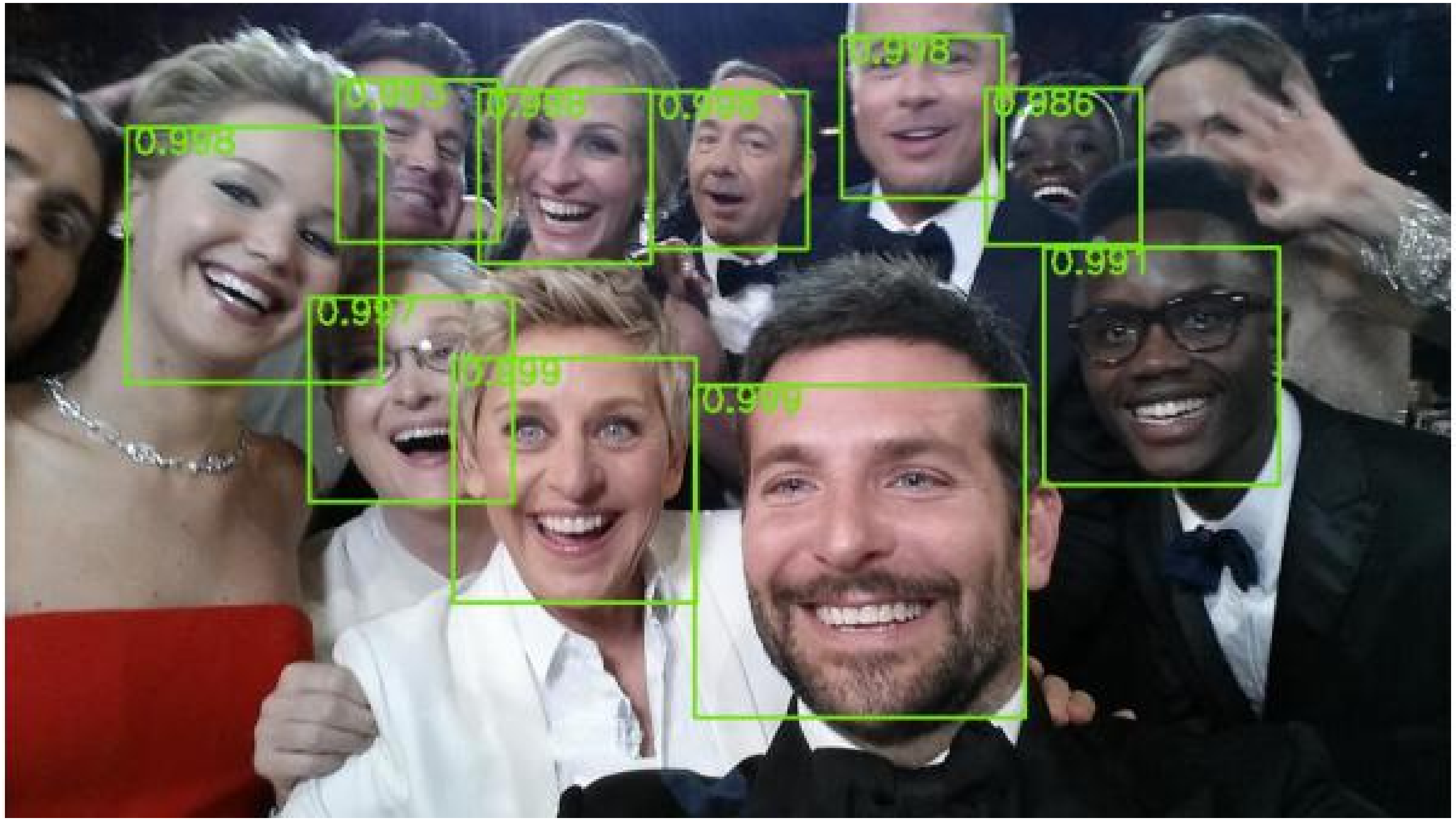
Penerapan Computer Vision

Computer Vision Application

Traffic Monitoring



Face Detection



Plant Detection

TERIMA KASIH

Skills For Future Jobs

Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media

Orbit Future Academy
@OrbitFutureAcademyIn1
OrbitFutureAcademy
Orbit Future Academy

AI Mastery Course

Module 4
Computer Vision

Section
Image Processing

Learning Objective

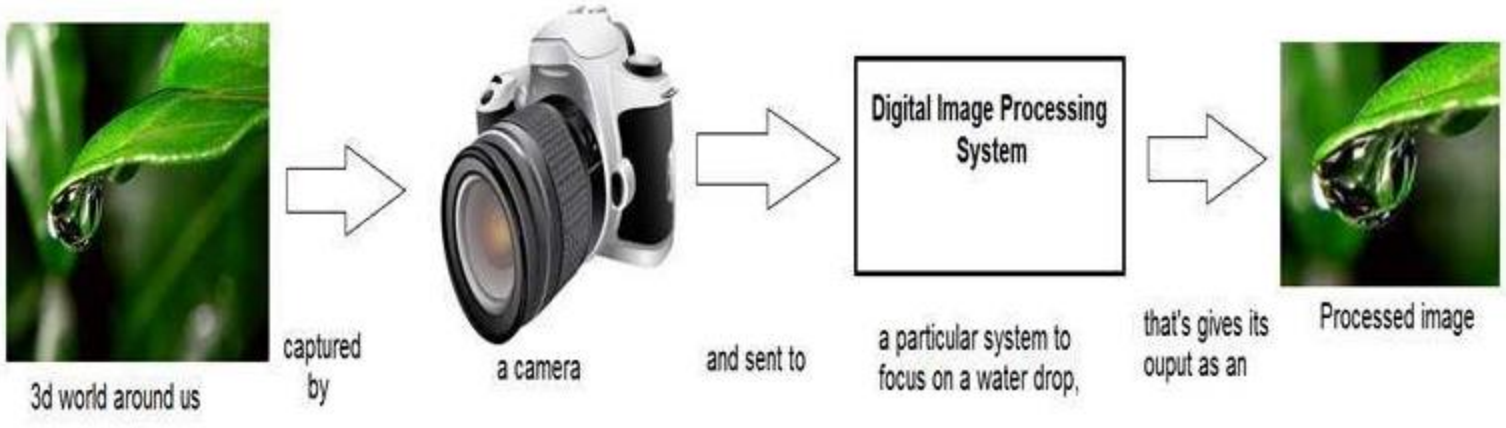
- Memhami prinsip kerja dari pengolahan citra
- Memahami komponen penyusun citra
- Memhami cara kerja *image enhancement*
- Memahami cara kerja *image filtering*
- Memahami cara kerja *edge detection*

01 Introduction

What You Should Know About Image Processing

Pengolahan Citra / Image Processing

Pengolahan citra adalah metode untuk melakukan beberapa operasi pada citra. Tujuannya adalah untuk mendapatkan citra yang disempurnakan atau mengekstrak informasi yang berguna pada citra.



Komponen pada pengolahan citra

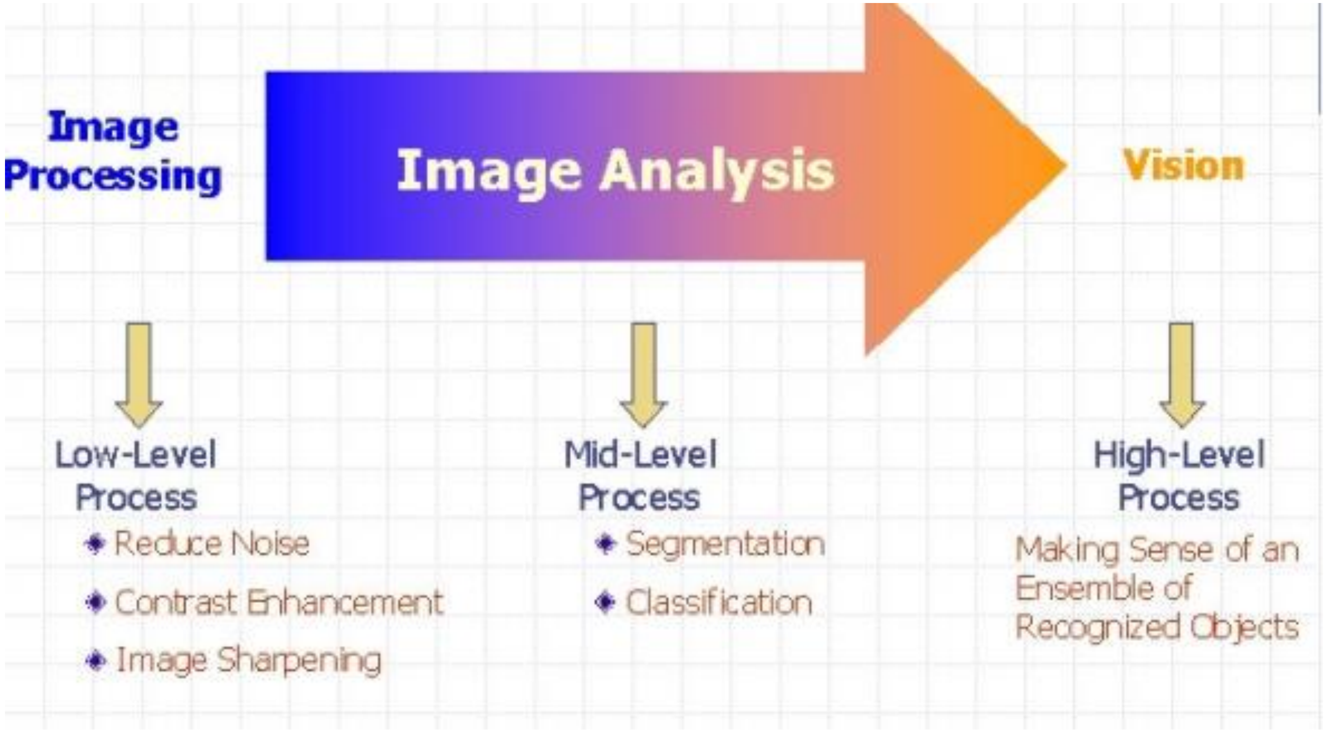
Apa itu Citra (image) ?

Merupakan fungsi dari intensitas cahaya yang direpresentasikan oleh sekumpulan piksel (*picture element*). Citra dibentuk oleh suatau matrix berukuran M x N dimana M merupakan jumlah baris (lebar citra) dan N merupakan jumlah kolom (panjang citra). Setiap piksel mempunyai dua informasi yait koordiant (x,y) dan intensitas f(x,y)

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix}$$

Komponen pada pengolahan citra

Bagaimana kita bisa mengolah citra menjadi informasi ?



Komponen pada pengolahan citra

Jenis-jenis representasi bit pada citra

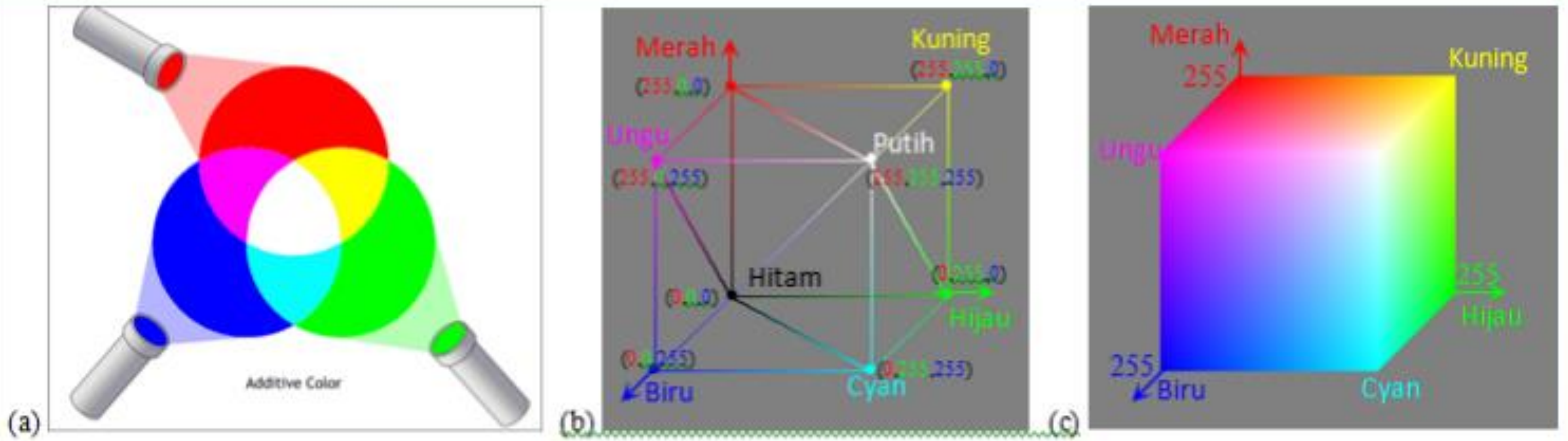
- BINARY IMAGE
- 8 bit COLOR FORMAT
- 16 bit COLOR FORMAT

Komponen pada pengolahan citra

Ruang Warna RGB

Ruang warna standard RGB (Red, Green, Blue) didasarkan pada hasil akuisisi frekuensi warna oleh sensor elektronik.

Setiap komponen warna dikodekan dalam 8 bit sehingga terbentuk 2^24.



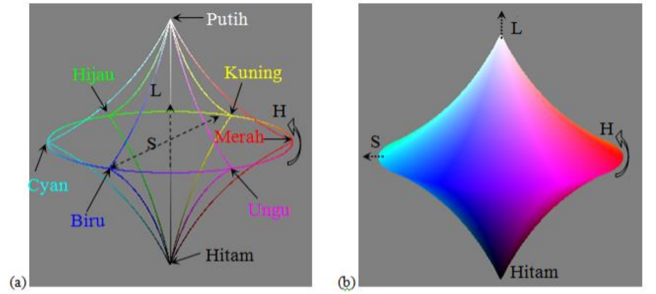
Gambar 3.5. Ruang warna RGB. (a). Gabungan tiga cahaya tampak merah, hijau dan biru, (b) dan (c) representasi nuansa warna dalam koordinat 3-D RGB.
(Sumber gambar 3.5-a: xaraxone.com. gambar b dan c adalah hasil simulasi).

Komponen pada pengolahan citra

Ruang Warna HSL

Mengacu pada hue, saturation, dan lightness. HSL merepresentasikan nuansa warna dalam koordinat silinder 3-D. Hal ini lebih mendekati intuisi dan persepsi visual manusia.

- Hue merupakan nilai yang merepresentasikan spectrum warna dari cahaya tampak (merah, jingga, kuning, hijau, biru).
- Saturasi merupakan nilai yang menunjukan tingkat kejenuhan atau kemurnian suatu warna
- Ligthness merupakan nilai yang menunjukan tingkat kecerahan warna



Gambar 3.7. Ruang warna HSL. (a). Representasi nuansa warna yang terletak pada 6 hue Masing-masing adalah merah, kuning, hijau, cyan, dan biru serta gray-level. (b). Distribusi semua nuansa warna dalam koordinat silinder 3-D HSL.

Komponen pada pengolahan citra

Display Citra

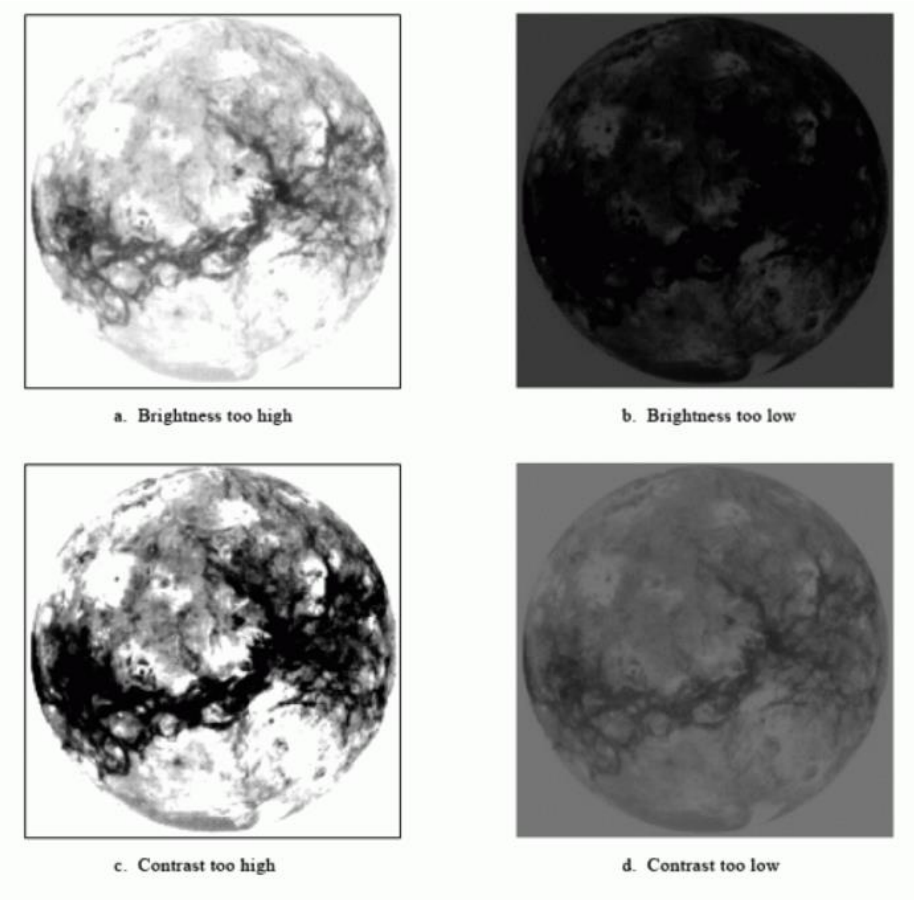
Citra harus memiliki **kecerahan/ brightness** dan **kontras** yang tepat agar mudah dilihat.

- Brightness

Kecerahan/Brightness mengacu pada kecerahan atau kegelapan pada keseluruhan gambar.

- Contrast

Kontras adalah perbedaan **Kecerahan/Brightness** antara objek atau wilayah.



02

Perbaikan Citra / Image Enhancement

Perbaikan citra

Perbaikan kualitas citra ***(image enhancement)*** merupakan salah satu tahapan yang dilakukan dalam pengolahan citra. Tujuan dari perbaikan kualitas citra antara lain adalah:

- Menonjolkan aspek tampilan tertentu agar lebih mudah dipahami atau diinterpretasi oleh penglihatan manusia
- Mereduksi atau menghilangkan aspek tampilan dari suatu citra yang tidak diperlukan misalnya noise/derau

Perbaikan citra

Teknik perbaikan citra dapat dibagi menjadi dua kategori besar:

- **Spatial domain** — peningkatan ruang citra yang membagi citra menjadi piksel-piksel yang seragam sesuai dengan koordinat spasial dengan resolusi tertentu. Metode domain spasial melakukan operasi pada piksel secara langsung.
- **Frequency domain** — peningkatan yang diperoleh dengan mengubahnya kedalam domain spasial. Dalam domain frekuensi, piksel dioperasikan dalam kelompok maupun secara tidak langsung.

Perbaikan citra

Operasi Transformasi Intensitas pada Citra

Transformasi intensitas diterapkan pada citra untuk manipulasi kontras citra. Ini berada dalam domain spasial, yaitu mereka dilakukan secara langsung pada piksel gambar yang ada, sebagai lawan dilakukan pada transformasi Fourier gambar.

Berikut ini adalah transformasi intensitas yang umum digunakan:

- Image Negatives (Linear)
- Log Transformations
- Power-Law (Gamma) Transformations
- Contrast Stretching

Perbaikan citra

Log Transformation

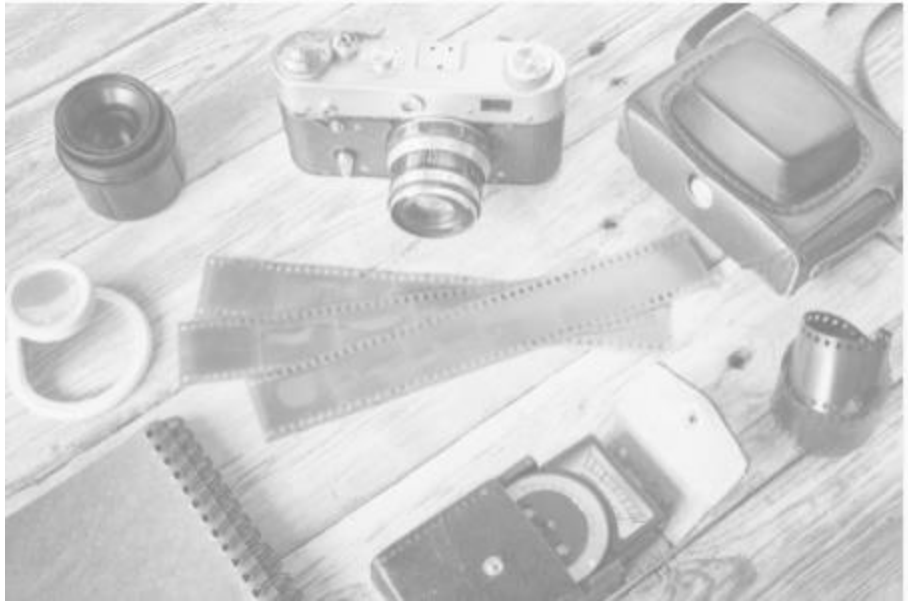
Transformasi log dari suatu gambar berarti mengganti semua nilai piksel, yang ada dalam gambar, dengan nilai logaritmanya. Transformasi log digunakan untuk peningkatan gambar karena memperluas piksel gelap gambar dibandingkan dengan nilai piksel yang lebih tinggi.

```
S = c * log (1 + r)

where,
R = input pixel value,
C = scaling constant and
S = output pixel value
```

```
c = 255 / (log (1 + max_input_pixel_value))
```

Perbaikan citra

Log Transformation



Perbaikan citra

Power law (Gamma) transformation

Power law transformation digunakan untuk menampilkan citra secara akurat di layar komputer. Power law transformation mengontrol kecerahan keseluruhan gambar. Citra yang tidak dikoreksi dengan benar dapat terlihat pucat atau terlalu gelap.

$$O = \left(\frac{I}{255}\right)^{\frac{1}{\gamma}} \cdot 255$$

- I – input pixel value [0, 255].
- O – output pixel value [0, 255].
- γ – gamma that controls image brightness. If gamma < 1 then image will be darker, if gamma > 1 then image will be lighter. A gamma = 1 has no effect.

Perbaikan citra

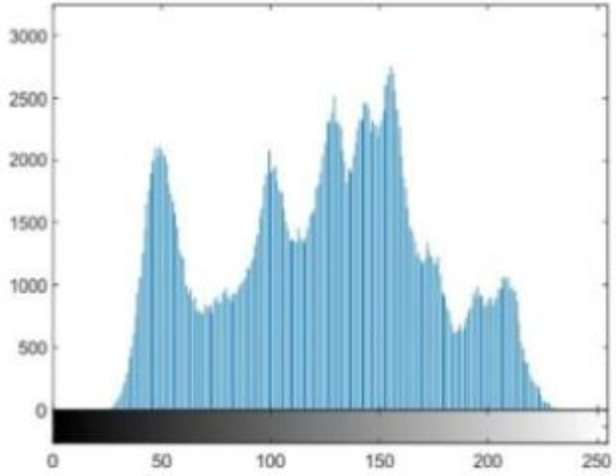
Power law (Gamma) transformation

Perbaikan citra

Histogram Citra

Histogram citra merupakan diagram yang menggambarkan distribusi frekuensi nilai intensitas piksel dalam suatu citra. Sumbu horizontal merupakan nilai intensitas piksel sedangkan sumbu vertikal merupakan frekuensi/jumlah piksel.

(a) Citra Grayscale



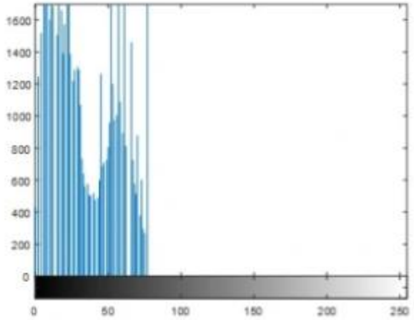
(b) Histogram Citra

Perbaikan citra

Citra Gelap

Citra gelap merupakan citra yang memiliki banyak piksel dengan nilai intensitas mendekati 0. Distribusi nilai intensitas citra gelap cenderung berada pada daerah sebelah kiri histogram.

(a) Citra gelap

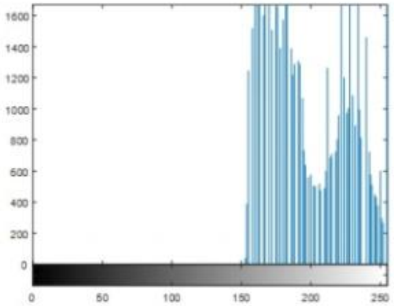


(b) Histogram citra gelap

Citra Terang

Citra terang merupakan citra yang memiliki banyak piksel dengan nilai intensitas mendekati 255. Distribusi nilai intensitas citra terang cenderung berada pada daerah sebelah kanan histogram..

(a) Citra terang



(b) Histogram citra terang

Perbaikan citra

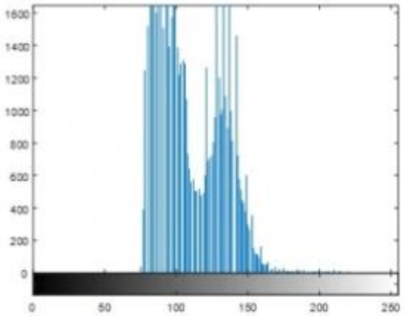
Citra dengan contrast rendah

Citra dengan kontras rendah merupakan citra yang memiliki range nilai intensitas yang sempit. Histogram citra pada Gambar di bawah ini menunjukkan bahwa citra berada pada range nilai intensitas 74-224. Sehingga tidak memiliki nilai intensitas antara 0-74 dan juga 224-255.

Citra dengan kontras tinggi

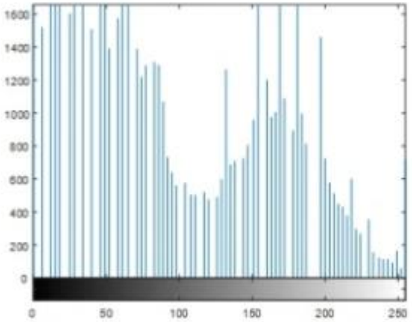
Citra dengan kontras tinggi merupakan citra yang memiliki range nilai intensitas yang lebar. Histogram citra pada Gambar di bawah ini menunjukkan bahwa citra berada pada range nilai intensitas 0-255

(a) Citra dengan kontras rendah



(b) Histogram citra dengan kontras rendah

(a) Citra dengan kontras tinggi



(b) Histogram citra dengan kontras tinggi

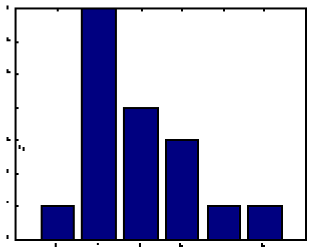
Perbaikan citra

Step step pada Histogram Equalization

1. Hitung histogram dari citra
2. Hitung hasil normalisasi dari histogram
3. Ubah citra masukan menjadi citra keluaran

4	1	3	2
3	1	1	1
0	1	5	2
1	1	2	2

input image



Perbaikan citra

Step step pada Histogram Equalization

1. Hitung histogram dari citra
2. Hitung hasil normalisasi dari histogram
3. Ubah citra masukan menjadi citra keluaran

intensity	sum	normalized sum
0	1	1/16*5=0.31255
1	8	2.5
2	12	3.75
3	14	4.375
4	15	4.6875
5	16	5.0

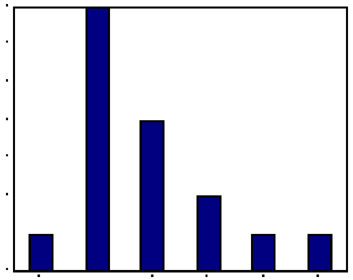
Perbaikan citra

Step step pada Histogram Equalization

1. Hitung histogram dari citra
2. Hitung hasil normalisasi dari histogram
3. Ubah citra masukan menjadi citra keluaran

5	3	4	4
4	3	3	3
1	3	5	4
3	3	4	4

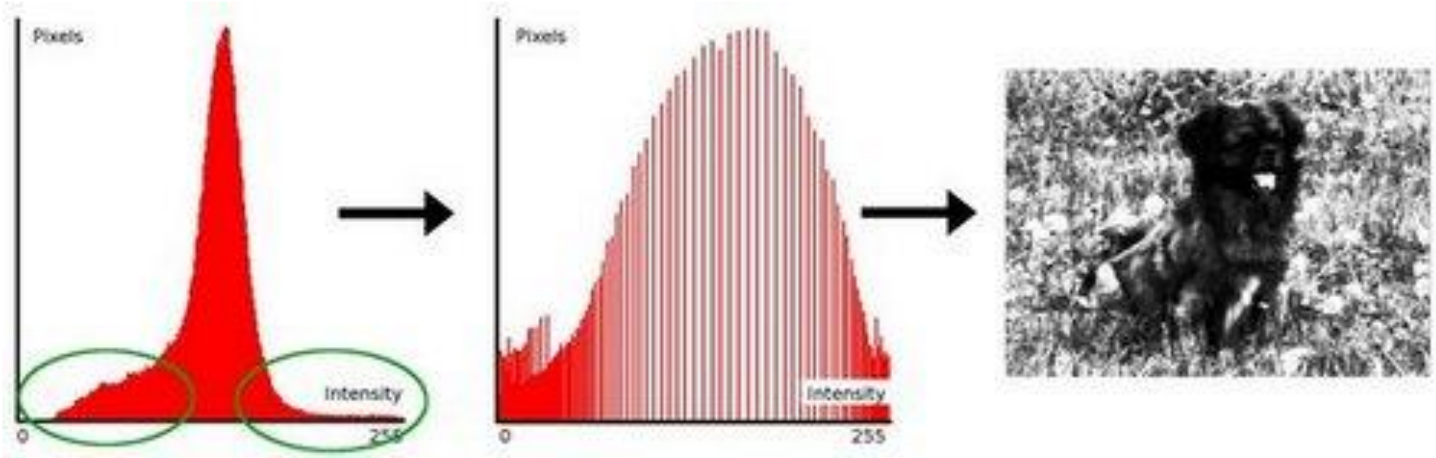
output image



Perbaikan citra

Step step pada Histogram Equalization

1. Hitung histogram dari citra
2. Hitung hasil normalisasi dari histogram
3. Ubah citra masukan menjadi citra keluaran



Orbit Future Academy | Skills For Future Jobs

03 Image Filtering

Image Filtering / Pemfilteran Citra

Image Filtering

Filtering atau penyaringan adalah teknik untuk memodifikasi atau meningkatkan gambar. Misalnya, Anda dapat memfilter gambar untuk menekankan fitur tertentu atau menghapus fitur lainnya. Operasi pemrosesan gambar yang diimplementasikan dengan penyaringan termasuk smoothing, sharpening, dan edge enhancement.

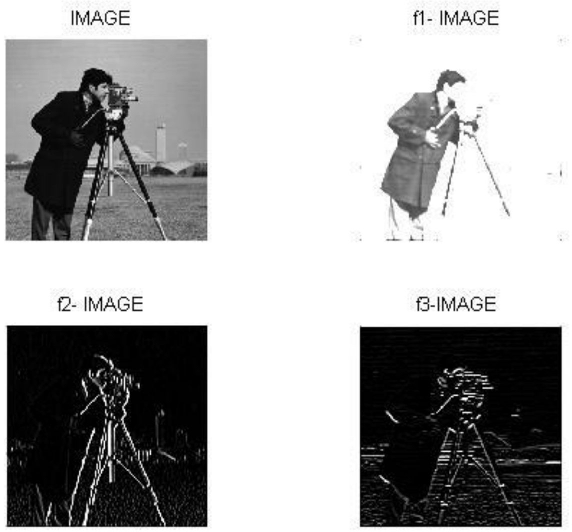


Image Filtering / Pemfilteran Citra

Prinsip Image Filtering

$$Y = H \circledast X$$

Konvolusi

Image Filtering / Pemfilteran Citra

Konvolusi

$$H \circledast X = \sum_y \sum_x H(x, y) \cdot X(T_x - x, T_y - y)$$

(x, y) Posisi filter

(T_x, T_y) Posisi titik yang difilter

Image Filtering / Pemfilteran Citra

Proses Konvolusi

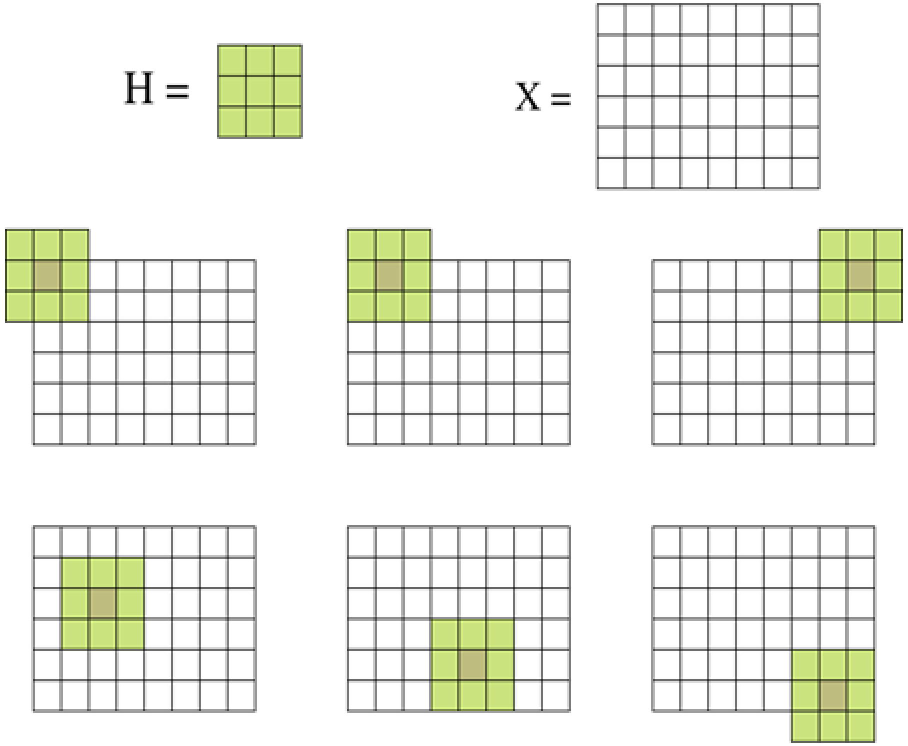


Image Filtering / Pemfilteran Citra

Contoh Konvolusi

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$y(i,j) = \sum_{v=-1}^{1} \sum_{u=-1}^{1} h(u+2, v+2) \cdot x(i+u, j+v)$$

Y(2,3)=H(1,1).X(1,2) + H(1,2).X(1,3) + H(1,3).X(1,4) +
H(2,1).X(2,2) + H(2,2).X(2,3) + H(2,3).X(2,4) +
H(3,1).X(3,2) + H(3,2).X(2,3) + H(3,3).X(3,4)
= (1)(0) + (1)(0) + (1)(0) + (1)(1) + (4)(1) + (1)(0) + (1)(1) + (1)(1) + (1)(0)
= 0 + 0 + 0 + 1+ 4 + 0 + 1 + 1 + 0
= 7

Image Filtering / Pemfilteran Citra

Contoh Konvolusi

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$Y = H \circledast X = \begin{bmatrix} 6 & 4 & 2 & 1 \\ 8 & 10 & 7 & 2 \\ 8 & 10 & 7 & 2 \\ 6 & 4 & 2 & 1 \end{bmatrix}$$

Image Filtering / Pemfilteran Citra

Filter Kernel

Low Pass Filter

$$H = \frac{1}{12}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

High Pass Filter

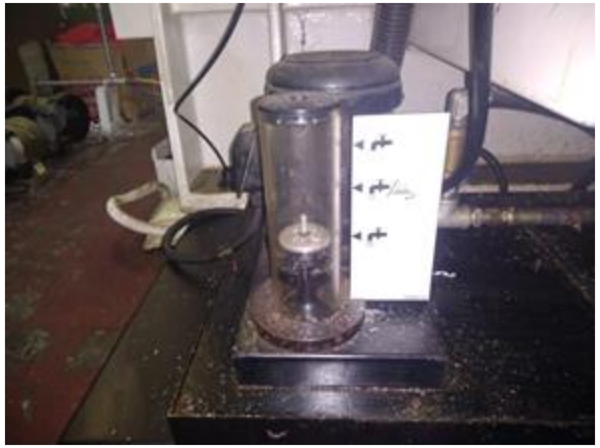
$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 3 \\ -3 & 0 & 1 \end{bmatrix}$$

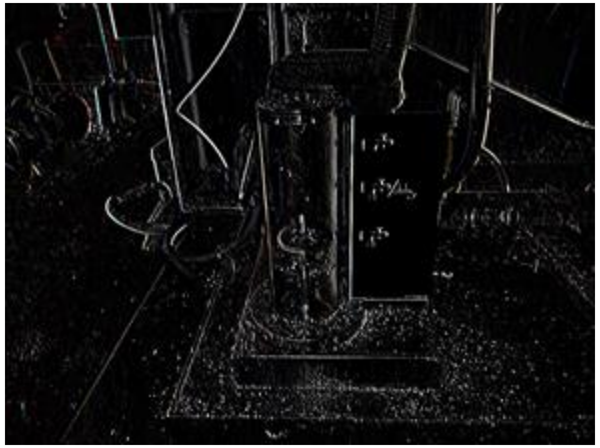
Band Stop Filter

$$H = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 0.5 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

Image Filtering / **Pemfilteran Citra**

Contoh Hasil Filter Kernel





04 Deteksi Tepi

Edge Detection / Deteksi Tepi

Metode Robert

$$\begin{bmatrix}-1\\1\end{bmatrix}$$

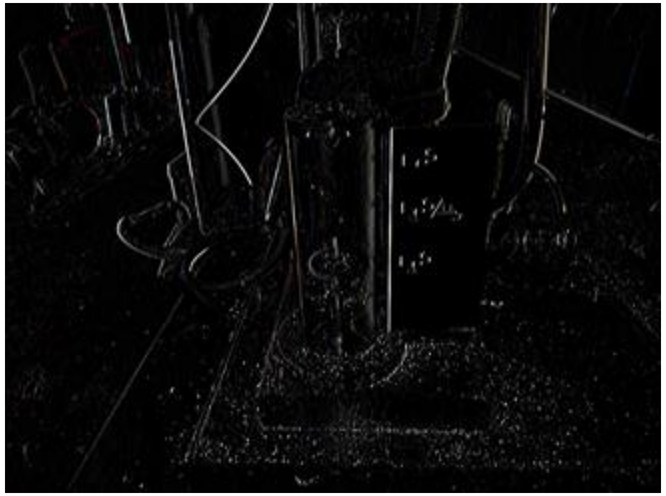
$$\begin{bmatrix}-1 & 1\end{bmatrix}$$

$$\begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}$$

Edge Detection / Deteksi Tepi

Metode Prewit

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

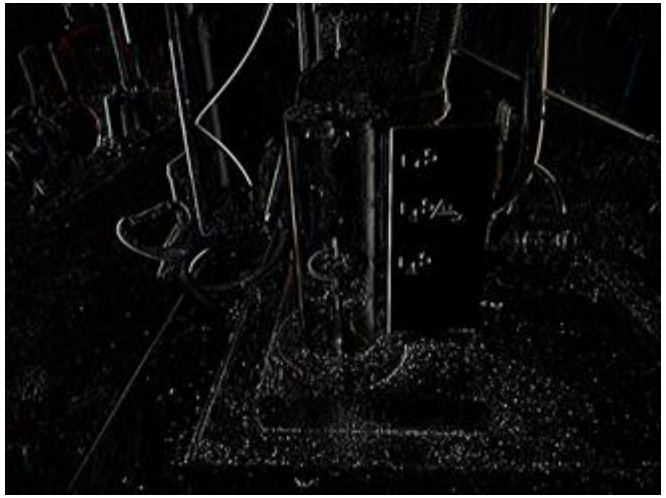


$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

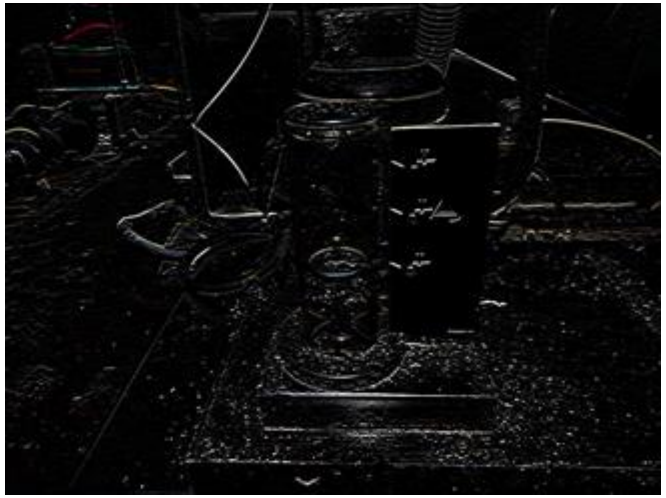
Edge Detection / Deteksi Tepi

Metode Sobel

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



TERIMA KASIH

Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media

Orbit Future Academy
@OrbitFutureAcademyIn1
OrbitFutureAcademy
Orbit Future Academy

Orbit Future Academy — Skills For Future Jobs

AI Mastery Course

Module 4
Computer Vision

Section
Deep Learning & CNN

Learning Objective

- Memahami Deep Learning untuk Computer Vision
- Memahami struktur CNN
- Memahami implementasi CNN menggunakan Keras

1

Introduction of Deep Learning for Image

2

CNN Overview

CNN Overview

- Convolutional Neural Networks (CNNs) adalah kategori Neural Network yang telah terbukti sangat efektif di berbagai bidang seperti pengenalan dan klasifikasi gambar.

- CNN telah berhasil mengidentifikasi wajah, objek, dan rambu lalu lintas lebih lanjut dapat digunakan sebagai robot vision dan self driving car.

- CNN juga digunakan dalam aplikasi smart grid

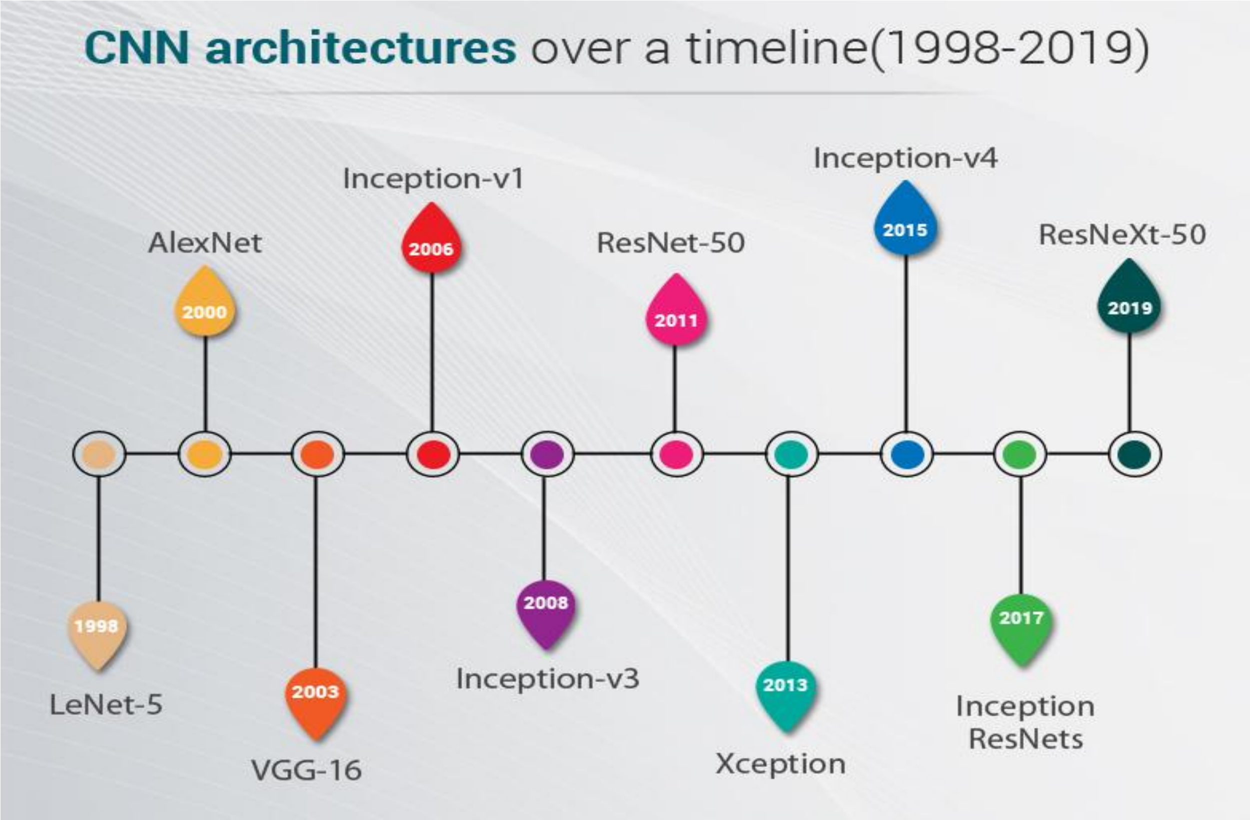
CNN timeline

Yann LeCunn

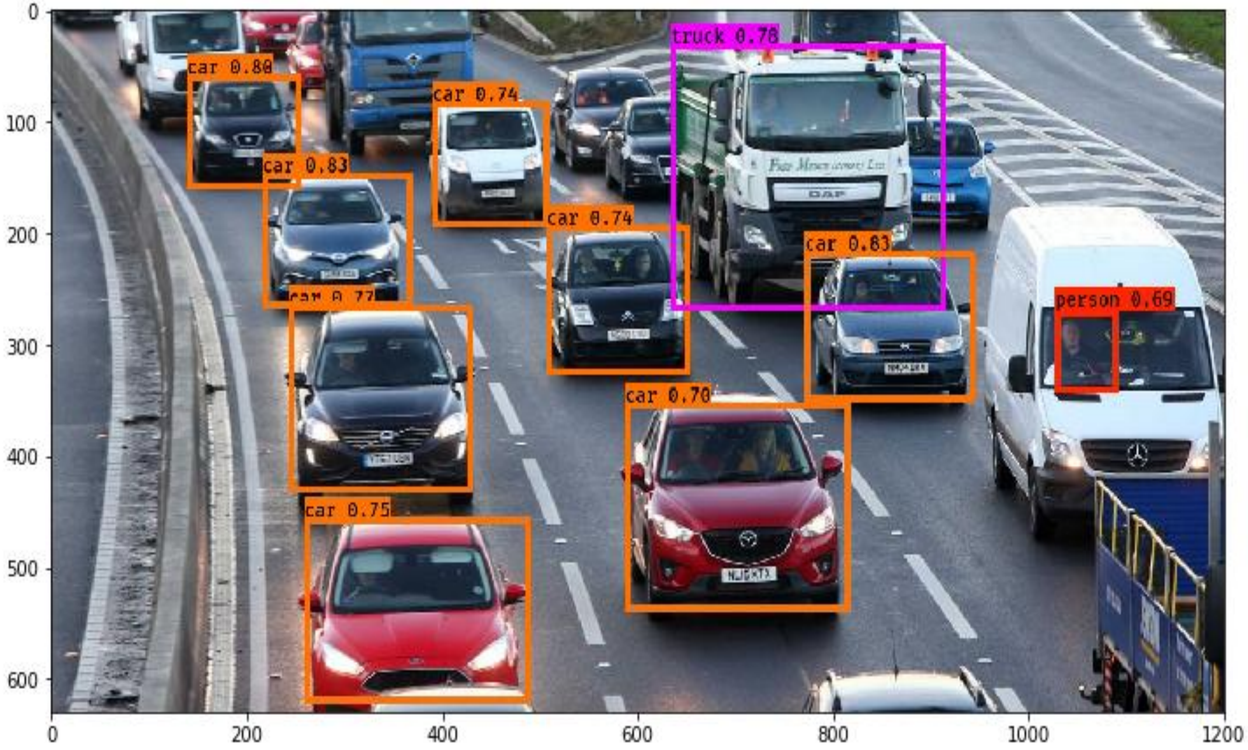
Pionir CNN, Direktor Facebook AI Research Group

Pembuat model CNN pertama yang disebut LeNet pada tahun 1988 yang digunakan untuk mengenali karakter seperti membaca digit, dan zip codes.

CNN timeline



CNN Overview



Cars recognition

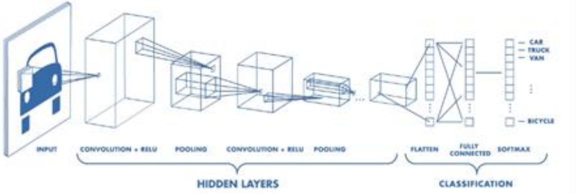
CNN Overview

Self driving car

CNN Overview

Input an image	Process image	Output probability values

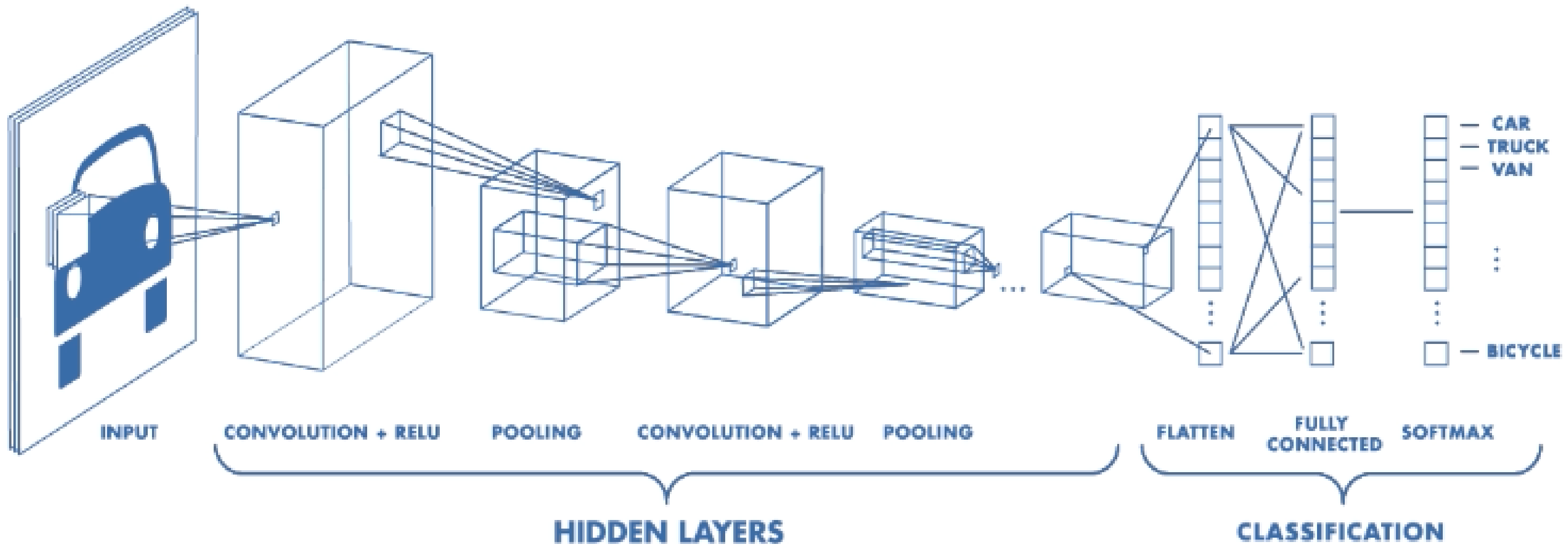
Convolutional Neural Network



Car	70 %
Truck	20 %
Bicycle	10 %

Cara Kerja CNN

Convolutional Layer	Rectified linear unit (ReLU)	Pooling Layer	Fully Connected Layer



CNN vs Neural Network

- Jika menggunakan Artificial Neural Network standar, jumlah neuron yang dibutuhkan untuk memproses data citra akan meningkat secara drastis yang dapat menyebabkan overfitting

- Struktur Convolutional Neural Network memiliki kemiripan dengan Artificial Neural Network

- Topologi CNN memiliki perbedaan dari ANN standar dimana neuron disusun menjadi 3 dimensi

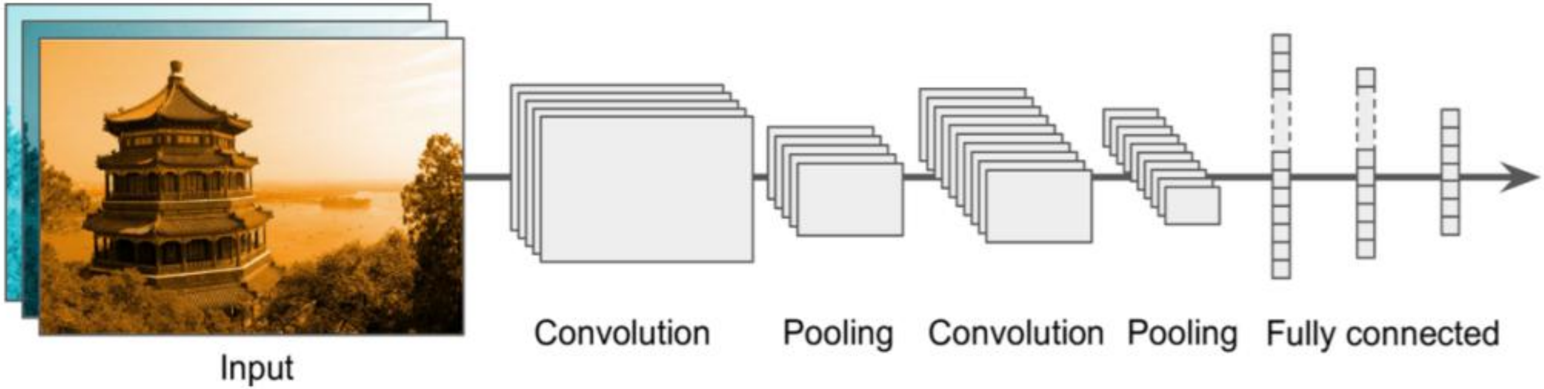
3

CNN Architecture

CNN Architecture

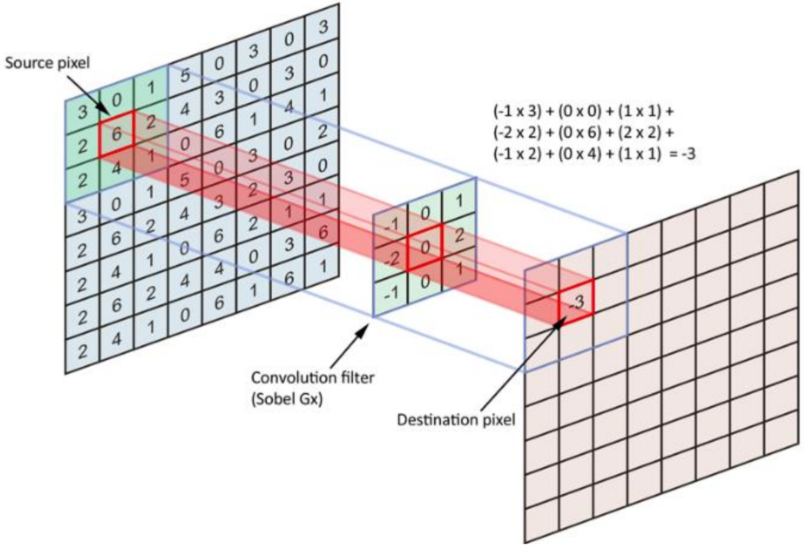
CNN memiliki tiga jenis lapisan (layer) untuk membangun arsitektur selain lapisan input:

- Convolutional Layer (with ReLu activation)
- Pooling Layer
- Fully Connected or Dense Layers

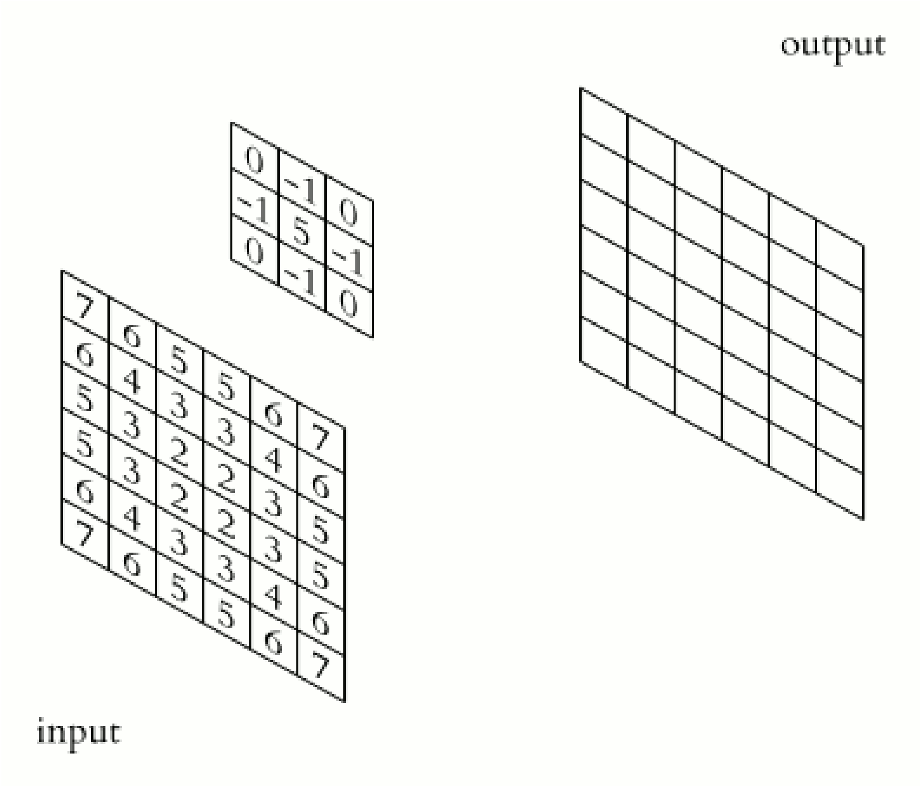


Convolutional Layer

Convolutional Layer: Tujuan utama dari konvolusi adalah untuk mengekstrak fitur dari citra masukan. Konvolusi mempertahankan hubungan spasial antara input dengan mempelajari fitur input.



Convolutional Layer



Convolutional Layer

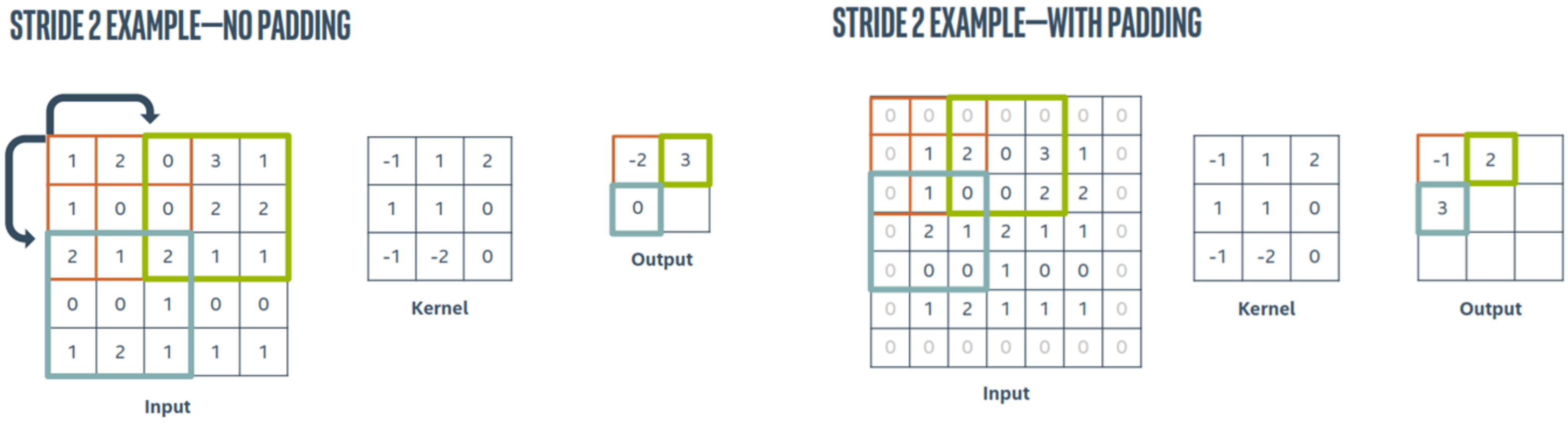
```
tf.keras.layers.Conv2D(
    filters,
    kernel_size,
    strides=(1, 1),
    padding='valid',
    data_format=None,
    dilation_rate=(1, 1),
    groups=1,
    activation=None,
    use_bias=True,
    kernel_initializer='glorot_uniform',
    bias_initializer='zeros',
    kernel_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    bias_constraint=None,
    **kwargs
    )

```

```
tf.keras.layers.Conv2D(
    16,
    (3,3),
    activation='relu'
    )
```

Arguments

- **filters**: Integer, banyaknya jumlah filter.
- **kernel_size**: integer atau 2-tuple integer, bila berisi 2-tuple, bilangan menyatakan height dan width dari jendela konvolusi. Bila hanya berisi satu angka artinya, ukuran height dan widthnya sama.
- **strides**: integer atau 2-tuple integer, menyatakan stride pada konvolusi.
- **padding**: bernilai `"valid"` ata `"same"`. `"valid"` artinya no padding. `"same"` artinya padding dengan zeros akan menghasilkan ukuran yang sama dengan dimensi dari input.
- **activation**: menyatakan fungsi aktivasi yang digunakan. bila tidak dideklarasikan, tidak ada fungsi aktivasi yang digunakan.

Stride dan Padding

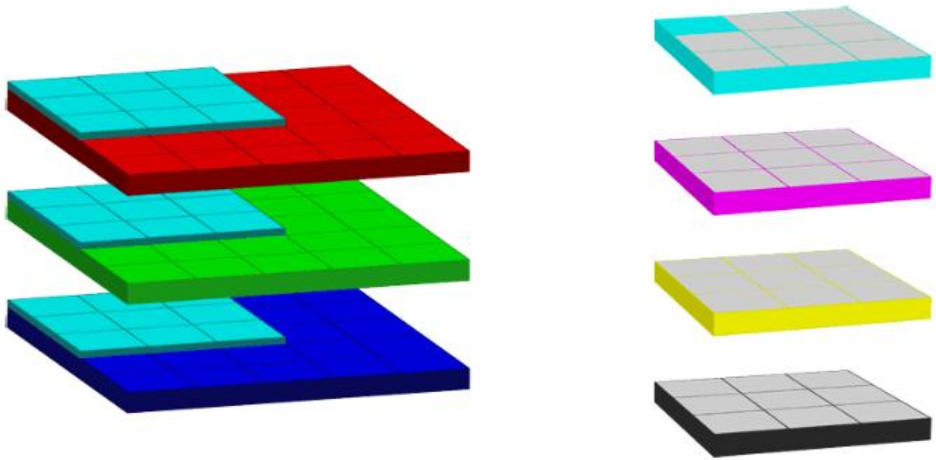


Stride adalah parameter yang menentukan berapa jumlah pergeseran filter.

Padding atau **zero padding** adalah parameter menentukan jumlah pixel (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input

Convolution

CONVOLUTIONS

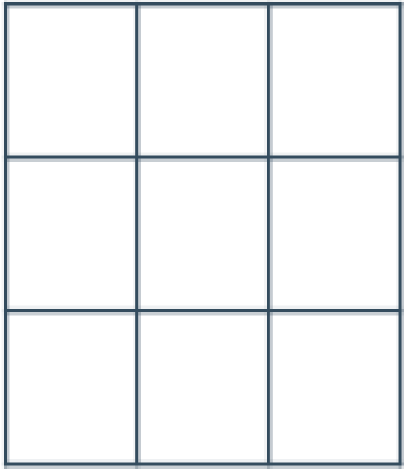


Convolutions Setting :

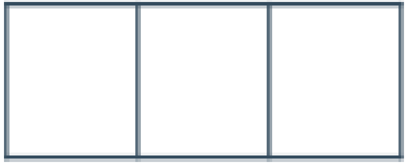
- **Grid size**
- **Padding**
- **Stride**
- **Depth**
- **Pooling**

Grid Size

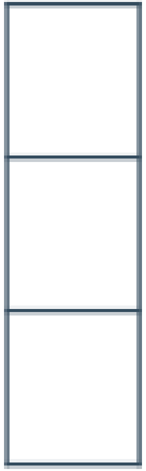
Height: 3, Width: 3



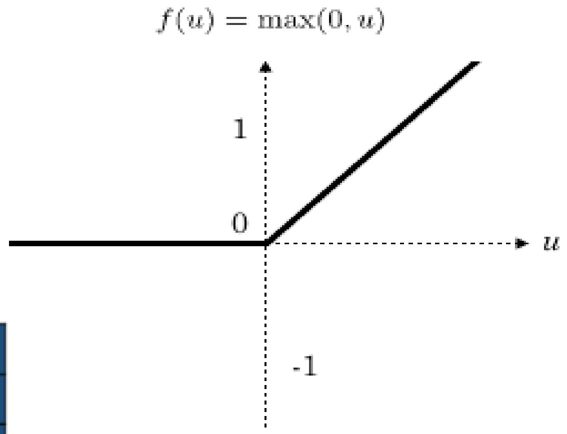
Height: 1, Width: 3



Height: 3, Width: 1

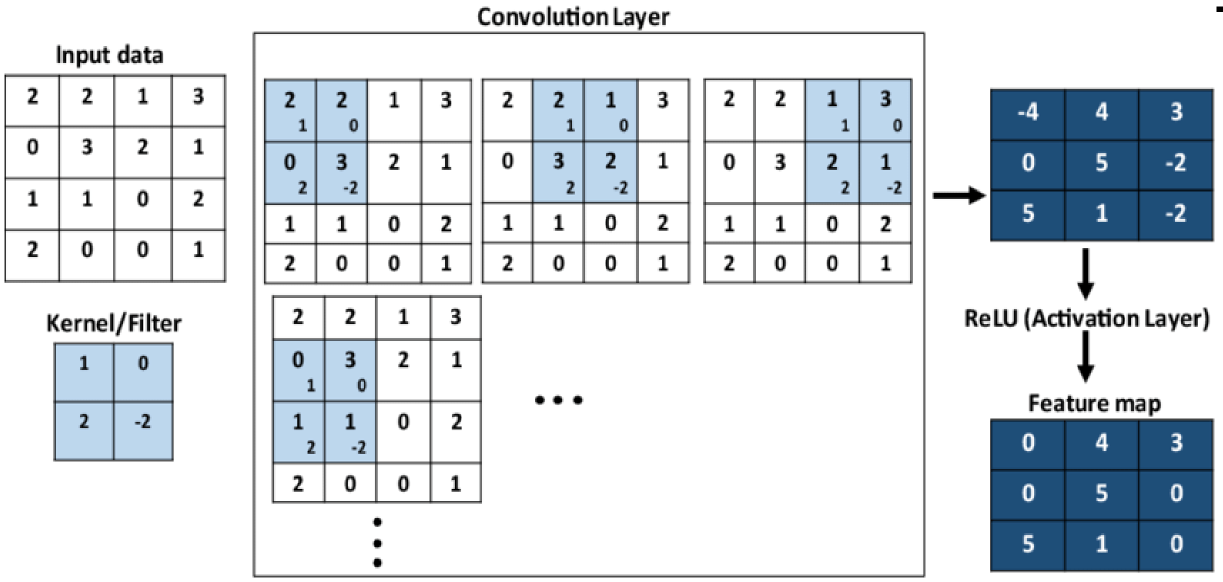


ReLU Layer



ReLU singkatan dari rectified linear unit, dan ini merupakan tipe dari fungsi aktivasi

$f(x) = \max(0, x)$



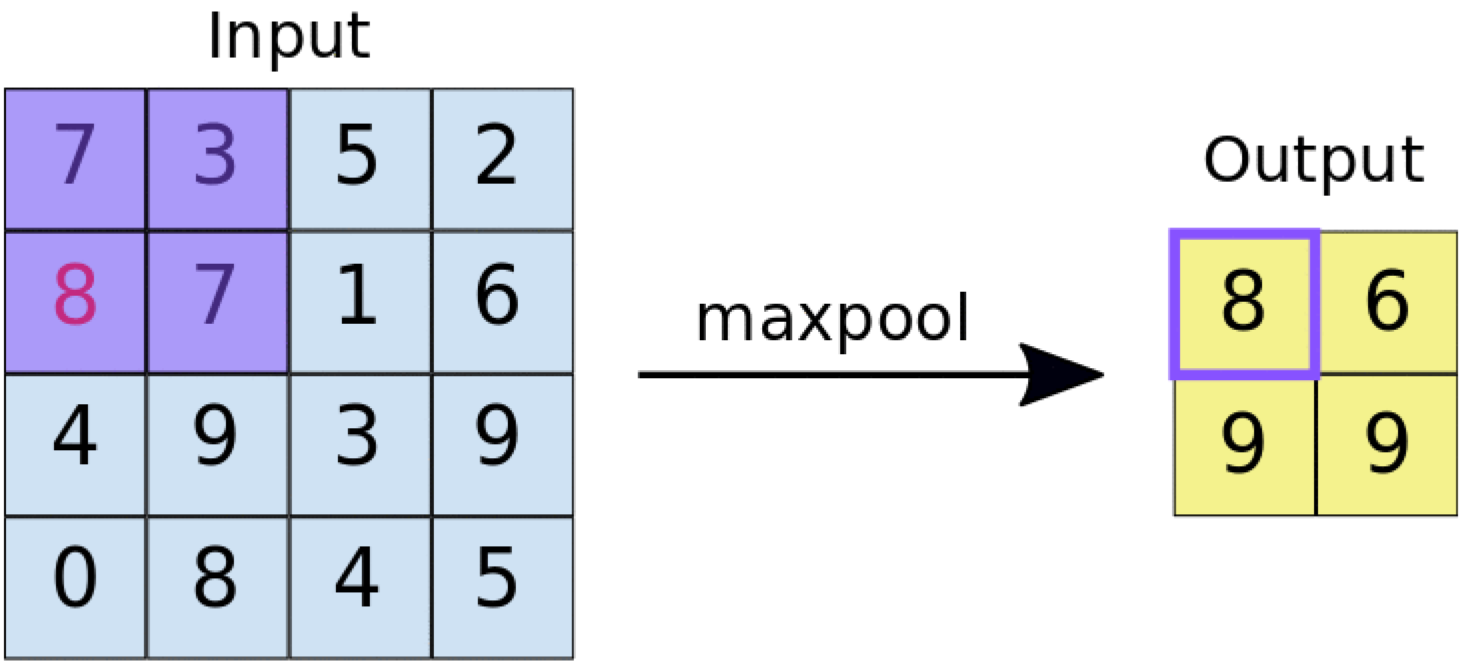
Pooling Layers

- Pooling layers akan mengurangi jumlah parameter ketika input terlalu besar.

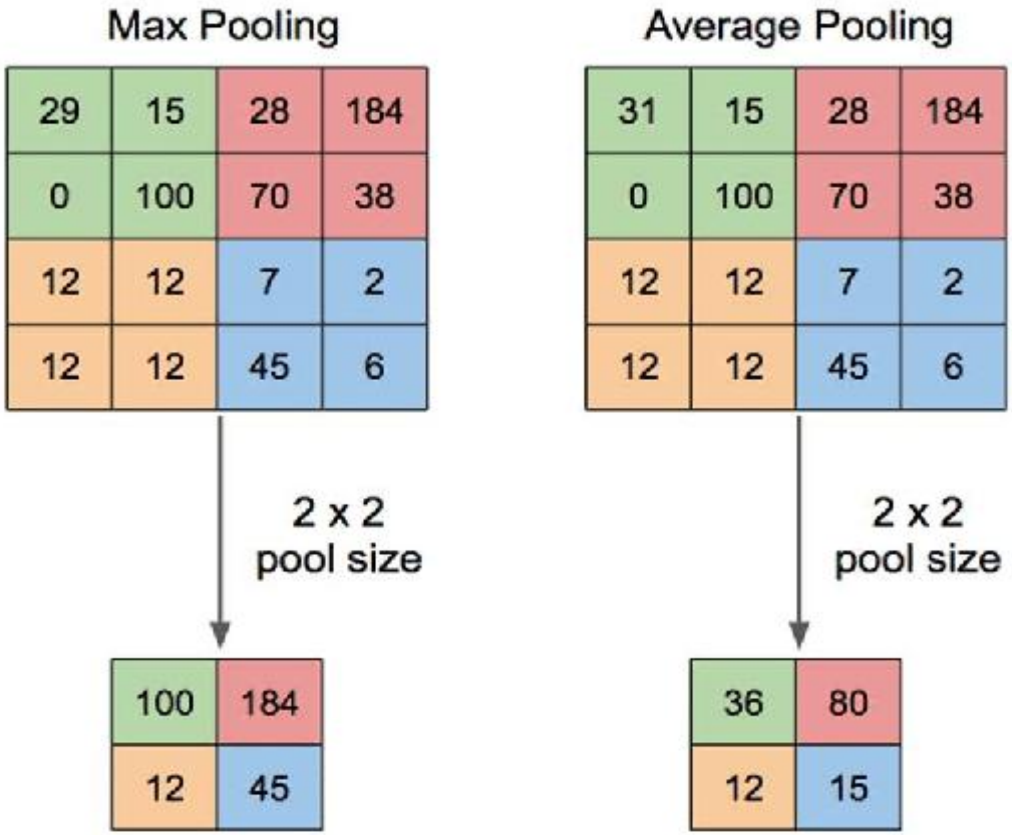
- Pooling juga disebut downsampling yang mengurangi dimensi setiap lapisan tetapi masih menyimpan informasi penting.

- Ada tiga jenis pooling yaitu, Max pooling, Average Pooling, Sum Pooling.

Pooling Layers



Pooling Layers



Pooling Layers

```
tf.keras.layers.MaxPooling2D(
    pool_size=(2, 2),
    strides=None,
    padding="valid",
    data_format=None,
    **kwargs
)
```

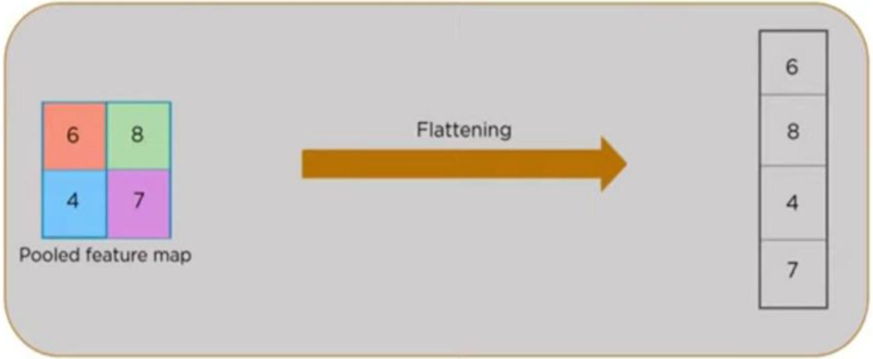
```
tf.keras.layers.MaxPooling2D(2, 2)
```

- **pool_size**: integer atau 2-tuple integer, Ukuran dari jendela yang ingin diambil nilai maksimumnya. `(2, 2)` akan mengambil nilai maksimum dari pooling window berukuran 2.
- **strides**: integer, 2-tuple integer, atau None. menjelaskan seberapa jauh pooling window berpindah. Jika diisi None, secara default akan bernilai `pool_size`.
- **padding**: bernilai `"valid"` atau `"same"`.. `"valid"` artinya no padding. `"same"` artinya padding dengan zeros akan menghasilkan ukuran yang sama dengan dimensi dari input.

Flatten Layers

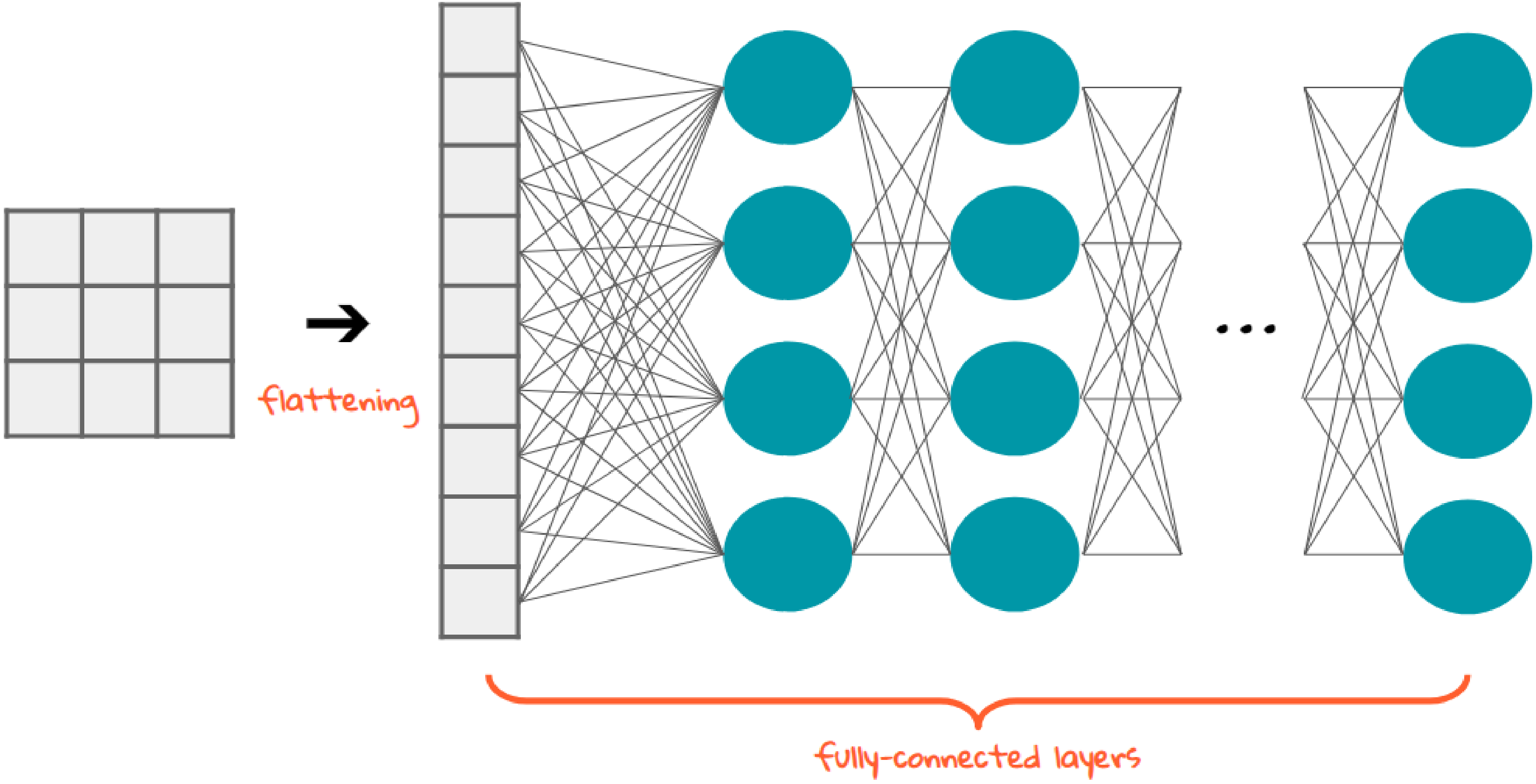
flatten function meratakan **tensor input multidimensi** menjadi satu dimensi, sehingga kita dapat memodelkan lapisan input dan membangun model jaringan saraf, lalu meneruskan data tersebut ke setiap neuron model secara efektif.

The feature matrix akan diubah menjadi vektor (x1, x2, x3, …)

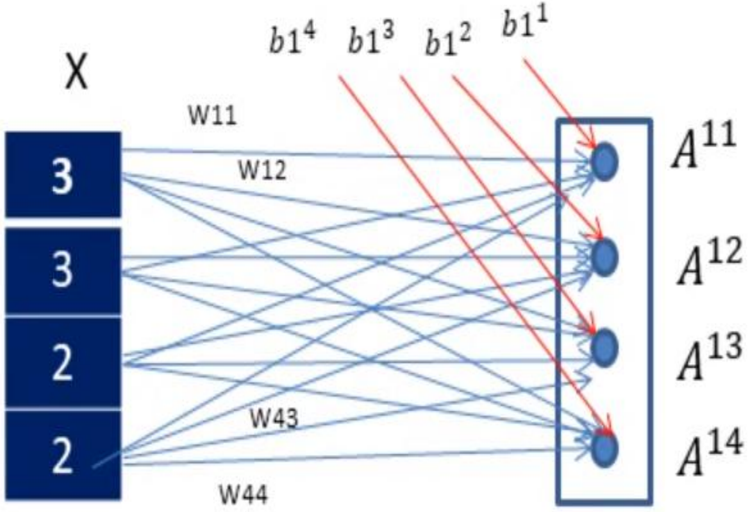



```
tf.keras.layers.Flatten()
```

Flatten Layers



Full connected Layers with ReLU

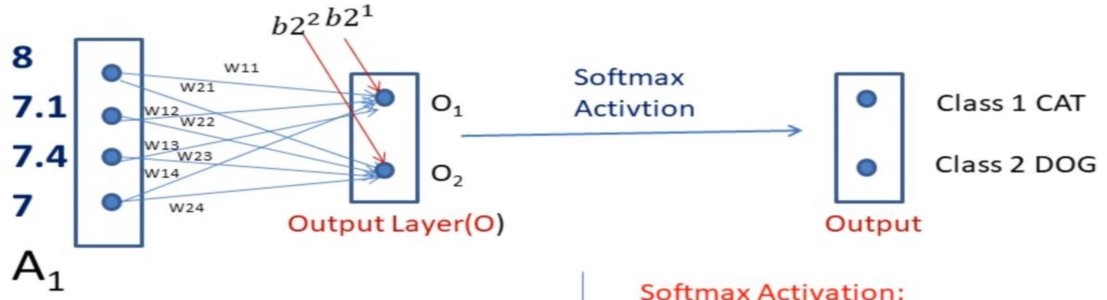


$$A_1 = W_1 X + b_1$$

$$\begin{bmatrix} A11 \\ A12 \\ A13 \\ A14 \end{bmatrix} = \begin{bmatrix} 1.0 & 1.0 & 0.2 & 0.8 \\ 1.0 & 0.5 & 0.5 & 0.8 \\ 0.8 & 1.0 & 0.2 & 0.8 \\ 0.5 & 0.5 & 1.0 & 1.0 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \\ 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 8 \\ 7.1 \\ 7.4 \\ 7 \end{bmatrix}$$

$$\text{ReLU}\begin{bmatrix} 8 \\ 7.1 \\ 7.4 \\ 7 \end{bmatrix} = \begin{bmatrix} 8 \\ 7.1 \\ 7.4 \\ 7 \end{bmatrix}$$

Full connected Layers & Softmax Activation



$$O = W_2 A_1 + b_2$$

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.1 & 0.5 & 0.5 \\ 0.1 & 0.5 & 0.1 & 0.2 \end{bmatrix} \begin{bmatrix} 8 \\ 7.1 \\ 7.4 \\ 7 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

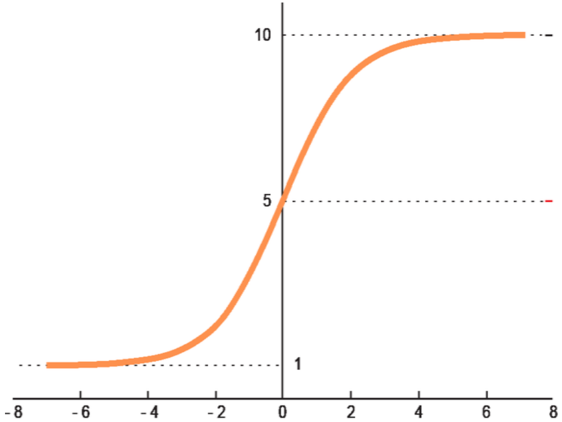
$$= \begin{bmatrix} 9.61 \\ 6.49 \end{bmatrix}$$

Softmax Activation:

$$\sigma(O_i) = \frac{e^{o_i}}{\sum e^{o_i}}$$

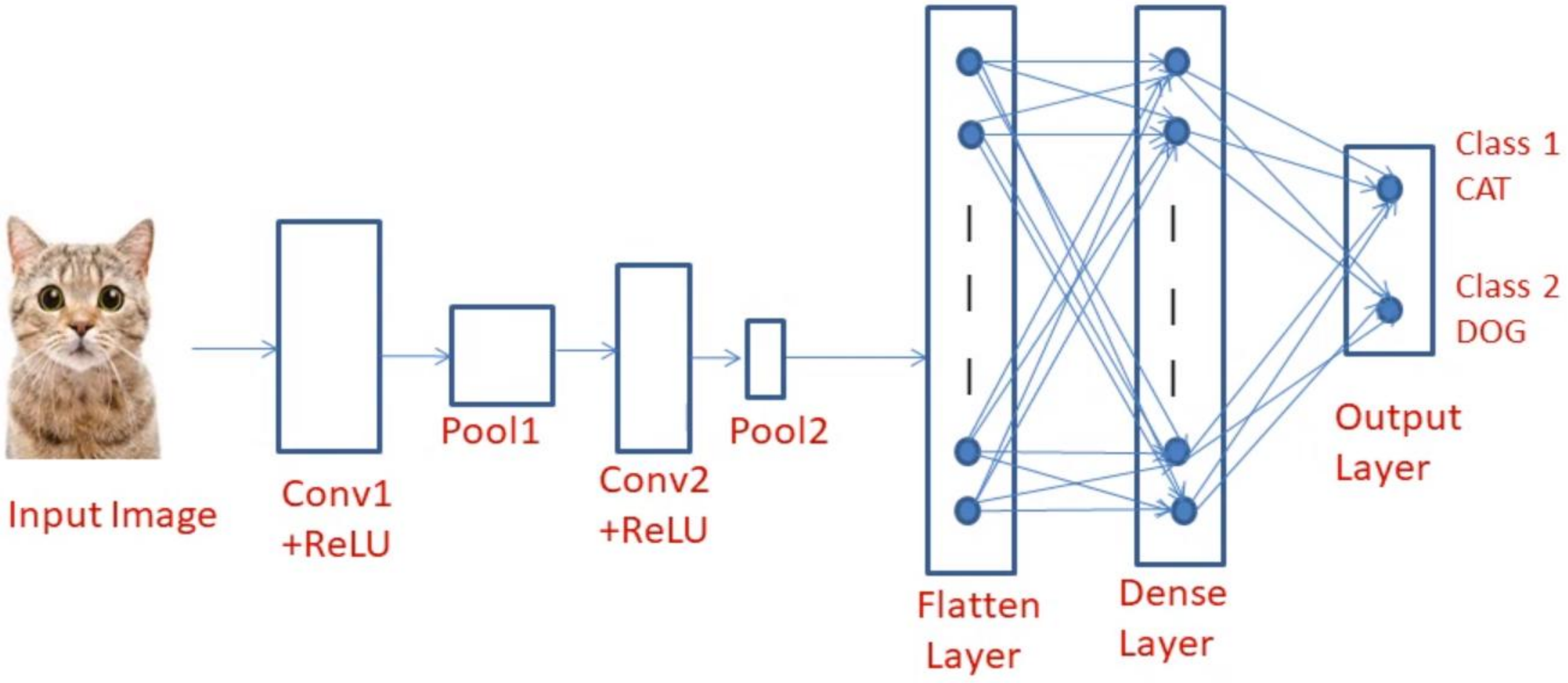
$$\sigma(O1) = \frac{e^{o1}}{e^{o1}+e^{o2}} = \frac{14913.17}{14913.17+658.52} = 0.95$$

$$\sigma(O2) = \frac{e^{o2}}{e^{o1}+e^{o2}} = \frac{658.52}{14913.17+658.52} = 0.04$$



Softmax Activation

Stacked CNN Architecture

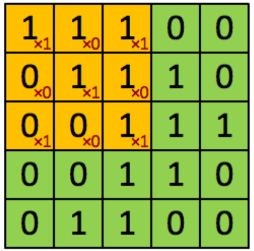


Activate

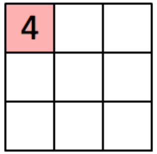
CNN

Skills For Future Jobs

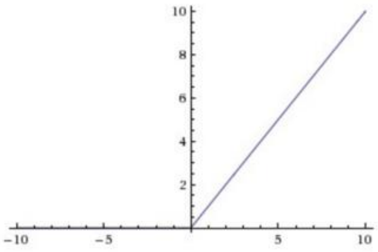
Convolutional Layer	Rectified linear unit (ReLU)	Pooling Layer	Fully Connected Layer



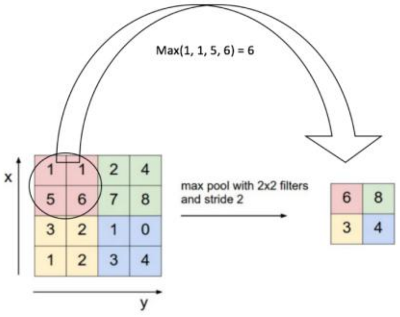
Image



Convolved Feature



Output = Max(zero, Input)



Rectified Feature Map

Car	70 %
Truck	20 %
Bicycle	10 %

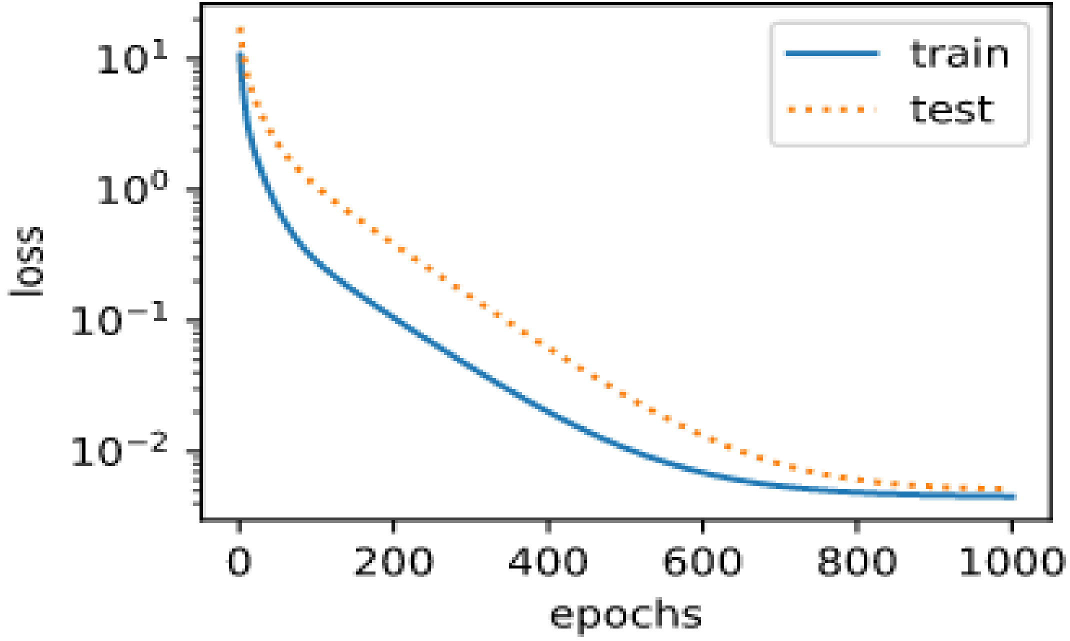
Reduce size, improve feature, give probability value

Stacked CNN Architecture

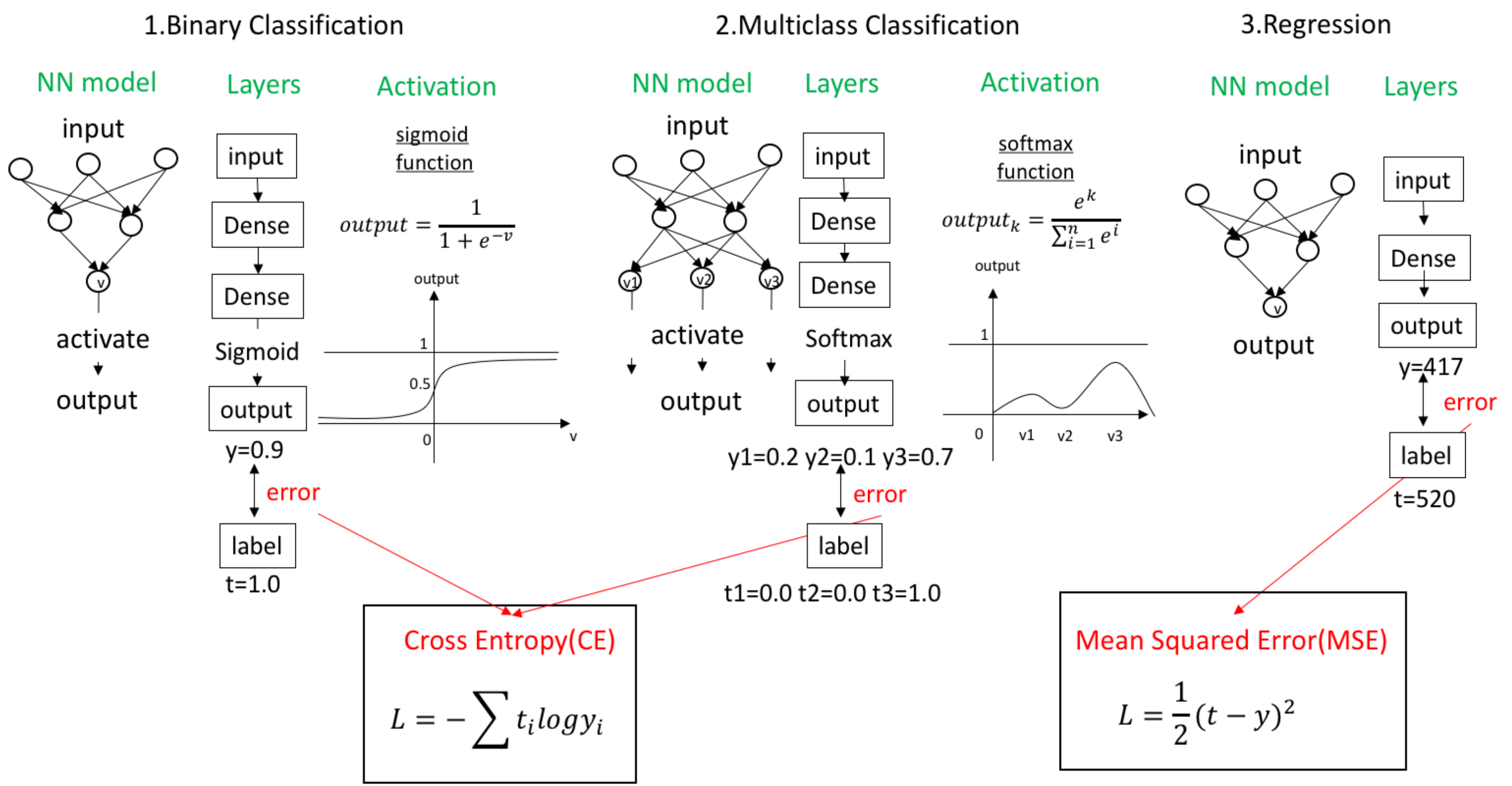
```
# Arsitektur CNN dengan 2 Convolution Layers
model = tf.keras.Sequential()

# Convolution layers pertama
model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=2, padding='same',
                                 activation='relu', input_shape=(28,28,1)))
model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

# Convolution layers kedua
model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=2, padding='same',
                                 activation='relu'))
model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

# Flattern
model.add(tf.keras.layers.Flatten())
# Fully connected layers
model.add(tf.keras.layers.Dense(256, activation='relu'))
model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

Loss function

The loss function dalam Neural Network menakuantifikasi perbedaan antara hasil yang diharapkan dan hasil yang dihasilkan oleh model.



Loss function



TERIMA KASIH

Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media

- Orbit Future Academy
- @OrbitFutureAcademyIn1
- OrbitFutureAcademy
- Orbit Future Academy

Orbit Future Academy — Skills For Future Jobs

AI Mastery Course

Module 4
Computer Vision

Section
Object Detection with YOLO

Learning Objective

- Memahami prinsip kerja object detection
- Memahami prinsip kerja YOLO (You Only Look Once) Algorithm
- Implementasi YOLO

1

Overview

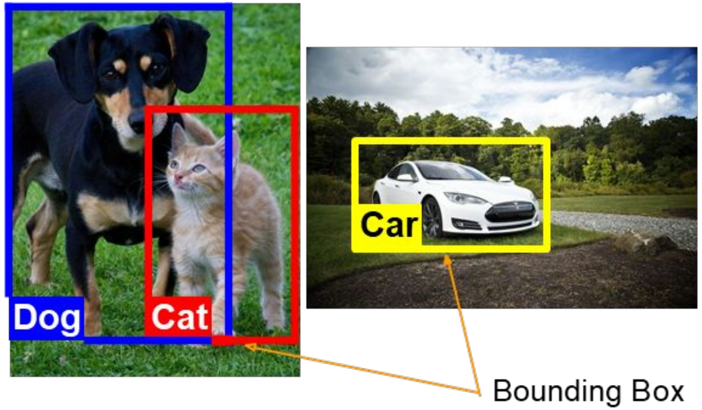
Getting Started

Classification

- Setiap image memiliki satu objek
- Model melakukan prediksi satu label

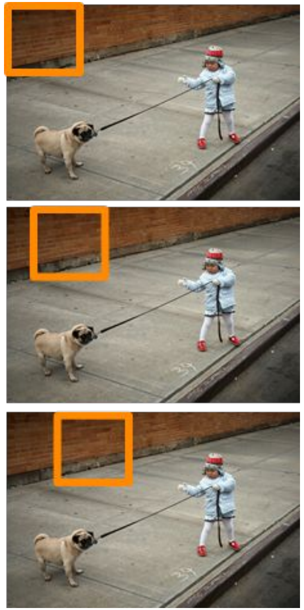
Object Detection

- Setiap image mengandung banyak objek
- Model mengklasifikasi objek dan mengidentifikasi lokasi

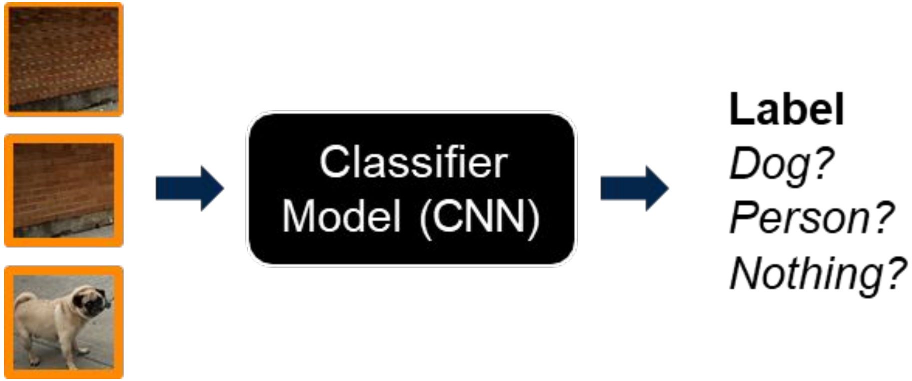


Naive Approach

Step1: Scan image menggunakan *sliding window*



Step2: Memasukkan image ke dalam model pengklasifikasi untuk memprediksi label untuk wilayah (region) itu.

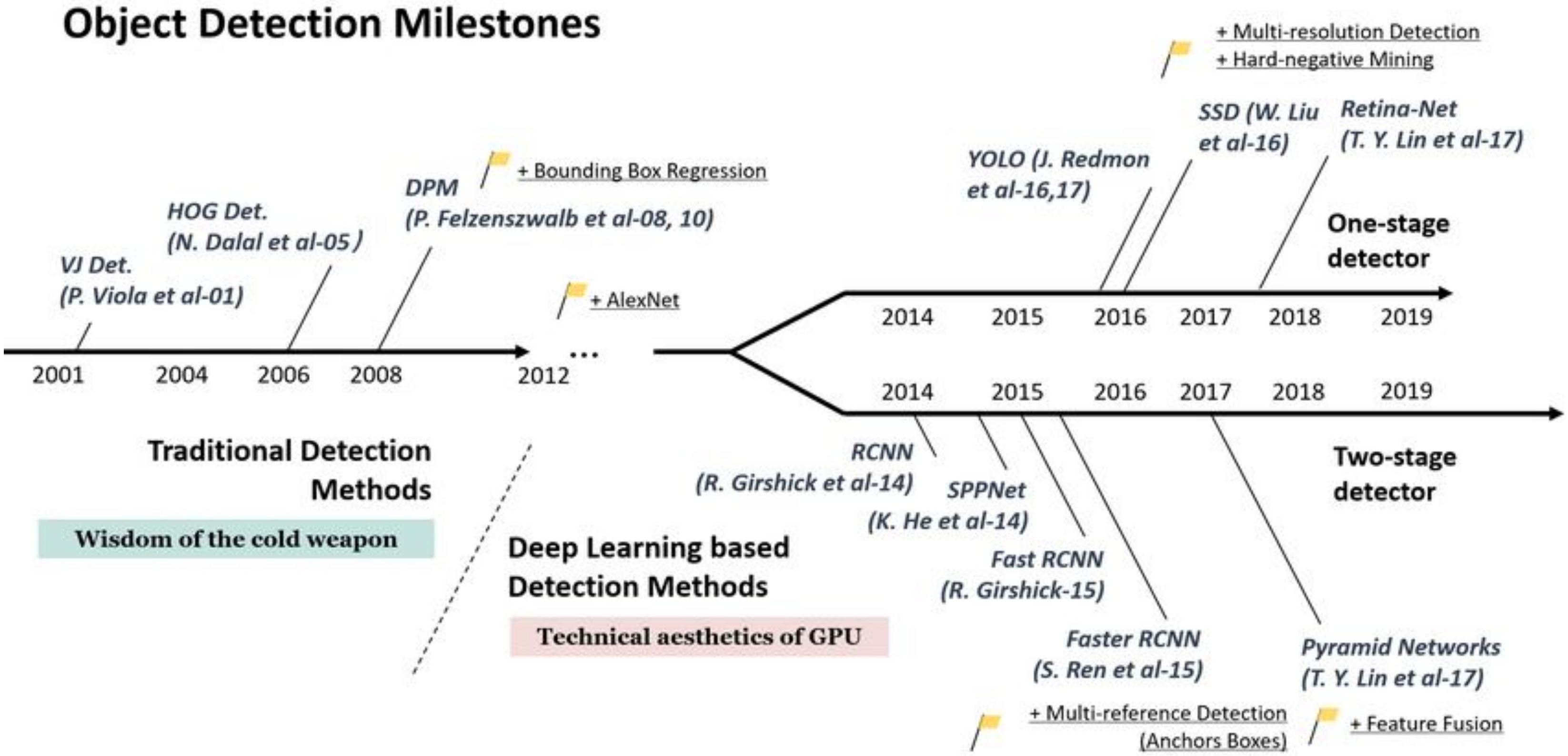


Naive **Approach**

Pendekatan ini lambat karena memeriksa banyak jendela (window) yang tidak berisi apa pun -> Tidak bagus untuk penggunaan waktu nyata (real time).

Region-based Convolutional Neural Network (R-CNN) adalah pendekatan berbasis region yang secara strategis mengidentifikasi objek dan lokasi melalui CNN.

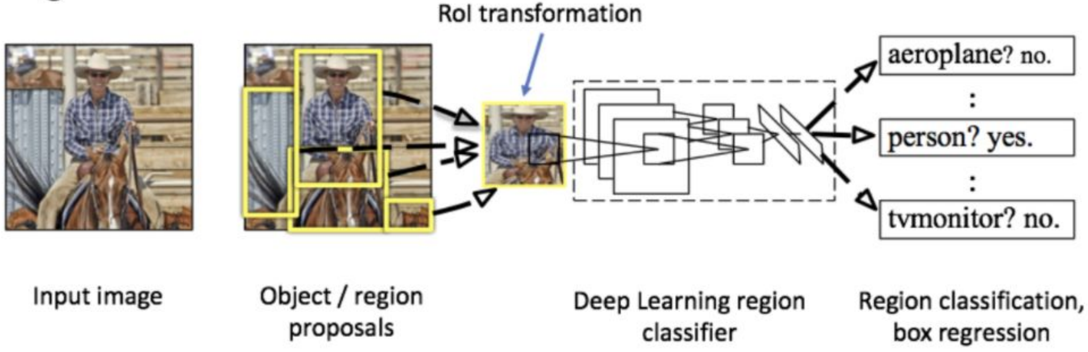
Timeline



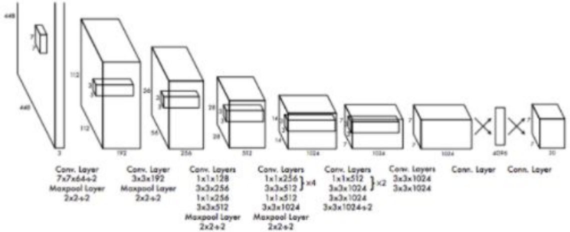
Object Detection Milestones
VJ Det.
(P. Viola et al-01)
HOG Det.
(N. Dalal et al-05)
DPM
(P. Felzenszwalb et al-08, 10)
+ Bounding Box Regression
+ AlexNet
2001
2004
2006
2008
2012
...
Traditional Detection Methods
Wisdom of the cold weapon
Deep Learning based Detection Methods
Technical aesthetics of GPU
YOLO (J. Redmon et al-16,17)
SSD (W. Liu et al-16)
Retina-Net (T. Y. Lin et al-17)
+ Multi-resolution Detection
+ Hard-negative Mining
One-stage detector
2014
2015
2016
2017
2018
2019
2014
2015
2016
2017
2018
2019
Two-stage detector
RCNN
(R. Girshick et al-14)
SPPNet
(K. He et al-14)
Fast RCNN
(R. Girshick-15)
Faster RCNN
(S. Ren et al-15)
Pyramid Networks
(T. Y. Lin et al-17)
+ Multi-reference Detection
(Anchors Boxes)
+ Feature Fusion

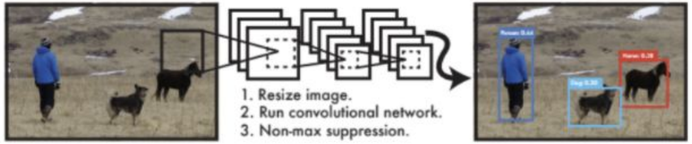
One Stage vs Two Stage

Many stage



One stage





Redmond et al. You Only Look Once:
Unified Real-time Object Detection. In CVPR 2016

One Stage vs Two Stage

Two stage detector, generate region proposal (e.g. selective search) seperti pada R-CNN dan Fast R-CNN. Kemudian setiap proposal dilakukan klasifikasi objek dan estimasi posisi. Stage ini memiliki kelebihan dalam hal **akurasi.** contoh : R-CNN, Fast R-CNN, Faster R-CNN

One stage detector, Klasifikasi objek dan regresi kotak pembatas dilakukan secara langsung tanpa menggunakan proposal wilayah yang telah dibuat sebelumnya (candidate object bounding boxes). Stage ini memiliki kelebihan dalam hal **kecepatan.** contoh : YOLO dan SSD

Image Detection Algorithms

METHOD	DETECTION SPEED	
DPM V5	0,07 FPS	14 second/gambar
R-CNN	0,05 FPS	20 second/gambar
Fast R-CNN	0,5 FPS	2 second/gambar
Fastest R-CNN	7 FPS	140 ms/gambar
YOLO	45 FPS	22 ms/gambar

2 YOLO Algorithm

You Only Look Once: Unified, Real-Time Object Detection

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

1. Introduction

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potential for general purpose, responsive robotic systems.

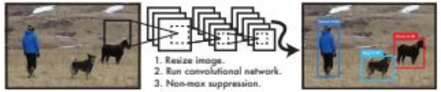


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene [13]. These complex pipelines are slow and hard to optimize because each individual component must be trained separately.

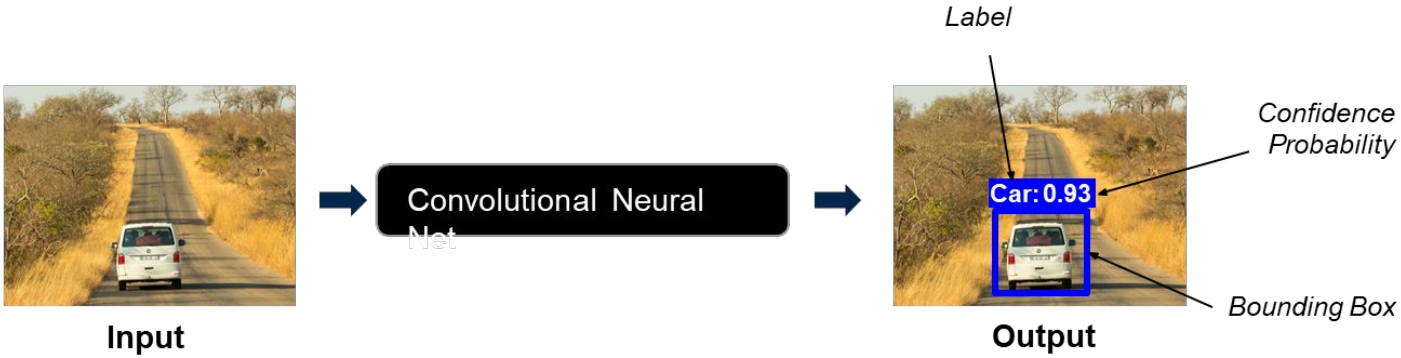
We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are.

YOLO is refreshingly simple: see Figure 1. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection.

First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. We simply run our neural network on a new image at test time to predict detections. Our base network runs at 45 frames per second with no batch processing on a Titan X

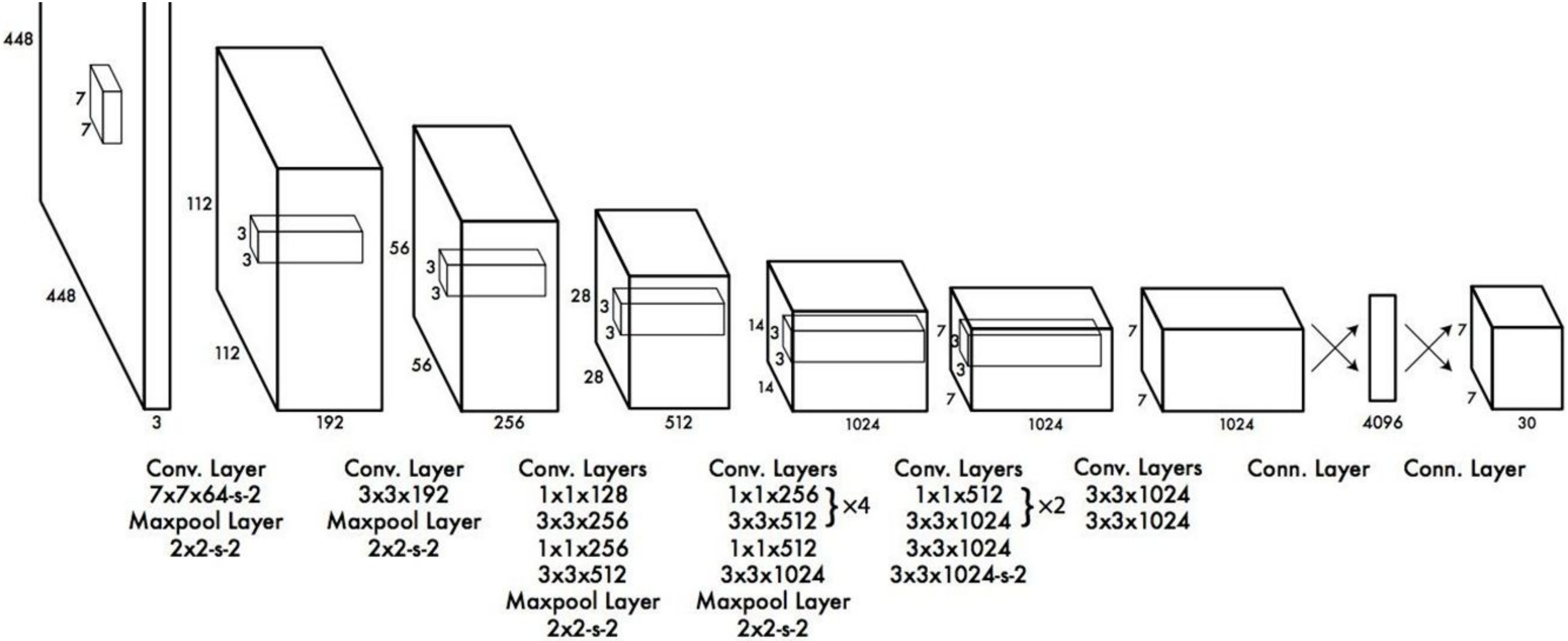
YOLO You Look Only Once

Alih-alih membuat prediksi di banyak wilayah (region) gambar, YOLO memproses seluruh gambar sekaligus ke CNN untuk memprediksi **labels, bounding boxes, dan confidence probabilitas** untuk setiap objek pada gambar.



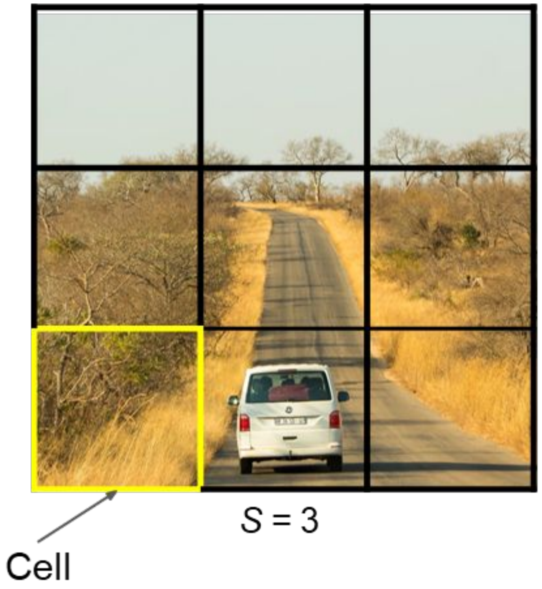
Berkeley SCET

Architecture



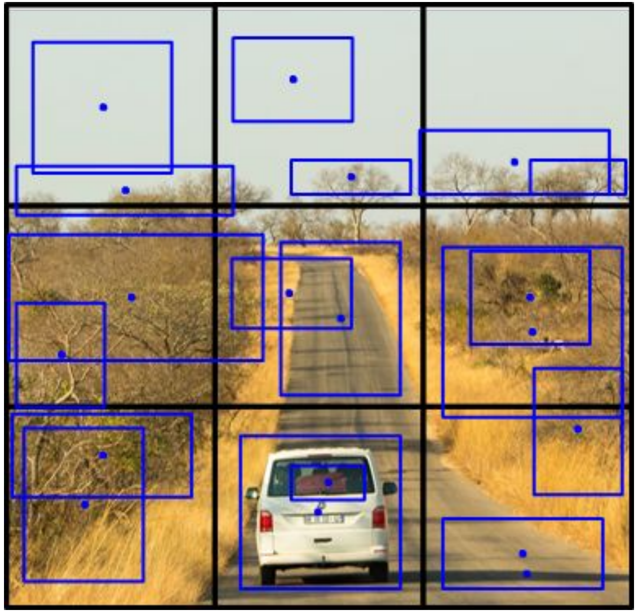
YOLO Steps

Step 1: Bagi gambar kedalam beberapa cells dengan ukuran **S x S grid**



YOLO Steps

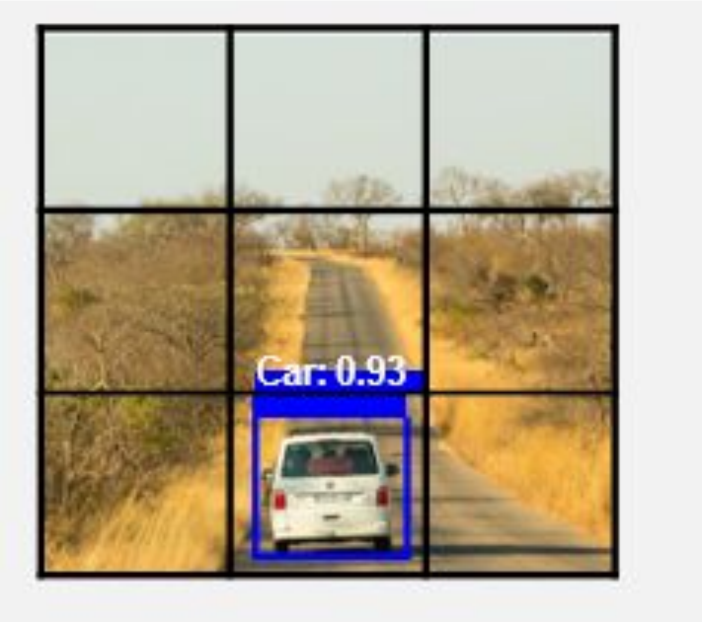
Step 2: Setiap cell memprediksi **B bounding boxes**



B = 2

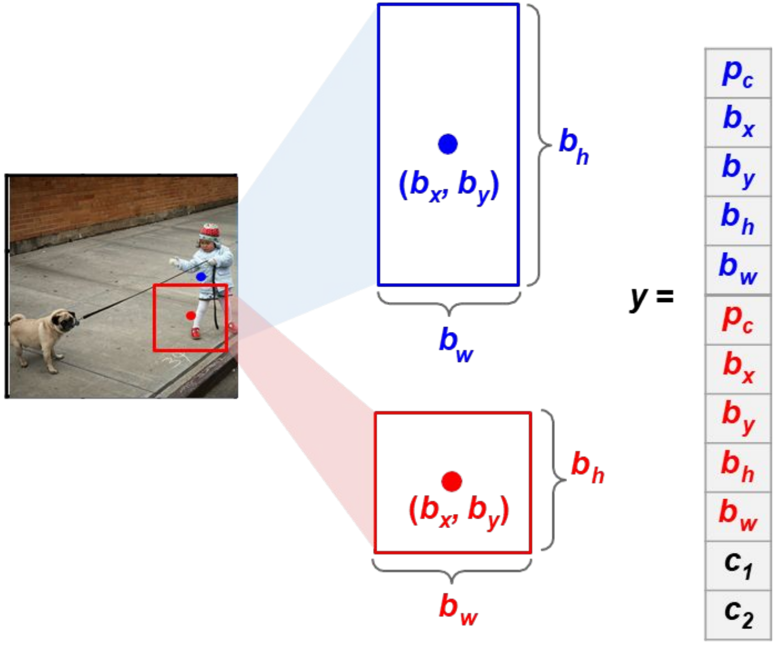
YOLO Steps

Step 3: Memproduksi **bounding boxes** dengan **confidence value**

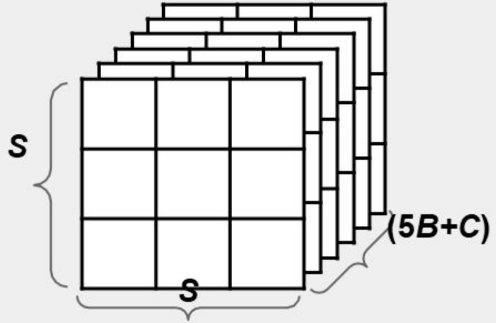


Encoding Multiple Bounding boxes

Apa yang terjadi jika kita memprediksi beberapa kotak pembatas per sel (B>1)?



The CNN will predict a ***y*** for each cell, so the size of the output tensor (multidimensional "matrix") should be: ***S×S×(5B+C)***



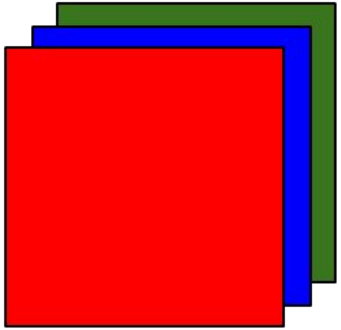
Notice that ***y*** has ***5B+C*** elements (***C*** is the number of classes).

YOLO Overview

Input

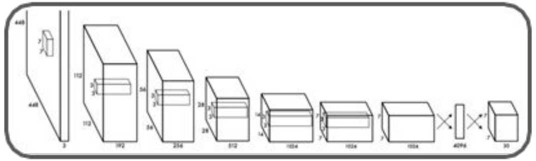
Convolutional Neural Net

Output

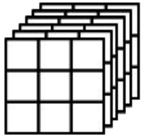


***W*×*H*×3**

***W*: Width of image in pixels**
L: Height of image in pixels
3: Number of color channels in RGB



Series of convolutional and pooling layers.



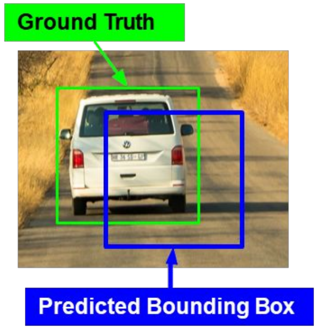
***S*×*S*×(5*B*+*C*)**

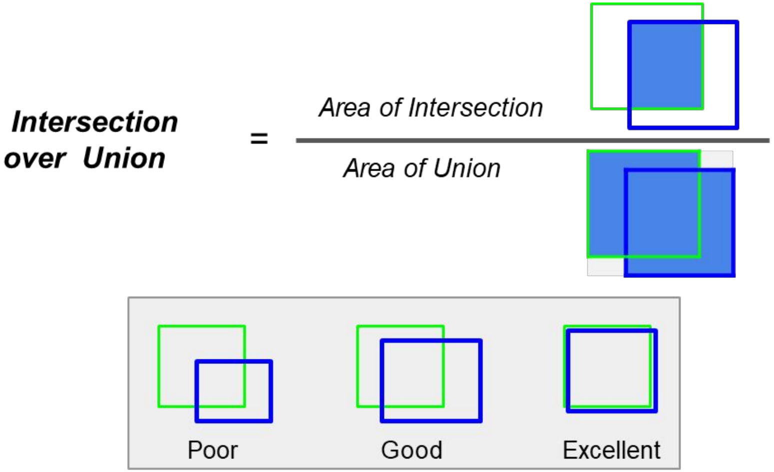
A tensor that specifies the bounding box locations and class probabilities.

Measuring Performance IoU

Intersection over Union (IoU) mengukur tumpang tindih antara dua boundary box.

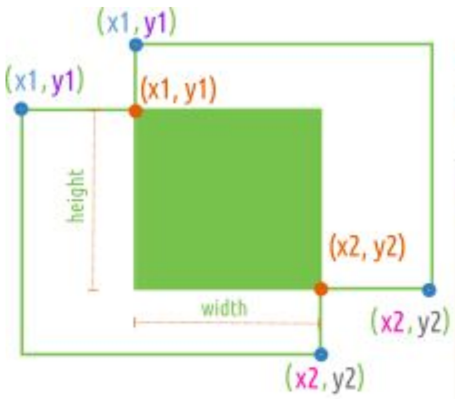
Selama proses training, akan dihitung **IoU** antara kotak pembatas yang diprediksi dan kepastian prediksi (kotak pembatas berlabel yang akan diprediksi)





Intersection over Union IoU

Calculating Overlapping Region



(x1, y1) = (max(x1),max(y1))
(x2, y2) = (min(x2) ,min(y2))

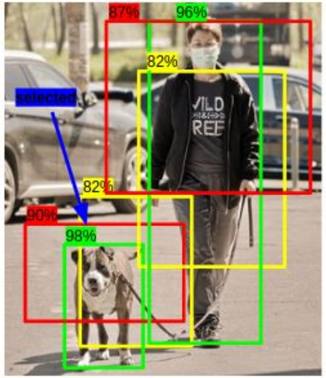
If width and height are both positive:
Overlap = width * height
= (x2-x1) * (y2-y1)
Else:
Overlapping Region = 0

Calculating Combined Region

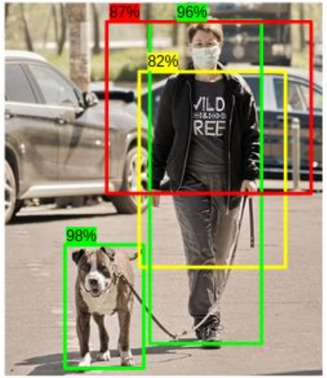
Combined Region = Area(boxA) + Area(boxB) - Overlapping Region

Double Counting Objects (Non-Max Suppression)

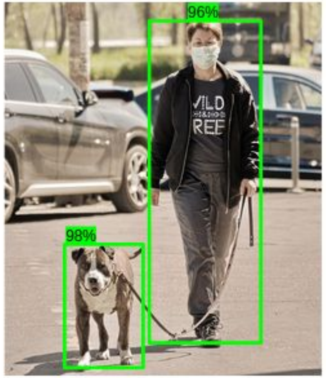
Saat memprediksi lebih dari 2 kotak pembatas per sel, terkadang objek yang sama akan terdeteksi beberapa kali (kotak yang tumpang tindih dengan label yang sama)



Step 1: Selecting Bounding box with highest score



Step 3: Delete Bounding box with high overlap



Step 5: Final Output

Loss function in YOLO

Regression loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Confidence loss

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Classification loss

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Confidence loss, determine whether there are objects in the prediction frame

Box regression loss, calculated only when the prediction box contains objects

Classification loss, decide which category the things in the prediction frame belong to.

4

Implementation

TERIMA KASIH

Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media

Orbit Future Academy
@OrbitFutureAcademyIn1
OrbitFutureAcademy
Orbit Future Academy

AI Mastery Course

Orbit Future Academy | Skills For Future Jobs

Module 4
Computer Vision

Section
Introduction of Autoencoder & Image Denoising

Learning Agenda

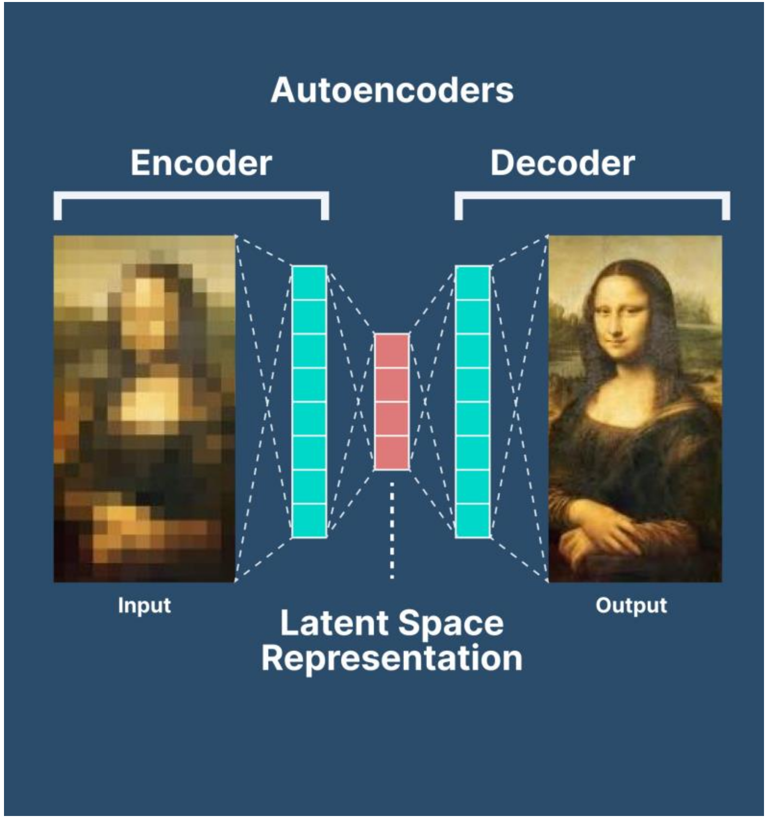
- Apa itu autoencoder?
- Properti pada autoencoder
- Arsitektur pada autoencoder
- Aplikasi dari autoencoder

1

Overview

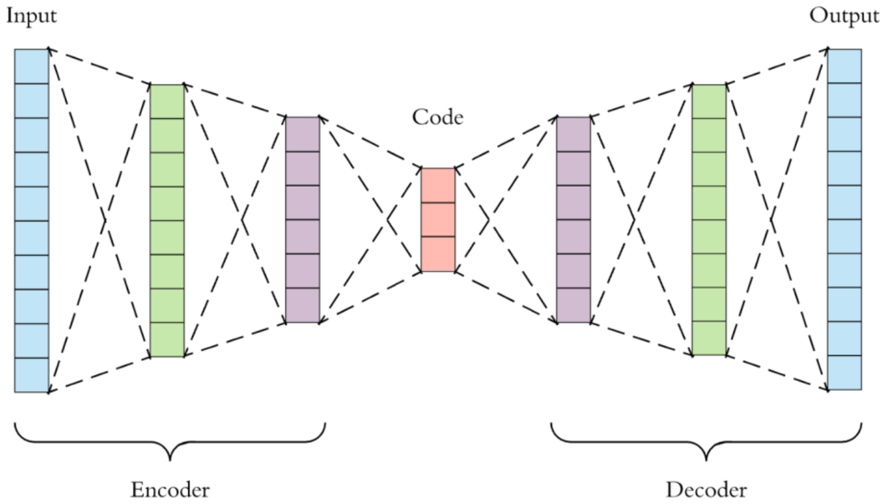
Introduction of Autoencoders

An **autoencoders** neural network adalah algoritme pembelajaran mesin tanpa pengawasan yang menerapkan backpropagation, Autoencoder mempelajari data input dan berusaha untuk melakukan rekonstruksi terhadap data input tersebut.



Components of Autoencoders

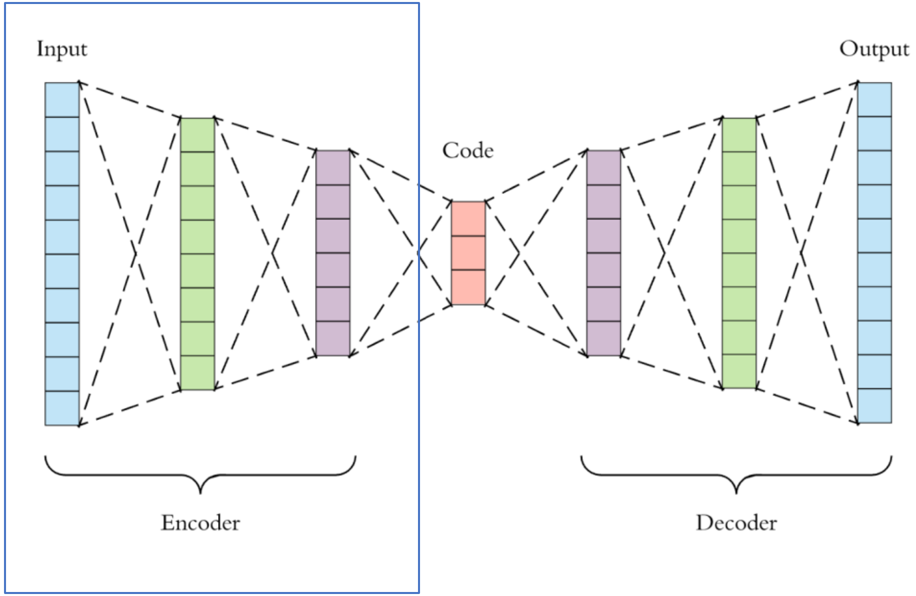
- Encoder
- Code
- Decoder



Components of Autoencoders

Encoder

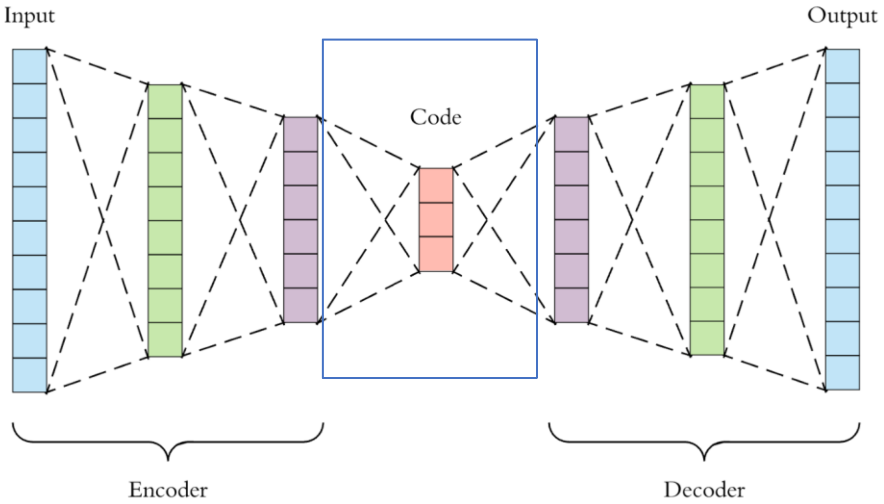
bagian jaringan ini bertujuan mengompres atau reduce dimension disimpan kedalam **latent space representation**



Components of Autoencoders

Code

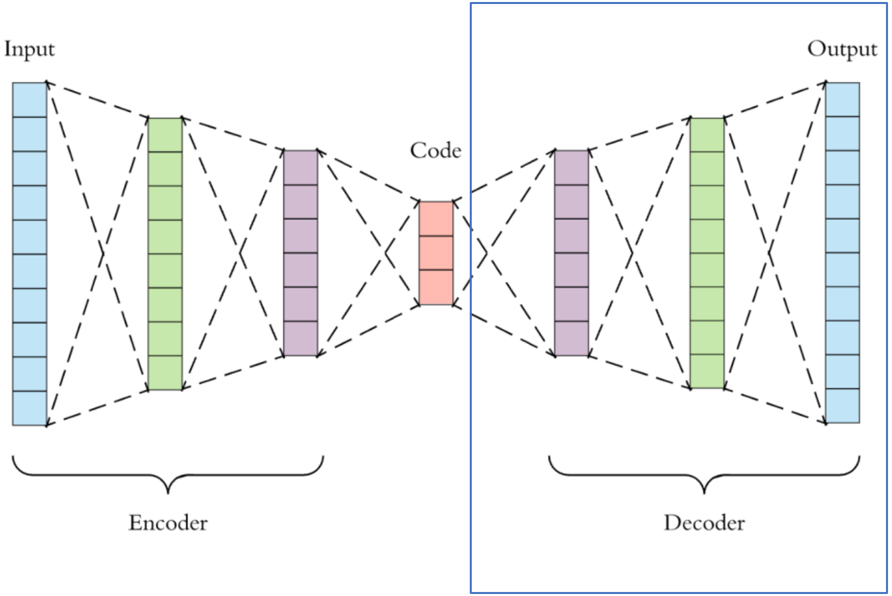
Ini adalah bagian dari jaringan yang mewakili input terkompresi yang diumpankan ke decoder



Components of Autoencoders

Decoder

Bagian ini bertujuan untuk merekonstruksi masukan dari the latent space representation



Properties of Autoencoders

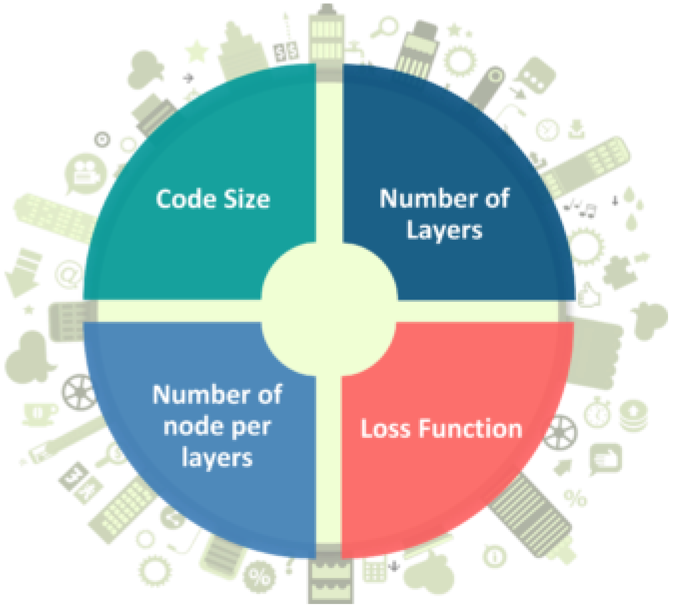
1. **Data-specific:** Autoencoder hanya dapat mengompresi data yang mirip dengan apa yang telah di latih.

1. **Lossy:** Output dari autoencoder tidak akan sama persis dengan input, itu akan menjadi representasi yang hampir mendekati original input tetapi terdegradasi.

1. **Unsupervised:** Untuk melatih autoencoder, kita tidak perlu melakukan sesuatu yang merepotkan, cukup lemparkan data input mentah ke model.

Hyperparameters of Autoencoders

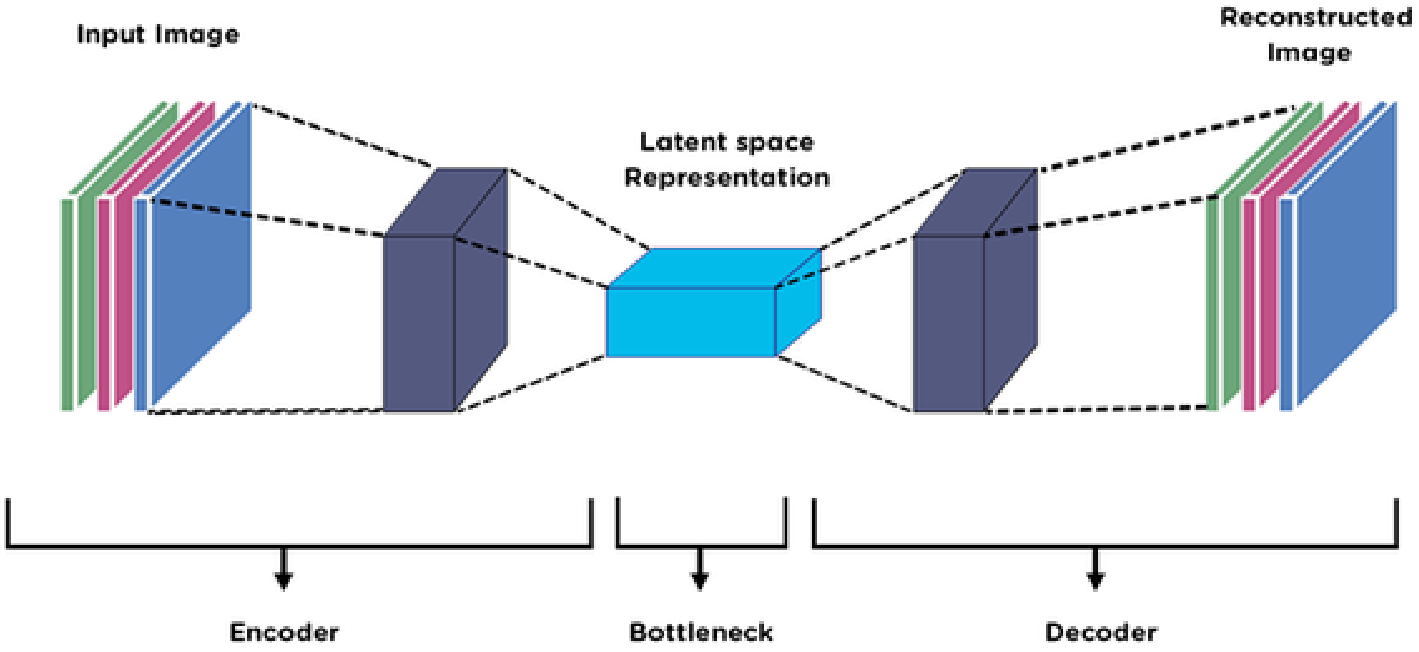
- **Code size,** Ukuran yang lebih kecil menghasilkan lebih banyak kompresi
- **Number of layers,** Autoencoder dapat memiliki banyak lapisan/layer
- **Loss function,** Mean squared error atau binary cross entropy
- **Number of node per layers,** Stacked autoencoders look like a sandwich



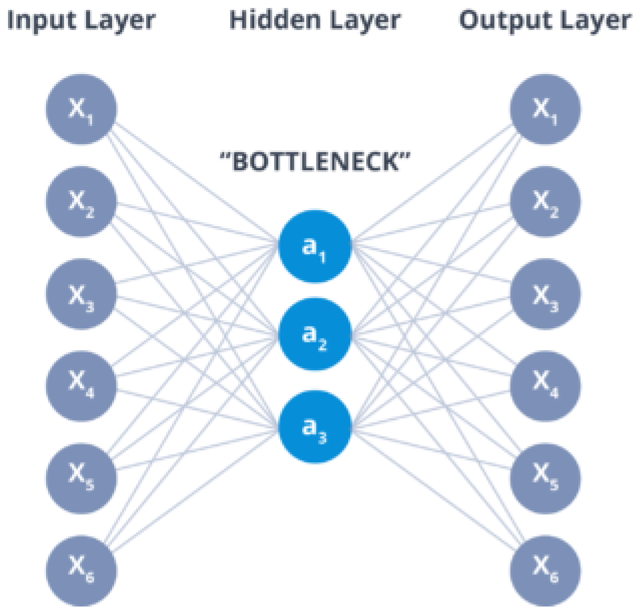
2

Architecture

Architecture of Autoencoders



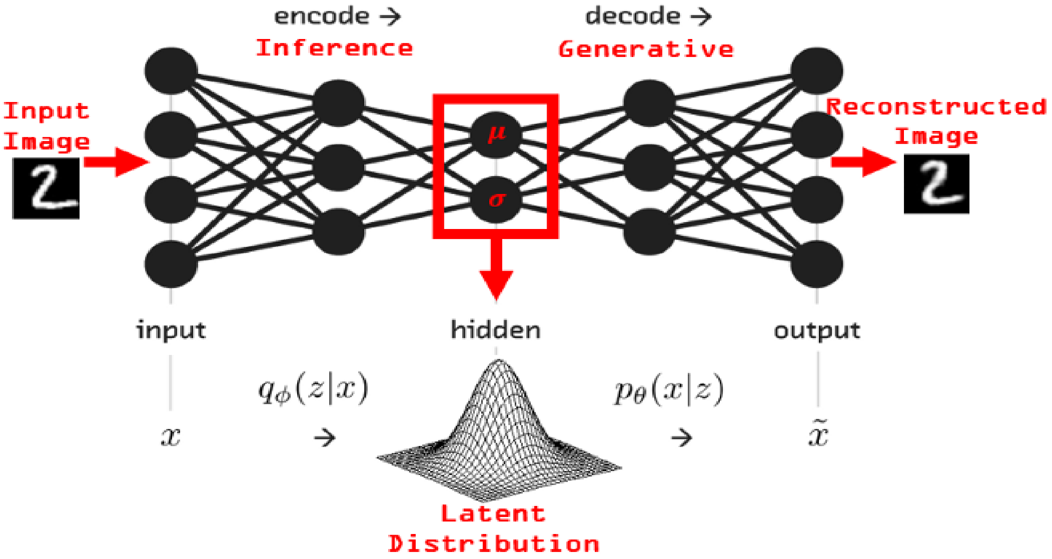
Architecture of Autoencoders



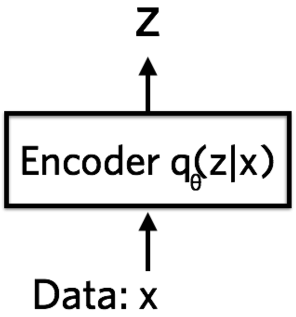
Bottleneck approach adalah pendekatan untuk memutuskan aspek mana dari data yang diamati yang merupakan informasi yang relevan dan aspek apa yang dapat dibuang

- Compactness of representation, measured as the compressibility
- Representation retains about some behaviourally relevant variables

Architecture of Autoencoders

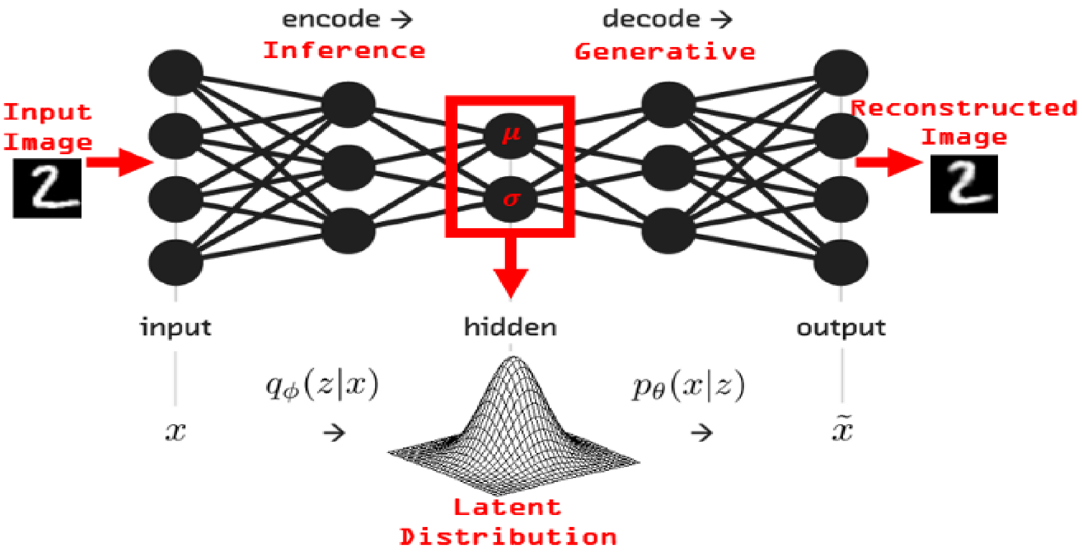


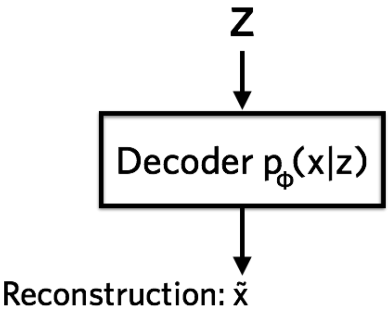
Encoder, is a neural network that outputs a representation **z** of data **x**



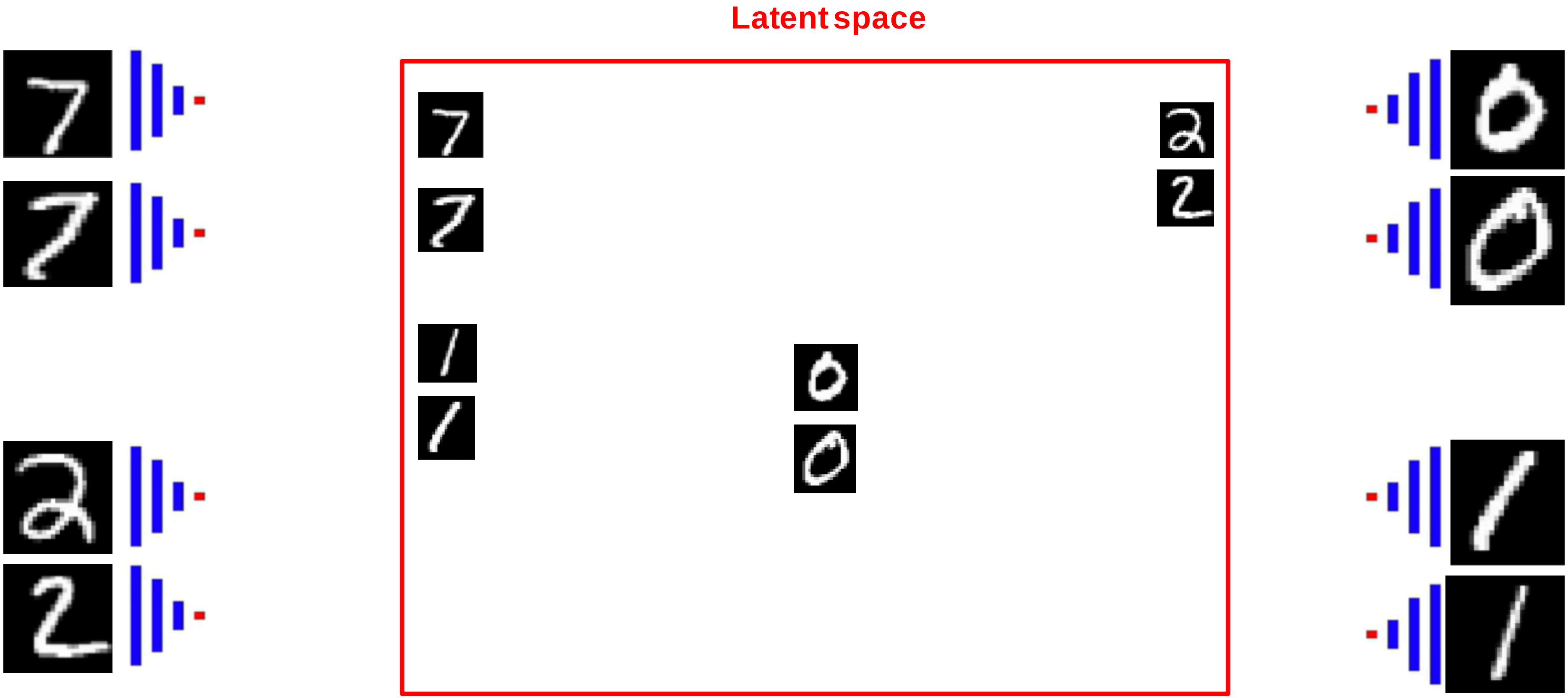
Architecture of Autoencoders

Decoder, is a neural net that learns to reconstruct the data **x** given a representation **z**

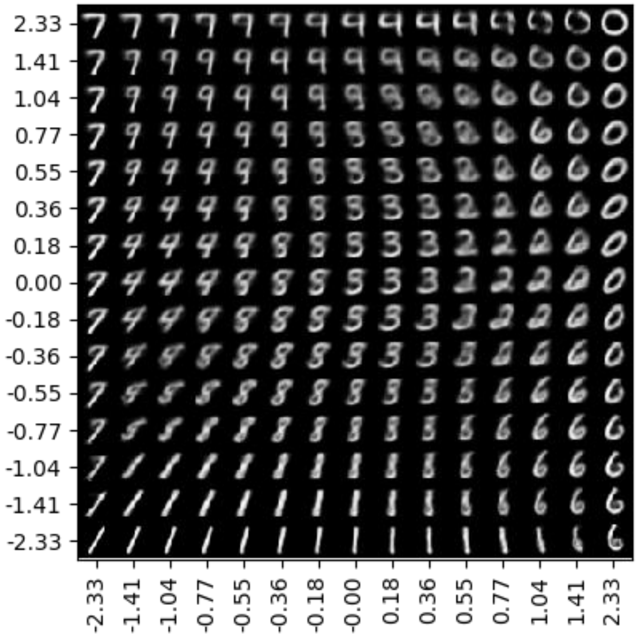


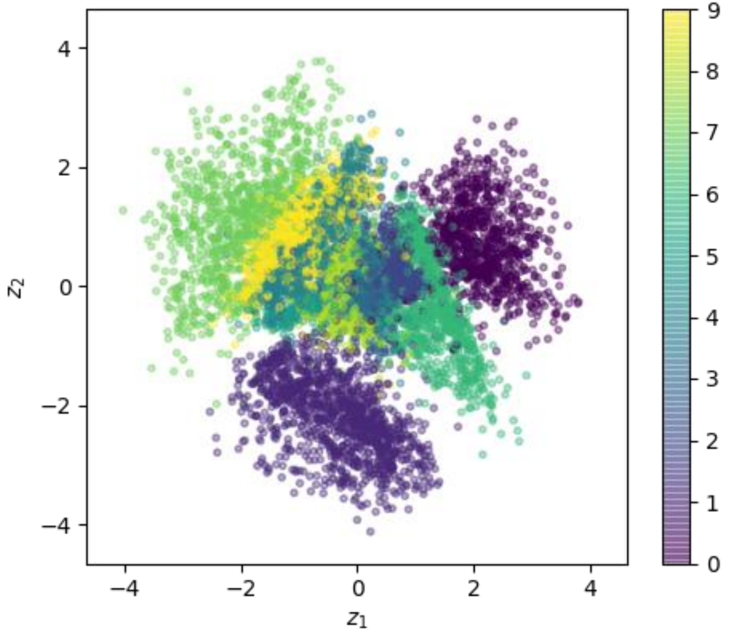


Latent Space Representation

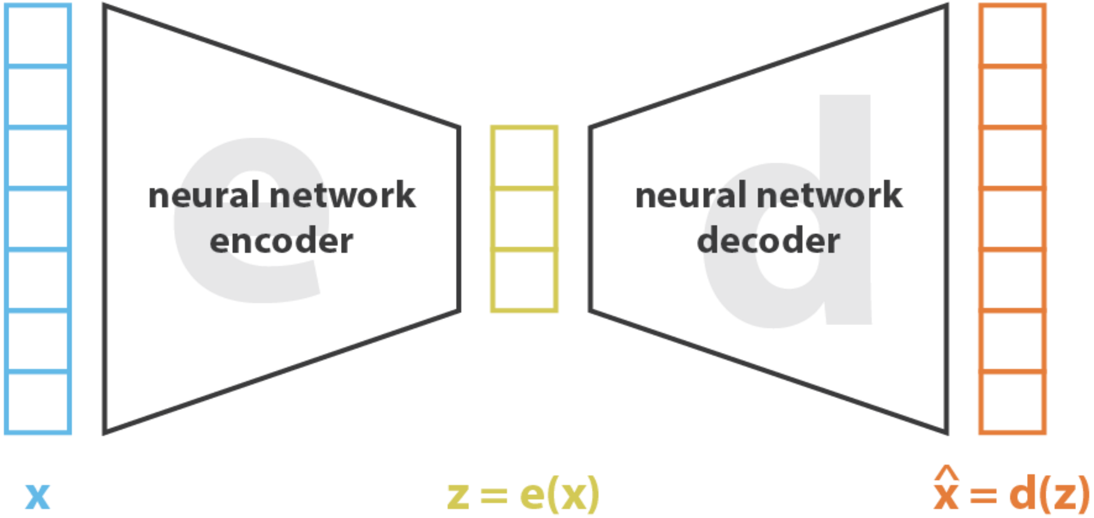


Latent Space Representation



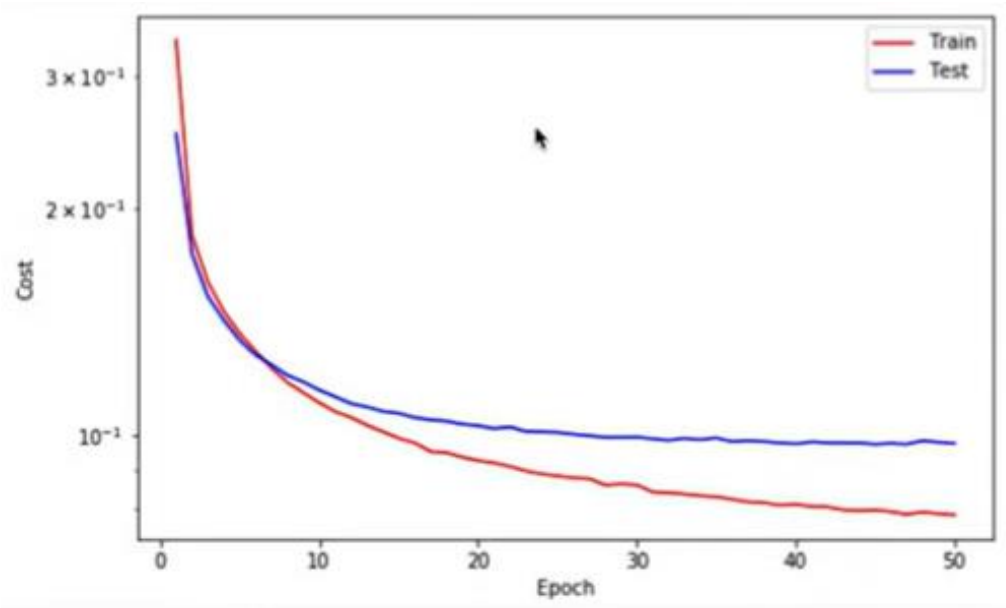


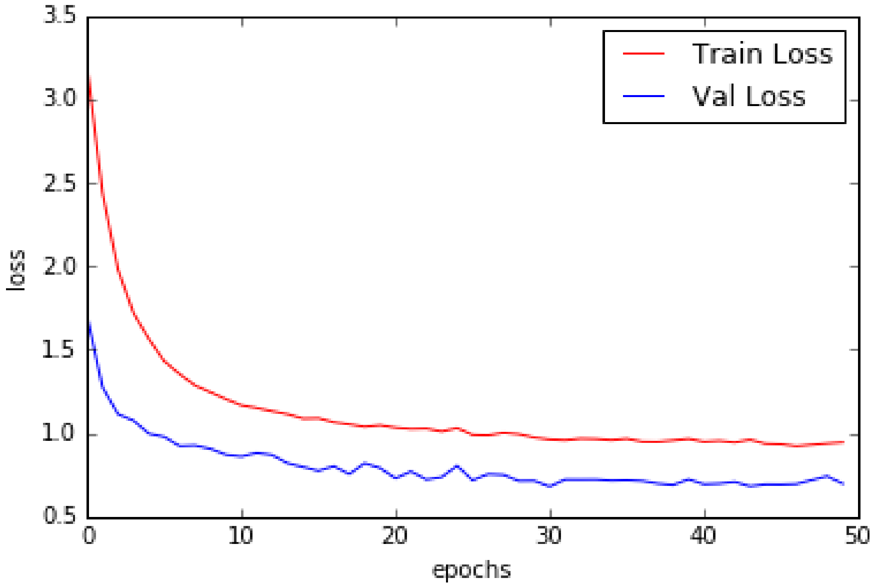
Loss function of Autoencoders



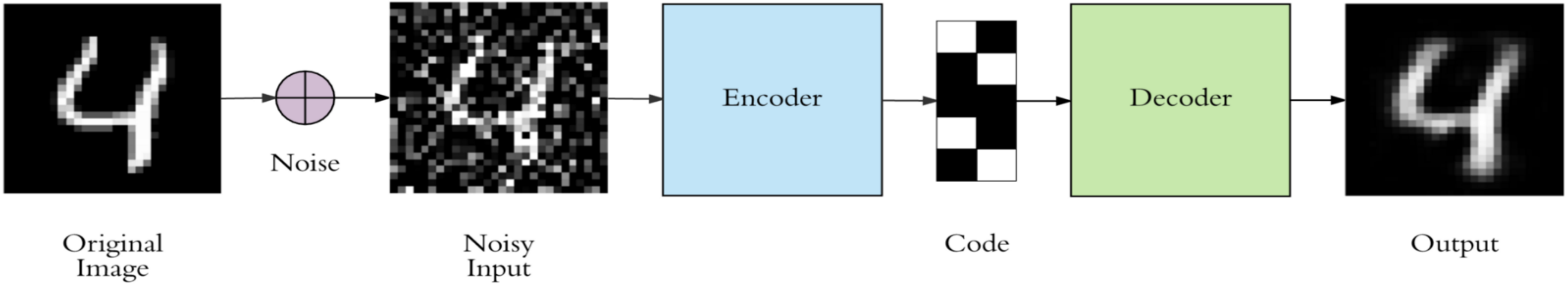
$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

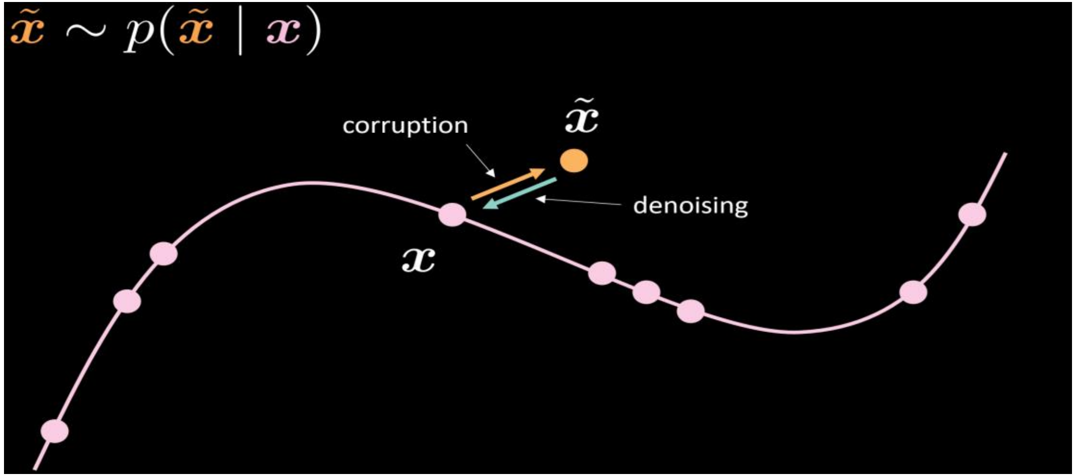
Loss function of Autoencoders





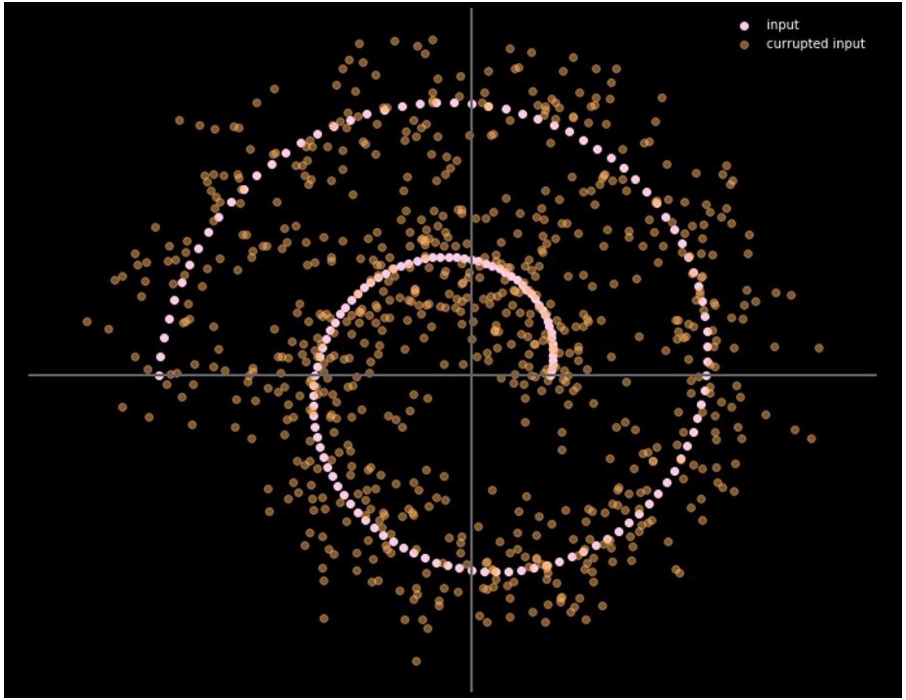
Denoising Autoencoders

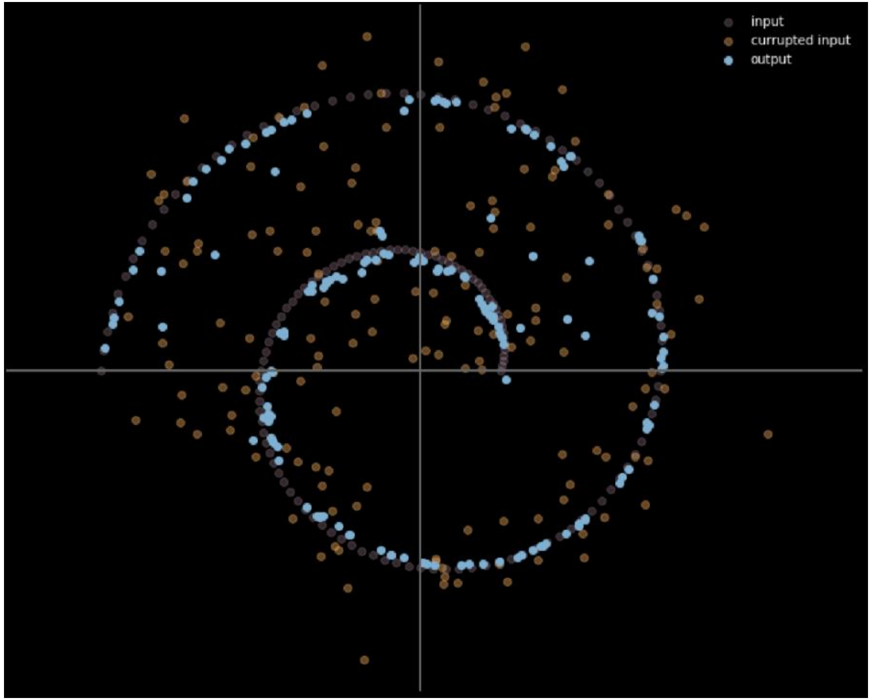




www.coursera.org

Denoising Autoencoders





hubungan antara data Input + Noise dan data Output

3 Aplikasi autoencoder

Image Reconstruction

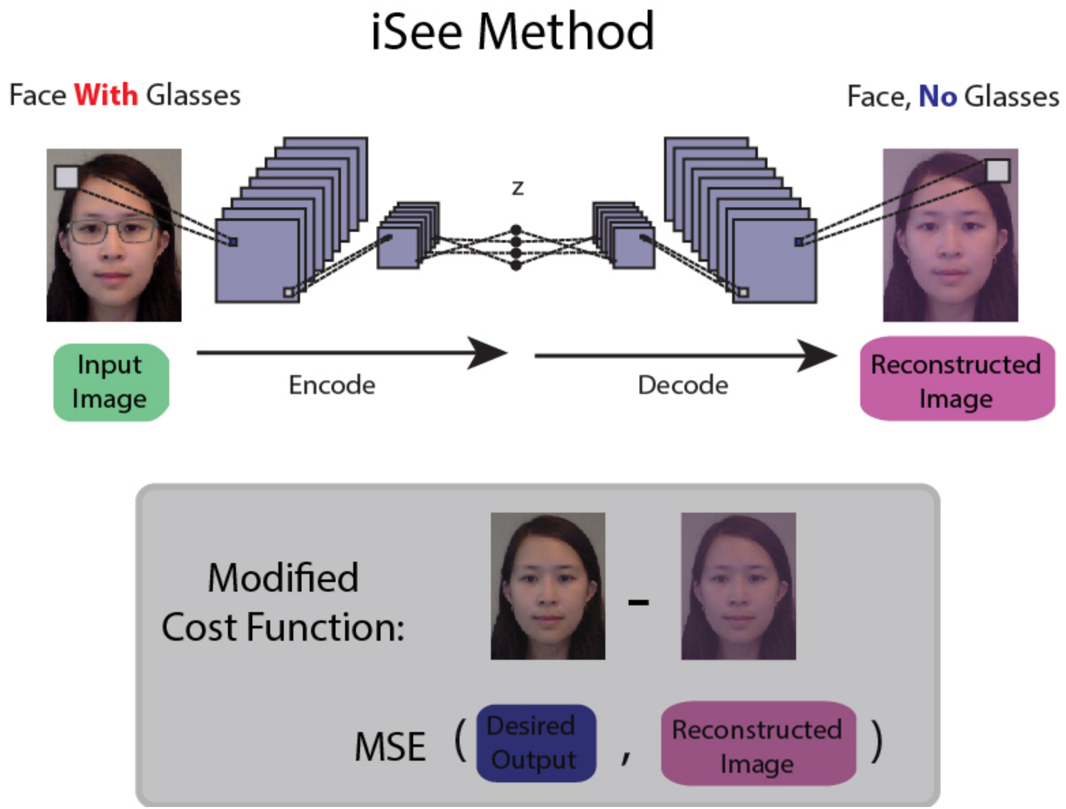


Image Coloring and Noise reduction

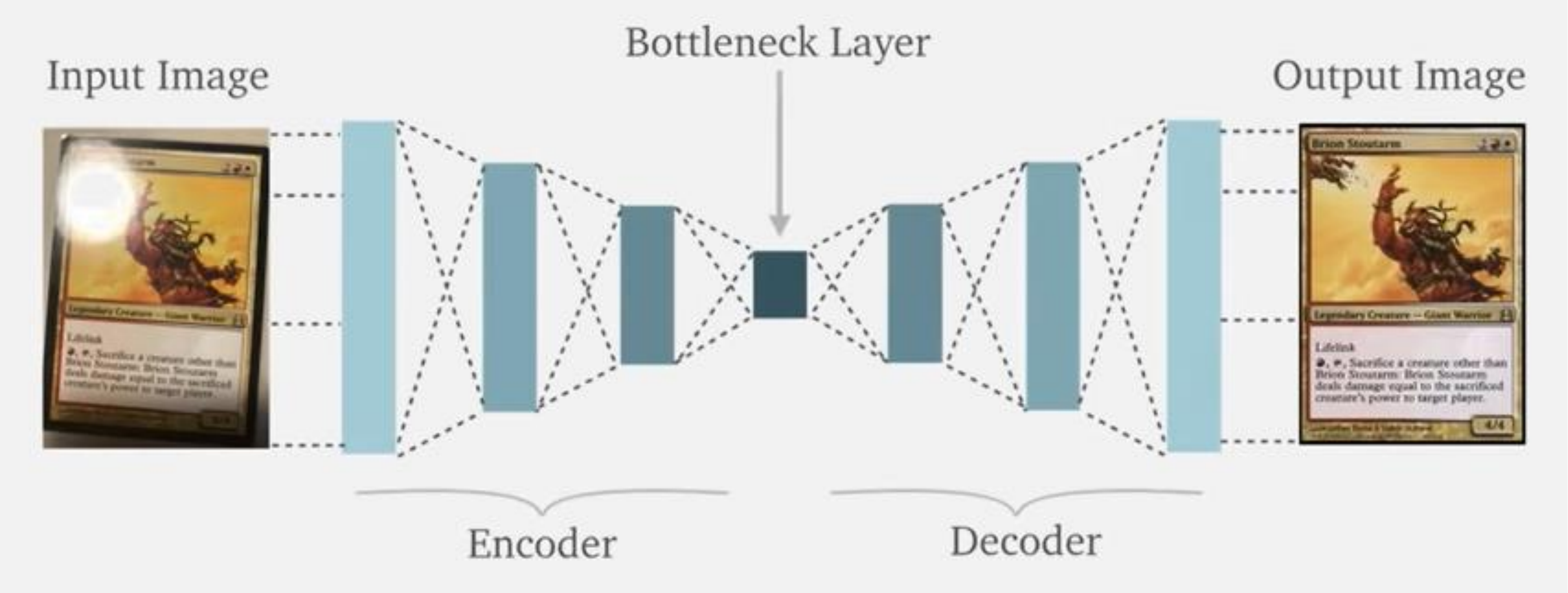
IMAGE COLORING

Before After

IMAGE NOISE REDUCTION

Before After

Feature Variation



4 Implementation

TERIMA KASIH

Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media

Orbit Future Academy
@OrbitFutureAcademyIn1
OrbitFutureAcademy
Orbit Future Academy

AI Mastery Course

Module 4
Computer Vision

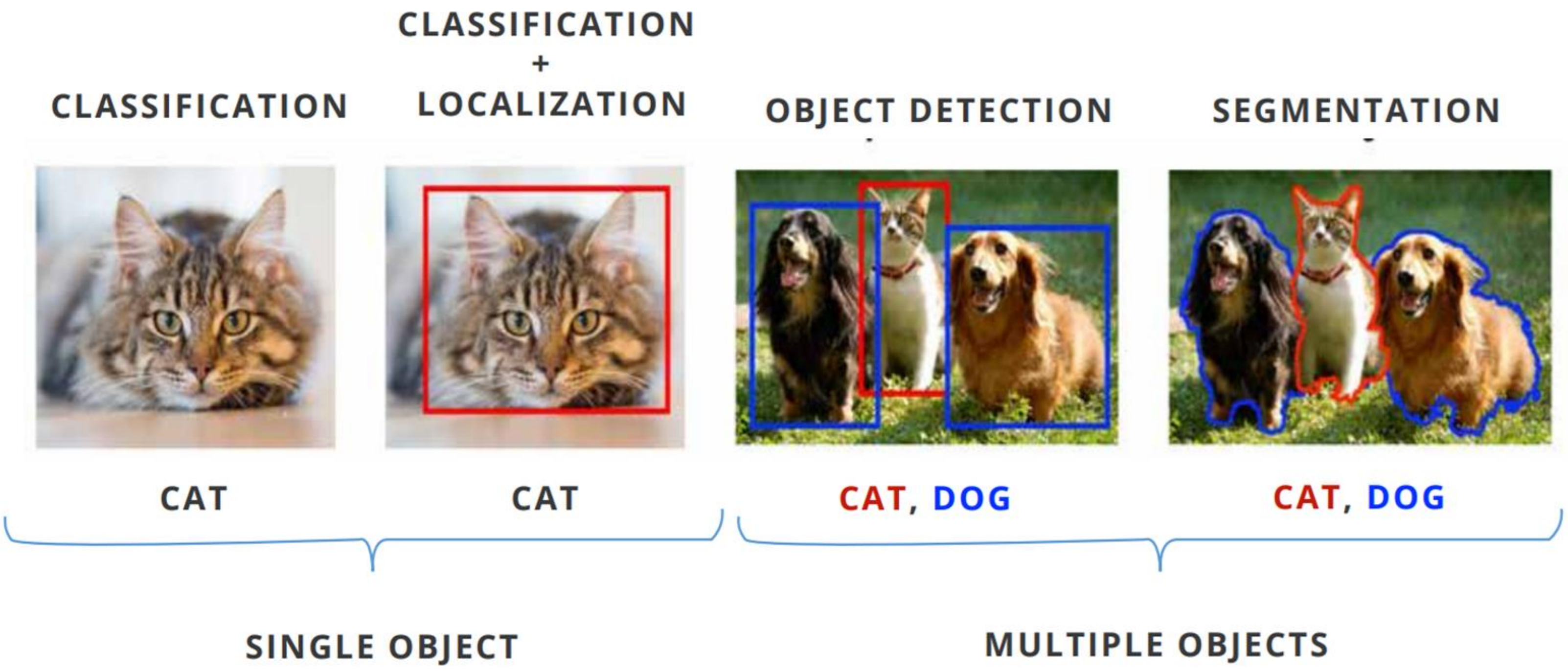
Section
Image Segmentation using U-Net

Learning Agenda

- Memahami prinsip kerja segmentasi
- Perbedaan antara instance dan semantic segmentation
- U-Net untuk instance segmentation
- Experiment

1 Overview

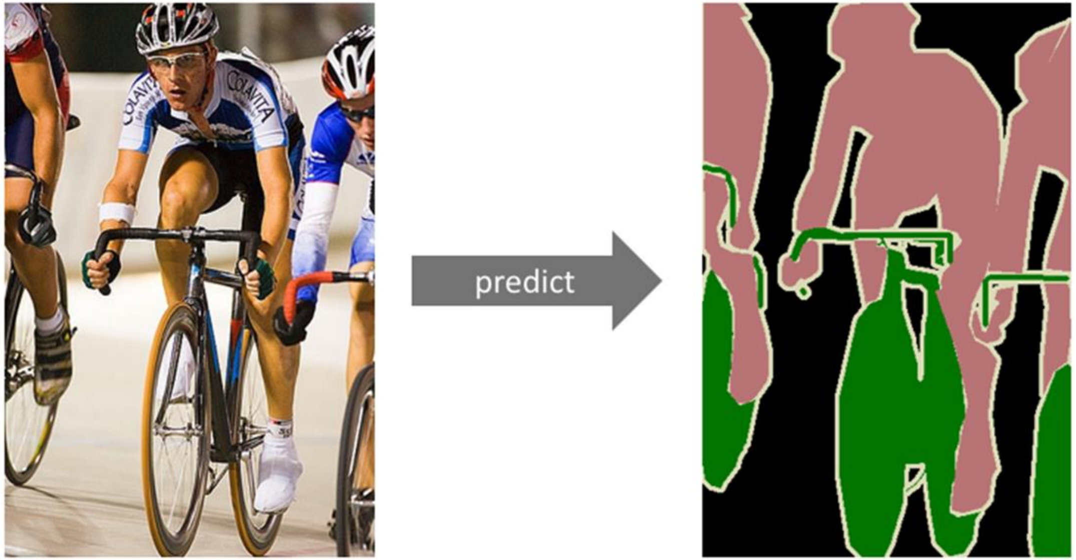
Computer Vision Landscape



Overview

The problem to classify the pixels dalam suatu gambar dan kemudian mengelompokkan gambar dengan menggambar garis di sekitar objek disebut Segmentasi

Overview

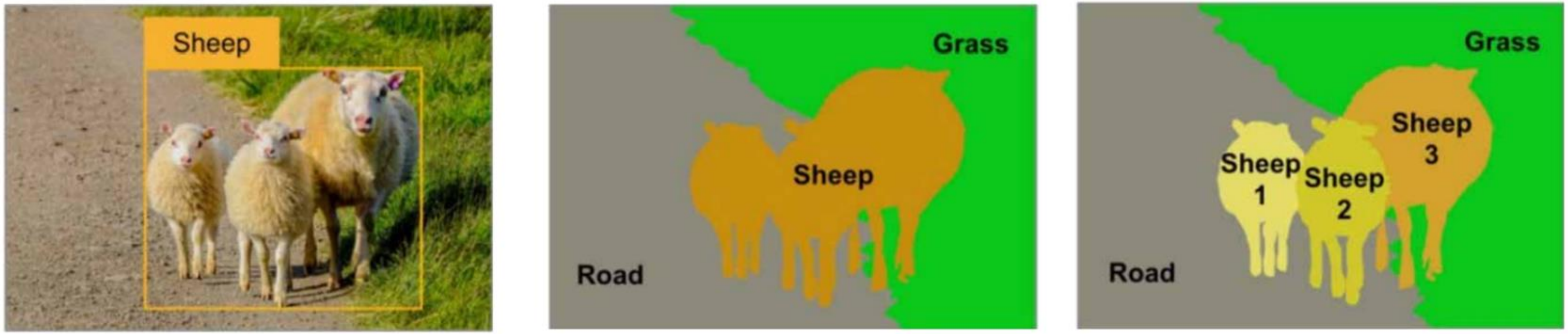


Person
Bicycle
Background

Type of Segmentation

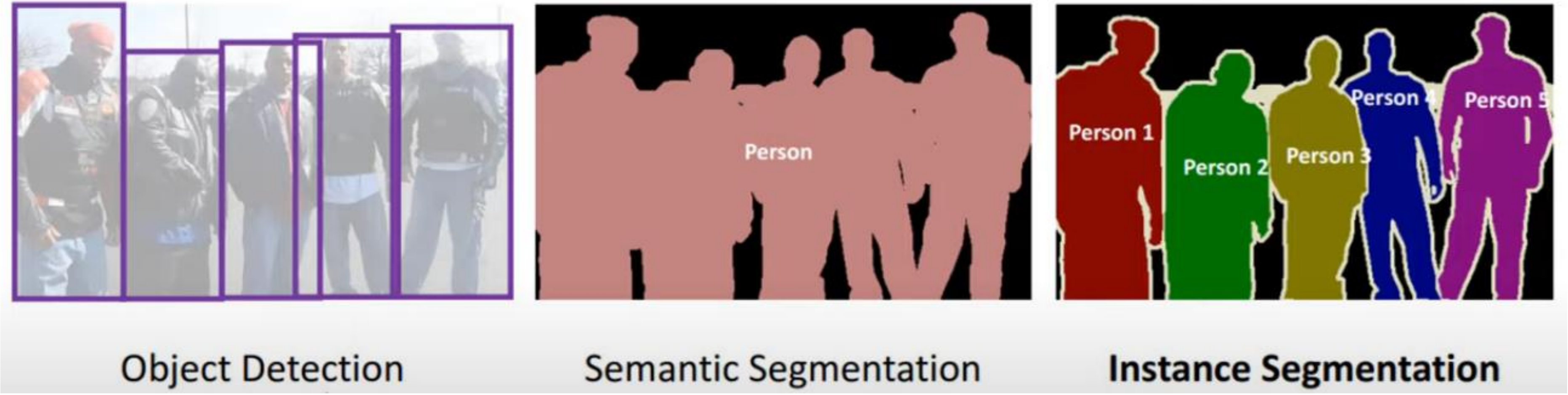
•Semantic Segmentation masks road, sheep and grass. dapat mengidentifikasi objek yang berbeda.

•Instance Segmentation masks different sheep. dapat mengidentifikasi berbagai contoh objek.



Classification and Localization | Segmantic segmentation | Instance segmentation

Type of Segmentation



2

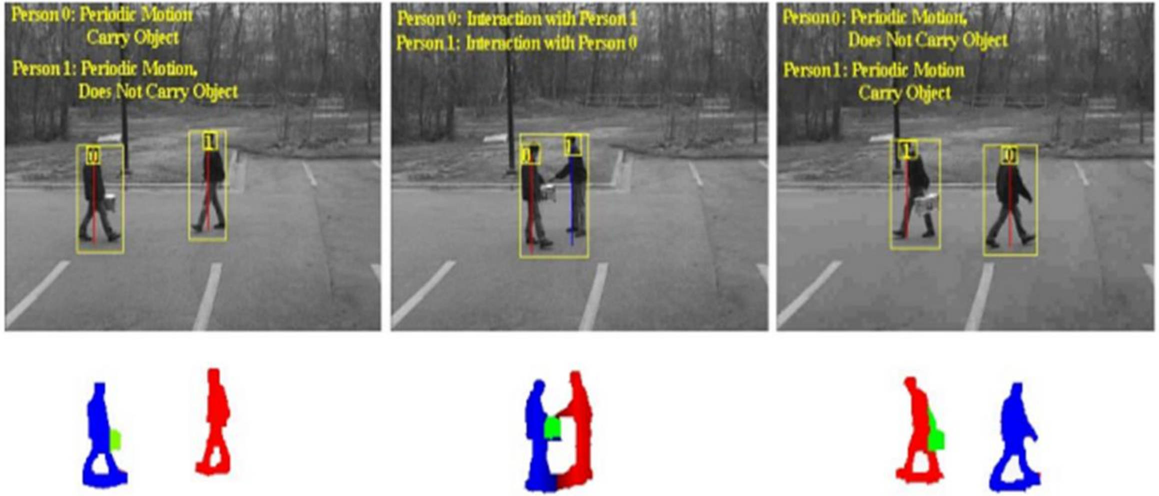
Application of Segmentation

Application

Road identification in Satellite imagery

Application

Activity Recognition



Application

Crop Monitoring

Application

Object Extraction

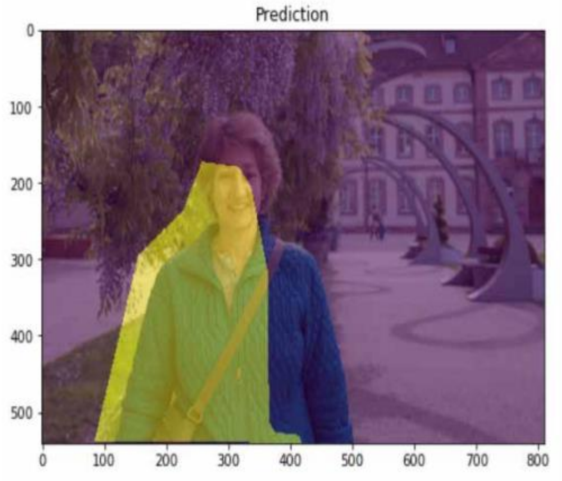
Application

Autonomous Car

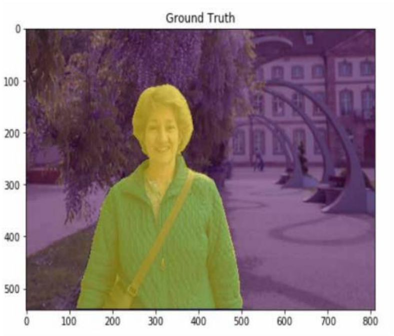
Evaluation

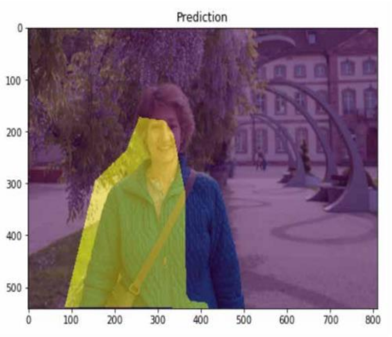
Dalam Segmentasi bertujuan memprediksi kelas setiap piksel dalam gambar.

The Intersection Over Union(IoU) adalah metrik yang memberi tahu nilai(persen) tumpang tindih antara target mask dan output yang di prediksi.

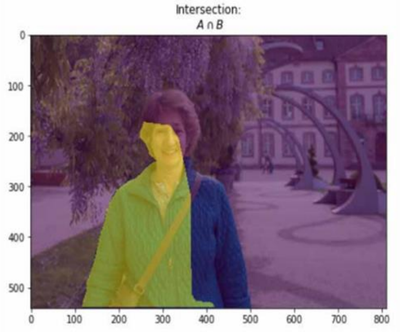


What is intersection over union?





IOU metric measures the pixels common between the target and the predicted masks divided by the total number of pixels present across both masks.

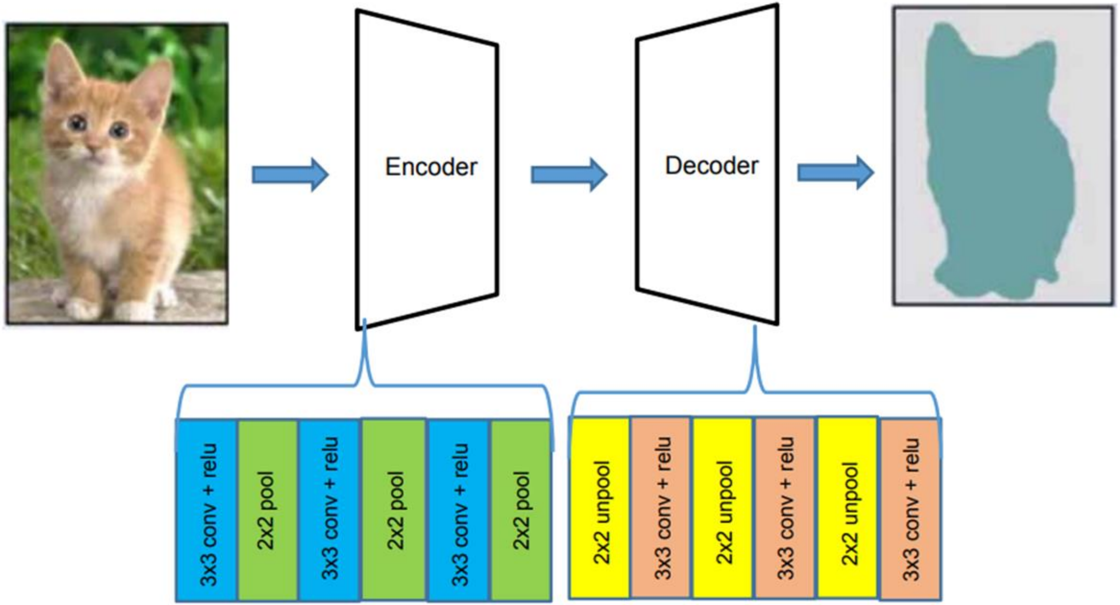


$$IoU = \frac{target \cap prediction}{target \cup prediction}$$

03 SEMANTIC SEGMENTATION

- Encoder Decoder Approach
- Concept of deconvolutions

Encoder decoder architecture

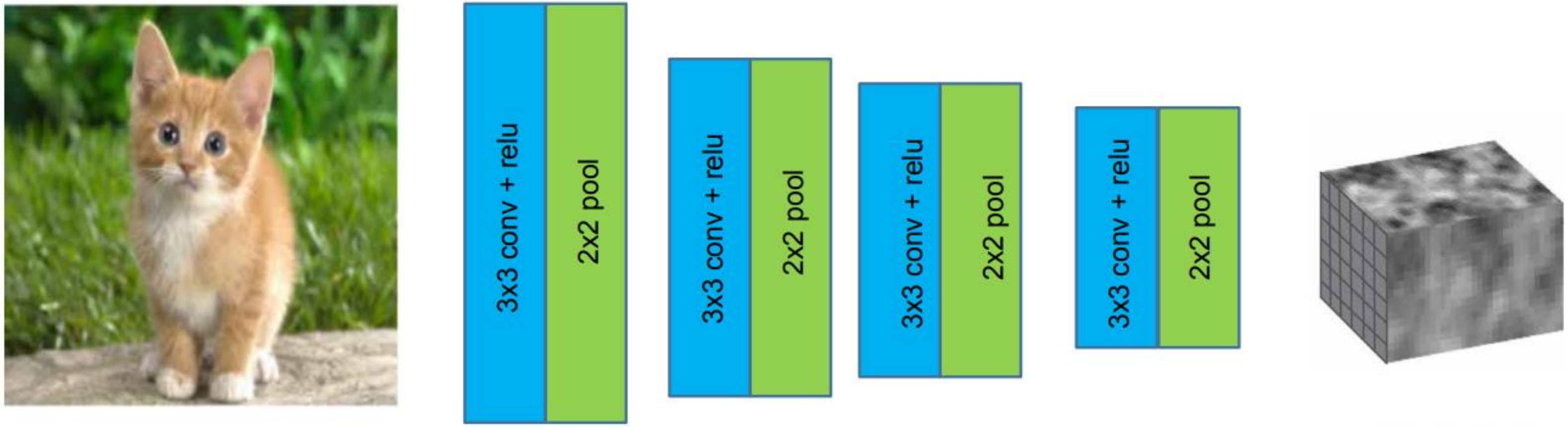


This method produces more accurate image segmentation.

Encoder decoder architecture

- Lapisan encoder dan decoder berukuran sama.
- Encoder digunakan untuk mengekstrak fitur.
- Decoder akan menggunakan fitur yang sudah diektrak untuk membuat mask predict pada original picture.

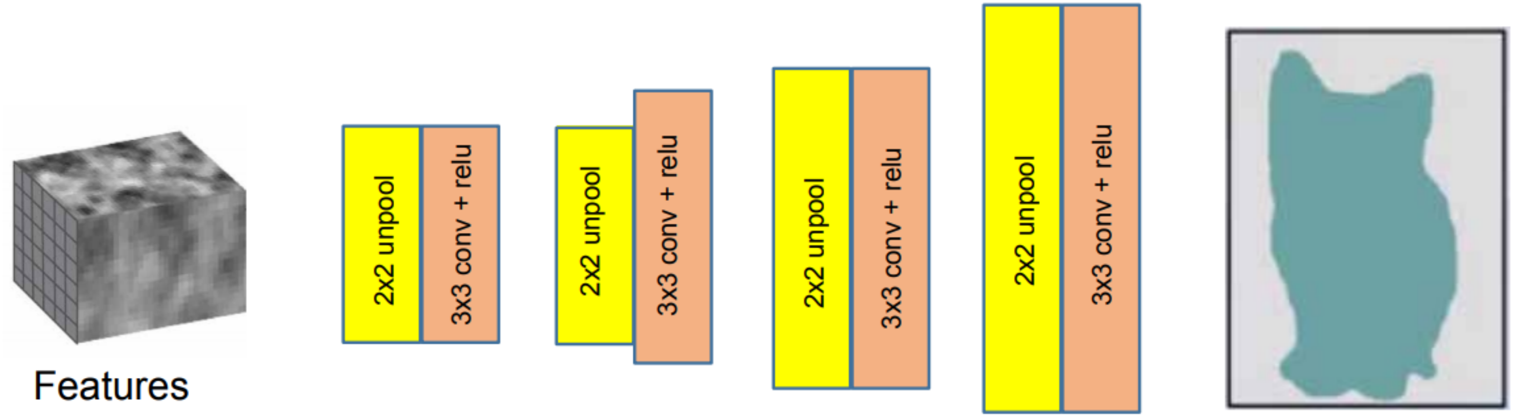
Encoder



Input Image

Features

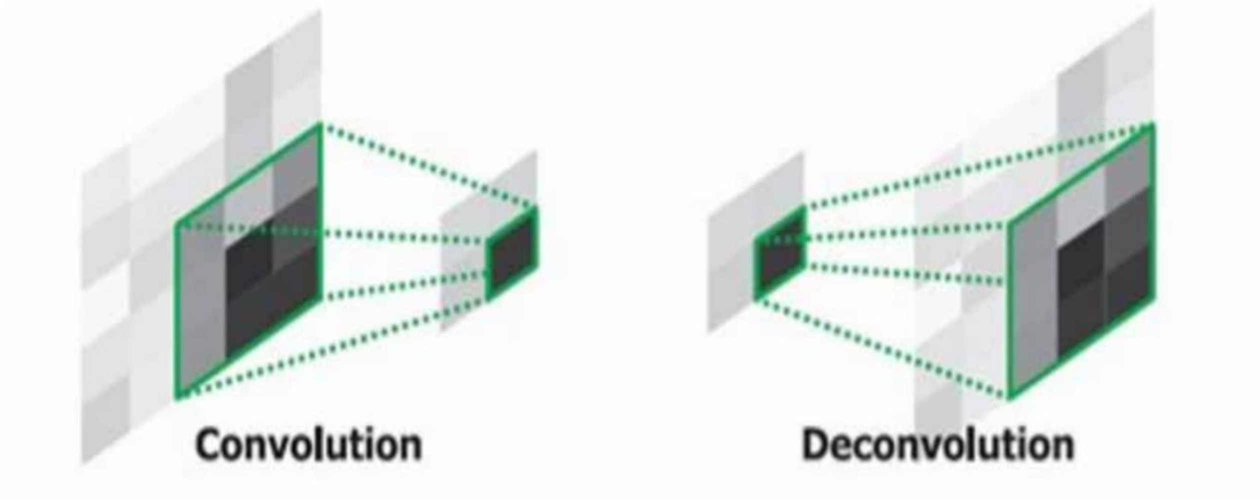
Decoder



Features
2x2 unpool
3x3 conv + relu
2x2 unpool
3x3 conv + relu
2x2 unpool
3x3 conv + relu
2x2 unpool
3x3 conv + relu

What is transpose convolution layer

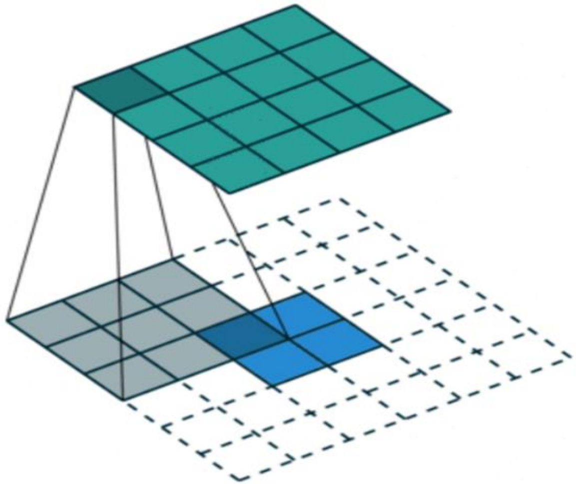
Konvolusi adalah proses untuk mendapatkan ukuran output yang lebih kecil. Deconvolution atau Transpose Convolution adalah proses dimana kita ingin upsample image untuk mendapatkan ukuran yang lebih besar.



How does deconvolution works

- warna biru sebagai input, Warna Grey 3x3 filter, Warna Hijau merupakan outputnya. Stride is 1. Perhitungannya yang dilakukan sama dengan perhitungan konvolusi biasa.

- jadi dengan input 2x2 dan kernel size 3x3, kita bisa mendapatkan ukuran gambar yang lebih besar menjadi 4x4 dengan stride =



Maths behind deconvolution

52	29
181	90

Input Image

0	1	0
0	1	0
0	1	0

Filter

0	52	29	0
0	233	119	0
0	223	119	0
0	181	90	0

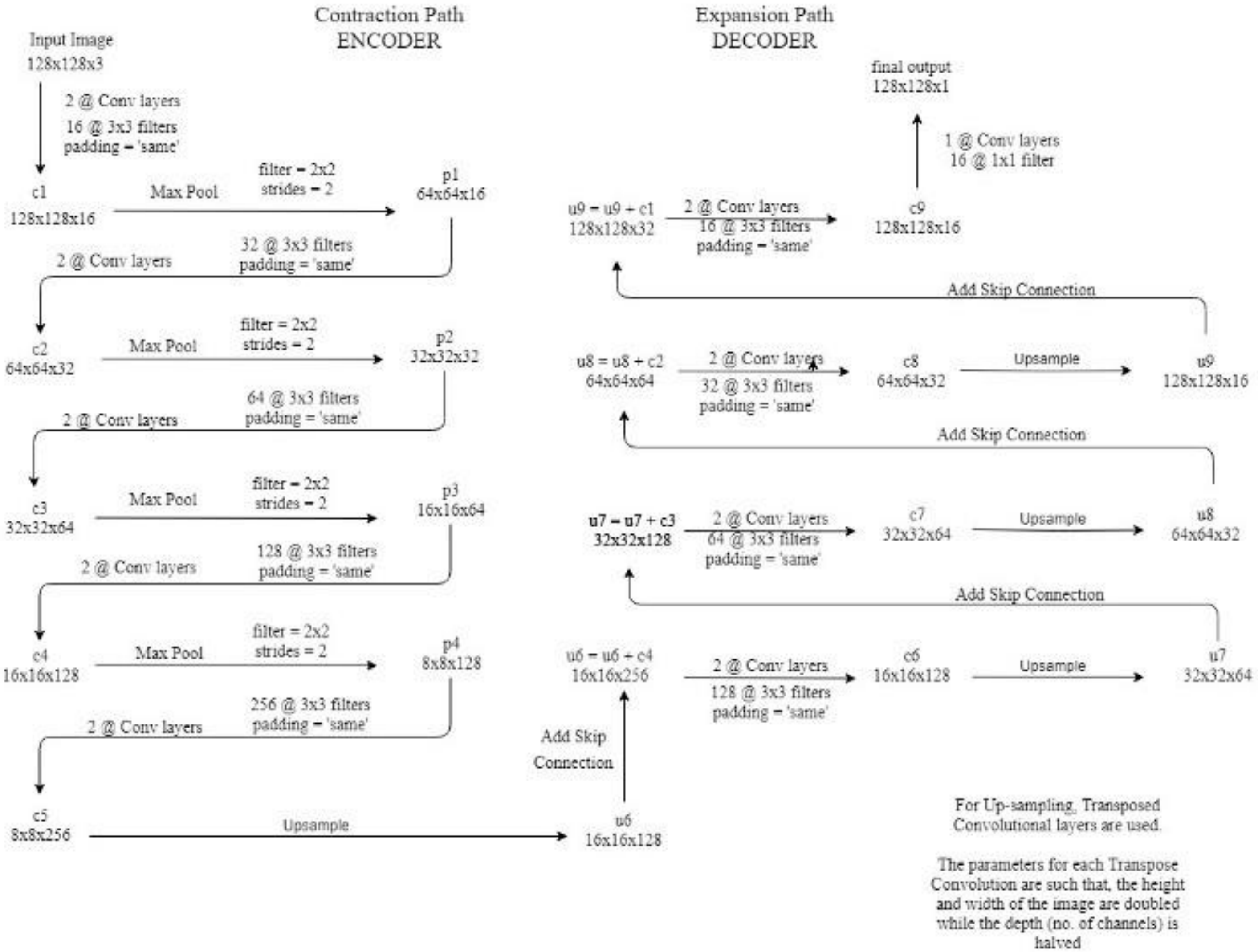
Convoluted Image

This is how we obtain a 4x4 image, when we perform deconvolution on 2x2 image with a 3x3 filter

Orbit FUTURE ACADEMY | Skills For Future Jobs

4

U-Net Architecture



https://arxiv.org/abs/1505.04597

U- Net Training

Soft-max:

$$p_k(x) = \exp(a_k(x)) \,/\, \sum_{k'=1}^{K} \exp(a_{k'}(x))$$

- k- Feature channel
- $a_k(x)$ - The activation in feature channel k at pixel position x

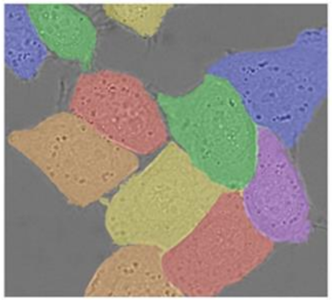
Cross-Entropy loss function:

$$E = -\sum_{x \in \Omega} w(x) log(\mathrm{p}_{\mathrm{l(x)}}(\mathrm{x}))$$

- $w(x)$- True label per a pixel

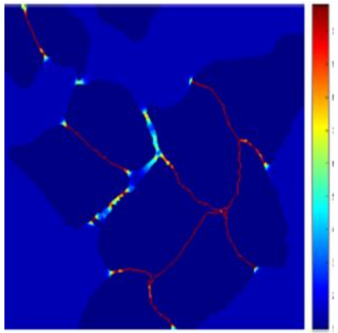
U- Net Training - **pixel-wise loss weight**

- Force the network to learn the small separation borders that they introduce between touching cells.



Colors :different instances

$$\mathbf{w}(\boldsymbol{x}) = \boldsymbol{w}_c(\boldsymbol{x}) + \boldsymbol{w}_0\, \boldsymbol{exp}\left(-\frac{\left(\boldsymbol{d}_1(\boldsymbol{x}) + \boldsymbol{d}_2(\boldsymbol{x})\right)^2}{\mathbf{2}\boldsymbol{\sigma}^2}\right)$$



- $w_c(x)$ - weight map to balance the class frequencies
- d_1/d_2 - Distance to the border of the nearest cell / second nearest cell
- w_0 - 10 , $\sigma \approx$ 5 pixels

Keuntungan vs Kerugian

U-net advantages

- Fleksibel dan dapat digunakan untuk tugas rational image masking task
- Akurasi tinggi (given proper training, dataset, and training time)
- Tidak mengandung lapisan fully connected layers
- Lebih cepat dari sliding-window (1-detik per image)
- Terbukti menjadi tool segmentasi yang sangat bagus dalam skenario dengan data terbatas
- Berhasil mencapai kinerja yang sangat baik pada aplikasi segmentasi biomedis yang berbeda.

U-net disadvantages

- Gambar yang lebih besar membutuhkan memori GPU yang tinggi.
- Membutuhkan banyak waktu untuk berlatih (relatively many layers)
- Pre-trained models tidak tersedia secara luas (it's too task specific)

TERIMA KASIH

Skills For Future Jobs

Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media

Orbit Future Academy
@OrbitFutureAcademyIn1
OrbitFutureAcademy
Orbit Future Academy