

COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL

**Dr. Arnita, S.Si., M.Si, Faridawaty Marpaung, S.Si., M.Si,
Fitrahuda Aulia, Nita Suryani S.Kom, Rinjani Cyra Nabila
S.Kom**



Pustaka Aksara

COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL

Penulis : Dr. Arnita, S.Si., M.Si, Faridawaty Marpaung,
S.Si., M.Si, Fitrahuda Aulia, Nita Suryani
S.Kom, Rinjani Cyra Nabila S.Kom
Editor : Ahmad Baharuddin Surya
Desain Sampul : Dicky Firmansyah
Tata Letak : Silviera Elsa Angelina

ISBN : 978-623-8230-27-3

Diterbitkan oleh : **PUSTAKA AKSARA, 2022**

Redaksi:

Surabaya, Jawa Timur, Indonesia

Telp. 0858-0746-8047

Laman : www.pustakaaksara.co.id

Surel : info@pustakaaksara.co.id

Anggota IKAPI

Cetakan Pertama : 2022

All right reserved

Hak Cipta dilindungi undang-undang

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun dan dengan cara apapun, termasuk memfotokopi, merekam, atau dengan teknik perekaman lainnya tanpa seizin tertulis dari penerbit.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas rahmat dan karunia yang dilimpahkan. Ucap syukur atas selesainya buku yang berjudul “Computer Vision dan Pengolahan Citra Digital” disusun dan diterbitkan. Buku ini disusun untuk membantu mahasiswa dan siapa saja yang sedang mempelajari pengolahan citra dan computer vision. Semoga buku ini dapat memberikan manfaat keilmuan dan menambah wawasan bagi siapa saja yang membaca dan menggunakan buku ini.

Kami menyadari bahwa tak ada yang sempurna, seperti pepatah yang menyebutkan “tak ada gading yang tak retak” karena kesempurnaan hanya milik Allah SWT. Oleh karena itu dengan senang hati dan secara terbuka kami menerima kritikan, masukan dan saran agar materi yang tersaji dalam buku ini menjadi lebih baik kedepannya.

Akhir kata, terima kasih kami ucapkan kepada seluruh pihak yang telah mendukung dan turut andil dalam rangkaian proses penyusunan dan penerbitan buku ini. Semoga buku ini memberi manfaat bagi semua pihak dan dapat memberikan kontribusi pengembangan ilmu pengetahuan di Indonesia dan dunia.

Medan, Oktober 2022

Penulis

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
BAB 1 Pendahuluan.....	1
A. Apa itu <i>Computer Vision</i> ?	1
B. Ikktisar Buku.....	2
BAB 2 Pengantar Pengolahan Citra	3
A. Citra Digital.....	3
B. Aplikasi Citra Digital	4
C. Tipe-tipe File Citra.....	8
D. Prinsip Dasar Pengolahan Citra	9
BAB 3 Pengenalan Dasar Citra	13
A. Pengertian Citra.....	13
B. Kuantisasi Citra	13
C. Kualitas Citra	14
D. Mengenal Jenis Citra	15
BAB 4 Operasi pada Citra	17
A. Operasi Piksel dan Histogram.....	17
B. Operasi Ketetanggaan Piksel.....	20
C. Operasi Geometrik	21
D. Operasi pada Citra Biner	26
BAB 5 Morfologi Pengolahan Citra	29
A. Operasi Dilasi.....	29
B. Operasi Erosi.....	30
C. Proses <i>Opening</i>	31
D. Proses <i>Closing</i>	31
BAB 6 Pengolahan Citra Berwarna	33
A. Dasar dan Ruang Warna.....	33
B. Ruang Warna RGB.	34
C. Ruang Warna XYZ.....	34

D. Ruang Warna CMY/CMYK.....	35
E. Ruang Warna YIQ.....	36
F. Ruang Warna $YCbCr$ dan $YPbPr$	36
G. Ruang Warna HIS, HSV, dan HSL	37
H. Ruang Warna CIELAB.....	38
BAB 7 Segmentasi Citra	39
A. Pengertian Segmentasi Citra	39
B. Deteksi Tepi.....	40
C. Deteksi Garis	41
D. <i>Thresholding</i>	41
E. Segmentasi Warna.....	42
BAB 8 Restorasi dan Peningkatan Citra	43
A. Restorasi Citra	43
B. <i>Noise</i> pada Citra.....	44
C. Jenis <i>Noise</i> pada Citra.....	45
D. Menghilangkan <i>Noise</i> pada Citra.....	45
BAB 9 Ekstraksi Fitur Citra	47
A. Fitur Bentuk.....	47
B. Fitur Kontur.....	48
C. Fitur Tekstur	49
D. Fitur Warna.....	51
E. Topologi.....	51
BAB 10 Pengantar <i>Computer Vision</i>.....	53
A. Pendahuluan.....	53
B. Pengertian <i>Computer Vision</i>	53
C. Cara Kerja <i>Computer Vision</i>	55
BAB 11 Perkembangan <i>Computer Vision</i>.....	60
A. Sejarah Perkembangan <i>Computer Vision</i>	60
B. Penerapan <i>Computer Vision</i>	61
BAB 12 <i>Machine Learning</i> dalam <i>Computer Vision</i>	65

A. Pengantar <i>Artificial Intelligence</i> (AI).....	65
B. Pengantar <i>Machine Learning</i>	67
C. Klasifikasi <i>Machine Learning</i>	69
BAB 13 Teknik dan Algoritma dalam <i>Computer Vision</i>	75
A. <i>K-Nearest Neighbors</i>	75
B. <i>Naïve Bayes</i>	78
C. <i>Support Vector Machine</i> (SVM).....	79
D. <i>Decision Tree dan Random Forest</i>	80
E. <i>Clustering</i>	82
F. <i>Manifold Learning</i>	85
G. <i>Convolutional Neural Network</i> (CNN).....	86
BAB 14 <i>Computer Vision Dataset</i>	93
A. Membuat Dataset.....	93
B. <i>Pre-processing Dataset</i>	97
C. <i>Splitting Dataset</i>	98
BAB 15 Deteksi dan Pencocokan Fitur	100
A. Deteksi Tepi	100
B. Deteksi Kontur.....	102
C. Teknik Segmentasi.....	102
BAB 16 <i>Object Detection & Recognition</i>.....	106
A. Klasifikasi Citra.....	106
B. Pencarian Gambar	113
C. <i>Face Recognition</i>	115
D. Deteksi Visual 3D	121
E. Deteksi Video dan <i>Motion</i>	122
BAB 17 Teknik Deteksi Objek	124
A. Detektor Objek Dua Tahap dan Satu Tahap	124
B. R-CNN (<i>Region-based CNN</i>).....	127
C. Fast R-CNN.....	129
D. Faster R-CNN.....	131
E. Mask R-CNN.....	134

F. R-FCN (<i>Region-based Fully Convolutional Network</i>)	138
G. YOLO (<i>You Only Look Once</i>).....	139
H. SSD (<i>Single Shot Detector</i>).....	147
I. RetinaNet.....	149
DAFTAR PUSTAKA	150
GLOSARIUM.....	165
PROFIL PENULIS.....	168

COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL

**Dr. Arnita, S.Si., M.Si, Faridawaty Marpaung, S.Si., M.Si,
Fitrahuda Aulia, Nita Suryani S.Kom, Rinjani Cyra Nabila
S.Kom**

BAB 1

Pendahuluan

A. Apa itu *Computer Vision*?

Computer Vision menjadi semakin penting dan efektif dalam beberapa tahun terakhir karena aplikasinya yang luas di berbagai bidang seperti pengawasan dan pemantauan cerdas (smart surveillance), kesehatan dan kedokteran, olahraga dan rekreasi, robotika, *drone*, mobil *self-driving*, dan lain sebagainya. Tugas pengenalan visual, seperti klasifikasi gambar, pelokalan, dan deteksi, adalah inti dari pemanfaatan aplikasi ini yang mana telah menghasilkan kinerja yang mumpuni dalam tugas dan sistem pengenalan visual.

Penerapan *computer vision* dapat ditemui hampir di setiap aspek kehidupan. Contoh sederhananya adalah deteksi pengguna masker di dalam ruangan, aplikasi *scanner* rumus matematika, estimasi tinggi badan melalui kamera, dan lain sebagainya. Hal ini menunjukkan bahwa visi komputer semakin efisien di dalam genggamannya. Cara kerjanya yang mudah dipahami yakni gambar digital 2-D yang ditangkap oleh sistem visi komputer terdiri dari piksel, area yang diproyeksikan, perimeter, atau fitur panjang dan lebar dapat diukur dalam gambar dengan menggunakan algoritma *image processing*.

Pembahasan visi komputer erat kaitannya dengan pengolahan citra (*image processing*) yang mana merupakan proses pengolahan dan analisis dengan ciri data masukan (*input*) dan informasi keluaran (*output*) yang berbentuk citra. Akan tetapi citra yang dihasilkan dari proses pengolahan ini memiliki kualitas yang lebih dibandingkan dengan citra aslinya baik karena sudah diolah dengan metode tertentu untuk mendapatkan hasil sesuai kebutuhan.

Computer Vision dan *Machine Learning* telah memainkan peran penting bersama dalam pengembangan berbagai aplikasi berbasis gambar dalam dekade terakhir. Selama ini, teknologi berbasis visi telah berubah dari hanya modalitas penginderaan menjadi sistem komputasi cerdas yang dapat memahami dunia

nyata. Dengan demikian, memperoleh pengetahuan visi komputer dan *machine learning (deep learning)* adalah keterampilan penting yang diperlukan di banyak bisnis inovatif modern dan kemungkinan akan menjadi lebih penting dalam waktu dekat.

B. Ikktisar Buku

Dalam buku ini kami mencoba memberikan gambaran sekilas tentang beragam bidang teknik analisis informasi visual. Bagaimana gambar digital diambil dengan perangkat elektronik yang dapat disimpan dan diolah. Beragam data citra yang diperoleh memiliki tipe file, elemen warna, dan komposisi yang berbeda. Dalam buku ini akan dibahas bagaimana suatu citra dapat diolah, dioperasikan, disegmenentasi, maupun rstorasi citra untuk menyempurnakan gambar sehingga gambar yang dihasilkan memiliki kualitas visual yang lebih baik, bebas dari *noise*. Peningkatan citra, penyaringan, dan restorasi telah menjadi beberapa aplikasi penting dari pengolahan citra karena akan mempengaruhi kredibilitas data citra itu sendiri. Selanjutnya dibahas pula bagaimana suatu mesin dapat belajar dari cara manusia melihat dunia 3D dalam waktu *realtime*. Untuk melakukan analisa citra dan video oleh komputer untuk memperoleh informasi dan pemahaman dari suatu obyek diperlukan proses pelatihan (*training*) dari data yang sudah dikumpulkan. Pada tahap akhir proses akan tercipta model yang nantinya akan bermanfaat baik dalam klasifikasi maupun deteksi citra. Kemajuan penerapan visi komputer dalam berbagai aspek kehidupan juga akan dibahas dalam buku ini.

BAB 2

Pengantar Pengolahan Citra

Pemrosesan dan pengolahan citra adalah metode untuk melakukan beberapa operasi pada gambar, untuk mendapatkan gambar yang disempurnakan atau untuk mengekstrak beberapa informasi yang berguna di dalamnya. Merupakan jenis pemrosesan sinyal di mana input adalah gambar dan output dapat berupa gambar atau karakteristik / fitur yang terkait dengan gambar itu. Saat ini, pemrosesan gambar adalah salah satu teknologi yang berkembang pesat dan membentuk area penelitian inti dalam disiplin ilmu teknik dan ilmu komputer. Pada bab ini akan dibahas tentang konsep citra, aplikasi citra digital dan tipe-tipe datanya hingga prinsip dasar citra digital.

A. Citra Digital

Menurut Iriyanto (2014), Citra atau *image* adalah suatu cahaya pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Citra digital sangat populer pada masa kini yang biasanya juga disebut dengan gambar. Banyak alat elektronik yang dibuat khusus untuk menghasilkan citra digital. Contohnya adalah *scanner*, kamera digital, mikroskop digital, *fingerprint reader*, kamera pemantau, dan lain sebagainya. Kualitas citra yang dihasilkan oleh alat-alat tersebut tidak sama, hal tersebut dikarenakan penyesuaian hasil gambar terhadap kebutuhan. Citra digital dapat diolah dengan bantuan perangkat lunak (*software*) tertentu untuk mengubah atau memanipulasi citra agar memiliki kualitas yang lebih baik. Contohnya adalah *Adobe Photoshop* dan *GIMP (GNU Image Manipulation Program)* menyajikan beragam fitur untuk memanipulasi citra digital.

Pada umumnya, istilah pengolahan citra digital berarti memproses gambar dua dimensi menggunakan komputer. Pengolahan yang dimaksud yaitu gradasi, memodifikasi,

maupun memanipulasi citra untuk tujuan tertentu. Contoh sederhananya adalah saat mengambil gambar dengan kamera ponsel, hasil yang didapat bisa diolah dengan aplikasi editor yang ada di ponsel. Mulai dari memberi filter, meningkatkan kecerahan, menambahkan objek, hingga memisahkan foto dari latar belakang. Gambaran tersebut adalah contoh sederhana yang dilakukan dalam pengolahan citra digital di kehidupan sehari-hari.



Gambar 1. Contoh media citra digital

B. Aplikasi Citra Digital

Pengolahan citra merupakan bagian penting yang mendasari berbagai aplikasi nyata, seperti pengenalan pola, penginderaan jarak-jauh melalui satelit atau pesawat udara, dan *machine vision*. Ada banyak contoh nyata aplikasi teknologi yang memanfaatkan citra digital. Yakni sebagai berikut:

1. Sistem Inspeksi Visual Otomatis

Sistem inspeksi visual otomatis sangat penting untuk meningkatkan produktivitas dan kualitas produk di industri manufaktur. Seperti Inspeksi otomatis filamen lampu pijar yang melibatkan inspeksi proses pembuatan bohlam. Sering kali filamen bohlam digunakan setelah durasi yang singkat karena geometri filamen yang salah, misalnya, ketidakteraturan nada kabel di lampu. Inspeksi manual tidak efisien untuk mendeteksi penyimpangan tersebut sehingga diberikan solusi alternatif dengan pemanfaatan visi computer. Contoh lainnya adalah identifikasi komponen yang salah biasanya mengidentifikasi komponen yang rusak

dalam sistem elektronik atau elektromekanis. Komponen yang rusak biasanya menghasilkan lebih banyak energi panas. Gambar infra-merah (IR) dapat dihasilkan dari distribusi energi panas di rakitan. Begitu pula untuk sistem pemeriksaan permukaan otomatis yang dapat mendeteksi cacat pada permukaan adalah persyaratan penting di banyak industri logam. Misalnya, di pabrik penggilingan panas atau dingin di pabrik baja, diperlukan untuk mendeteksi penyimpangan pada permukaan logam yang digulung. Ini dapat dicapai dengan menggunakan teknik pemrosesan gambar seperti deteksi tepi, identifikasi tekstur, analisis fraktal, dan sebagainya.

2. Interpretasi Jarak Jauh

Informasi mengenai sumber daya alam, seperti pertanian, hidrologi, mineral, hutan, sumber daya geologi dapat diekstraksi berdasarkan analisis citra penginderaan jauh. Untuk analisis pemandangan penginderaan jauh, gambar permukaan bumi ditangkap oleh sensor di satelit penginderaan jauh atau oleh pemindai multi-spektral yang ditempatkan di pesawat terbang dan kemudian ditransmisikan ke stasiun bumi untuk diproses lebih lanjut. Hal ini sangat menguntungkan dalam segi efisiensi waktu maupun biaya yang mungkin dikeluarkan serta menghindari resiko bahaya yang timbul bila mengunjungi lokasi tersebut secara langsung. Contohnya adalah pemantauan kondisi es di kutub utara melalui citra satelit.

3. Citra Biomedis

Di dunia kesehatan berbagai jenis perangkat pencitraan seperti *X-Ray*, *CT-scan*, *ultrasound*, *MRI*, dan sejenisnya digunakan secara luas untuk tujuan diagnosis medis. Penerapan ini dinilai sangat efektif dalam melokalisasi objek seperti organ manusia tanpa harus melakukan operasi. Selain itu dapat pula dilakukan pengukuran objek yang diekstraksi, misalnya penyakit tumor.

4. Pengawasan dan Keamanan

Penerapan teknik pemrosesan citra dalam pengawasan pertahanan merupakan bidang studi yang penting. Dalam mempertahankan keamanan dan kedaulatan wilayah negara terdapat kebutuhan berkelanjutan untuk memantau daratan dan lautan dengan menggunakan teknik surveilans acrial. Dalam pengamanan zona wilayah kelautan, perlu dilakukan pemantauan kapal-kapal yang memasuki wilayah. Tugas utamanya di sini adalah mengelompokkan objek yang berbeda di bagian badan permukaan air dari gambar yang diperoleh setelah mengekstrak lokasi, bentuk, panjang, lebar, untuk mengklasifikasikan objek yang disegmentasi. Contoh sederhana lainnya adalah kamera pengintai yang dipasang di rumah maupun perkantoran untuk menjaga keamanan.

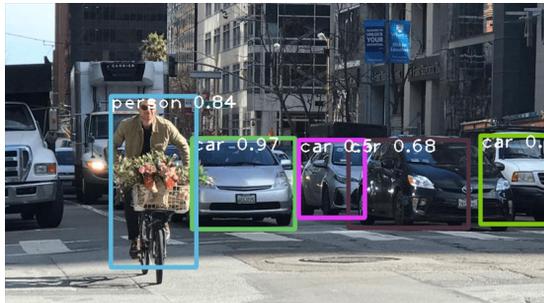
5. Pengambilan Gambar Berbasis Konten

Saat melakukan *scan* terhadap suatu dokumen untuk memperoleh data yang tertulis di dalamnya termasuk pengolahan citra digital. Dari hasil scan dapat mengimplementasikan alat pencarian untuk mengindeks dan mengambil informasi yang ada di dalamnya. Sejumlah mesin pencari yang baik tersedia saat ini untuk mengambil teks dalam bentuk yang dapat dibaca mesin, Fitur gambar digital (seperti bentuk, tekstur, warna, topologi objek, dll.) juga dapat digunakan sebagai kunci indeks untuk pencarian dan pengambilan informasi bergambar dari database gambar besar seperti yang diterapkan oleh *Google Lens*. Pengambilan gambar berdasarkan konten gambar seperti itu secara populer disebut pengambilan gambar berbasis konten

6. Pelacakan Objek Bergerak

Pelacakan objek bergerak, untuk mengukur parameter gerak dan memperoleh rekaman visual objek bergerak, merupakan area aplikasi penting dalam pemrosesan gambar. Secara umum ada dua pendekatan berbeda untuk pelacakan

objek yaitu pelacakan berbasis pengenalan dan pelacakan berbasis gerak. Sistem untuk melacak target cepat (misalnya pesawat militer, rudal, dll.) dikembangkan berdasarkan teknik prediksi berbasis gerakan. Dalam sistem pelacakan objek berbasis pemrosesan gambar otomatis, objek target yang memasuki bidang pandang sensor diperoleh secara otomatis tanpa campur tangan manusia. Dalam pelacakan berbasis pengenalan, pola objek dikenali dalam bingkai gambar yang berurutan dan pelacakan dilakukan menggunakan informasi posisinya, misalnya adalah kamera yang mendeteksi apakah seseorang di dalam ruangan menggunakan masker atau tidak.



Gambar 2. Deteksi objek citra berdasarkan bentuk.

7. Kompresi Gambar dan Video

Pengembangan teknologi kompresi untuk gambar dan video paling sering ditemui dalam aplikasi multimedia. Walaupun sudah banyak penyedia layanan penyimpanan seperti *cloud*, kebutuhan penyimpanan data gambar dan video dalam ukuran kecil tetap diperlukan. Kendala yang dialami dengan gambar maupun video tanpa kompresi adalah saat melakukan siaran *real-time*. Teknik kompresi gambar dan video pada dasarnya mengurangi kualitas data dengan mereduksi bingkai gambar dengan jumlah bit yang jauh lebih kecil dan karenanya mengurangi persyaratan untuk penyimpanan dan *bandwidth* komunikasi yang efektif.

C. Tipe-tipe File Citra

File citra memiliki format yang beragam. Berikut disajikan beberapa standar format file gambar yang paling populer digunakan (Gonzalez & Woods, 2008).

1. *Tagged Image Format* (.tif, .tiff). Merupakan format yang sangat luas, yang dapat menangani mulai dari *bitmap* hingga gambar palet warna terkompresi. Format tiff mendukung beberapa skema kompresi, tetapi sering digunakan untuk gambar yang tidak dikompresi juga. Format ini populer, relatif sederhana, dan memiliki warna.
2. JPEG (.jpg). Adalah format standar yang paling banyak digunakan untuk transmisi informasi bergambar. Baik JPEG dan JPG merupakan gambar raster, yakni gambar yang bukan merupakan vector. Ciri khas yang dimilikinya seperti memiliki *gradasi* warna atau bayangan yang halus dan garis dan bentuk yang sangat tidak jelas. Karena sifatnya yang berbasis piksel, gambar jenis ini mengalami masalah kualitas saat ukurannya diperbesar (menjadi bergerigi dan mengekspos piksel individu). Biasanya didefinisikan dan ditampilkan pada satu resolusi tertentu, yang diukur dalam titik per inci (dpi). Semakin tinggi dpi sebuah gambar raster semakin jernih pula gambar tersebut. Gambar raster adalah standar *Web*, artinya lebih mudah ditemukan di internet.
3. *Portable Network Graphics* (.png). Format file PNG menyediakan *penyimpanan* gambar raster tanpa kehilangan dan terkompresi dengan baik. Format sederhana ini mencakup fungsi utama .tiff. Skala abu-abu, palet warna, dan mendukung gambar dengan warna asli.
4. MPEG (.mpg). Format ini banyak digunakan di seluruh Web dan hanya *digunakan* untuk gambar bergerak. Tipe file ini juga biasanya melalui proses kompresi.
5. *Graphics Interchange Format* (.gif). Format ini mendukung gambar palet *warna* 8-bit yang portabel di berbagai platform dan aplikasi. Sebagian besar digunakan untuk logo dan gambar berkualitas tajam. File GIF juga mampu berisi data gambar animasi.

6. RGB (.rgb). Merupakan standar file gambar dari Silicon Graphics *untuk* gambar berwarna.
7. RAS (.ras). Adalah citra hasil *scan* tanpa kompresi dari tiga pita warna *untuk* gambar *Sun Raster*.
8. Postscript (.ps, .eps, .epsf). Format gambar ini terutama *digunakan* saat memperkenalkan gambar atau gambar dalam buku atau catatan dan untuk pencetakan. Dalam format *postscript*, gambar tingkat abu-abu diwakili oleh angka desimal atau heksadesimal yang dikodekan dalam ASCII.
9. *Portable Image*. Mencakup *Bitmap Portabel*, *Graymap Portabel*, *Pixmap Portabel*, dan *Network Map Portabel*. Adapun ekstensi default format ini adalah .pbm, .pgm, .ppm, dan .pnm. Format ini mendukung semua jenis gambar dengan kompleksitas yang meningkat mulai dari bit hingga tingkat abu-abu hingga *pixmap* warna dari berbagai jenis.
10. PPM. File PPM terdiri dari dua bagian yaitu *header* dan data gambar. Meskipun tipe ini sudah jarang ditemukan akhir-akhir ini, kita masih dapat menemukannya dengan mencari file ekstensi PPM di *internet*. Format PPM muncul pada akhir 1980-an untuk mempermudah berbagi gambar antar platform yang berbeda.
11. PGM. Format ini identik dengan PPM perbedaannya adalah PGM menyimpan informasi skala abu-abu, yaitu satu nilai per piksel bukan tiga (RGB).
12. PBM. PBM Format bitmap portabel adalah format file monokrom *paling* rendah. Awalnya dirancang sebagai bahasa umum dari keluarga besar filter konversi gambar *bitmap*.

D. Prinsip Dasar Pengolahan Citra

1. Kecerahan dan kontras

Gambar yang diambil oleh lensa kamera terlihat gelap karena kurangnya intensitas cahaya dapat diperbaiki dengan pengolahan citra berupa peningkatan kontras. Cara kerjanya yaitu dengan meningkatkan persepsi objek dalam pemandangan dengan meningkatkan perbedaan kecerahan antara objek dan latar belakangnya. Peningkatan kontras

biasanya dilakukan sebagai peregangan kontras diikuti oleh peningkatan nada, meskipun keduanya dapat dilakukan dalam satu langkah. Pada Gambar 3 di bawah dapat dilihat perbedaan sebuah gambar sebelum dan setelah mengalami perubahan kontras.



Gambar 3. Peningkatan kontras pada citra

2. Penghilangan *noise*

Gambar yang akan diproses seringkali dalam keadaan terdistorsi atau mengandung derau (*noise*). Untuk kepentingan tertentu, derau tersebut perlu dibersihkan terlebih dulu dengan cara mengurangi atau menghilangkan visibilitasnya melalui proses menghaluskan seluruh gambar meninggalkan area di dekat batas kontras. Proses ini diperlukan untuk mendapatkan gambar dengan kualitas lebih baik dan bebas dari objek yang mengganggu.



Gambar 4. Penghilangan *Noise* pada citra

3. Deteksi tepi objek

Deteksi pinggir atau *edge detection* merupakan teknik yang sering digunakan pada bidang pengolahan citra dan

juga *computer vision*, terutama pada bidang *features detection* dan *feature extraction*. objek perlu dipisahkan terlebih dulu dari latar belakangnya. Salah satu pendekatan yang umum dipakai untuk keperluan ini adalah penemuan batas objek (Iriyanto & Zaini, 2014). Deteksi tepi (*edge detection*) adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra. Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Dalam hal ini, batas objek berupa bagian tepi objek dalam suatu citra digital, tepi atau *edge* adalah sebuah kurva yang mengikuti yang berubah dengan cepat berdasarkan intensitasnya. Deteksi tepi digunakan untuk mengidentifikasi tepi dalam citra digital.

Pelacakan tepi merupakan operasi untuk menemukan perubahan intensitas lokal yang berbeda dalam sebuah citra. Gradien adalah hasil pengukuran perubahan dalam sebuah fungsi intensitas, dan sebuah citra dapat dipandang sebagai kumpulan beberapa fungsi intensitas kontinu sebuah citra. Perubahan mendadak pada nilai intensitas dalam suatu citra dapat dilacak menggunakan perkiraan diskrit pada gradien. Gradien disini adalah kesamaan dua dimensi dari turunan pertama dan didefinisikan sebagai vektor (Lusiana 2019). Oleh karena itu teknik deteksi tepi sering digunakan sebagai dasar teknik segmentasi untuk proses segmentasi yang lain

Ada beberapa metode yang terkenal dan banyak digunakan untuk pendektasian tepi di dalam citra, yaitu operator Robert, operator Prewitt dan operator Sobel. Metode Sobel II-2 paling banyak digunakan sebagai pelacak tepi karena kesederhanaan dan keampuhannya. Kelebihan dari metode ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi. Masing-masing metode deteksi memiliki sub metode yang cukup banyak, tetapi metode deteksi citra yang baik adalah metode

yang dapat mengeliminasi derau (noise) yang semaksimal mungkin.

Metode deteksi tepi-tepi lain yang sering digunakan adalah *Canny Edge Detection*. Metode Canny berbeda dari metode deteksi tepi yang lain karena menggunakan dua ambang batas yang berbeda (untuk mendeteksi tepi yang jelas dan tepi yang tidak jelas. Pada Gambar 5 di bawah adalah contoh sederhana deteksi tepi dari sebuah gambar bunga.



Gambar 5. Deteksi tepi pada citra

BAB 3

Pengenalan Dasar Citra

Sebuah gambar tidak lebih dari sinyal dua dimensi yang didefinisikan oleh fungsi matematika $f(x,y)$ di mana x dan y adalah dua koordinat horizontal dan vertikal. Bab 3 menguraikan pembahasan tentang pengertian citra, kuantisasi dan kualitas citra, serta jenis-jenisnya.

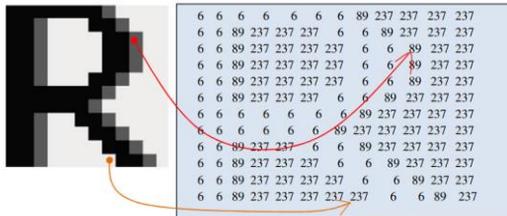
A. Pengertian Citra

Citra digital adalah kumpulan titik yang dinamakan piksel (*pixel* atau *picture element*). Setiap piksel digambarkan sebagai satu kotak kecil yang memiliki koordinat posisi. Citra, atau *image* adalah fungsi dua dimensi $f(x,y)$ yang didapatkan dari peralatan sensor yang mencatat harga ciri citra pada semua titik elemen (x,y) . Sebuah citra digital a $[m,n]$ dapat dijelaskan sebagai ruang diskrit berdimensi-dua yang berasal dari sebuah citra analog a (x,y) menjadi citra kontinu 2D melalui proses *sampling* yang sering disebut sebagai digitalisasi. Jika citra 2D kontinyu a (x,y) dibagi menjadi N baris dan M kolom, maka perpotongan baris dan kolom disebut piksel. Nilai yang diberikan ke integer koordinat $[m,n]$ dengan $\{m = 0,1,2, \dots, M-1\}$ dan $\{n = 0,1,2, \dots, N-1\}$ adalah $[m,n]$. Dalam banyak hal suatu citra (x,y) , kita dapat menganggap sebagai sinyal fisik sensor 2D yang sebenarnya adalah fungsi dari banyak variabel termasuk kedalaman (z), warna (λ), dan waktu (t) (Gonzalez & Woods, 2008).

B. Kuantisasi Citra

Citra digital dibentuk melalui pendekatan yang disebut kuantisasi. Kuantisasi adalah Konversi intensitas piksel analog sampel ke bilangan bulat bernilai diskrit. Kuantisasi melibatkan penetapan nilai tunggal untuk setiap sampel sedemikian rupa sehingga gambar yang direkonstruksi dari nilai sampel terkuantisasi memiliki kualitas yang baik dan kesalahan yang ditimbulkan karena kuantisasi kecil. Rentang dinamis nilai yang

dapat diasumsikan oleh sampel gambar dibagi menjadi sejumlah interval yang terbatas, dan setiap interval diberi satu level. Bila jumlah tingkat kuantisasi meningkat, maka gambar terkuantisasi akan mendekati gambar asli yang bernilai kontinu baik dan kemungkinan kesalahan kuantisasi yang lebih kecil.



Gambar 6. Penyusun nilai citra

C. Kualitas Citra

Beberapa hal yang mempengaruhi kualitas citra adalah intensitas kecerahan dan jumlah piksel yang digunakan untuk menyusun suatu citra. Istilah resolusi citra biasa dinyatakan dengan banyaknya jumlah piksel pada satuan gambar yang memiliki lebar dan tinggi. Resolusi piksel biasa dinyatakan dengan notasi $m \times n$, dengan m menyatakan tinggi dan n menyatakan lebar dalam jumlah piksel.

Kualitas gambar dapat ditingkatkan. Cara terbaik adalah meningkatkan radiasi objek dengan menggunakan sumber cahaya yang lebih kuat atau desain pengaturan iluminasi yang lebih baik. Dengan meningkatkan penguatan, kecerahan dan resolusi gambar meningkat memiliki efek samping yaitu tingkat kebisingan yang juga meningkat. Pada Gambar 7 dapat dilihat perbedaan jumlah piksel pada citra bedampak signifikan pada kualitas citra. Citra dengan piksel piksel rendah terlihat *blur*.



Gambar 7. Perbandingan citra 300 dpi (kiri) dan 72 dpi (kanan)

D. Mengenal Jenis Citra

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra. Ketiga jenis citra tersebut yaitu citra berwarna, citra berskala keabuan, dan citra biner.

1. Citra berwarna

Citra berwarna dikenal dengan istilah *truecolor* atau citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (*red*/merah), G (*green*/hijau), dan B (*blue*/biru) yang jika digabungkan akan membentuk suatu susunan warna yang luas. Setiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 hingga 255). Oleh karena itu, kemungkinan warna yang bisa disajikan mencapai $255 \times 255 \times 255$ atau 16.581.375 warna. Satu jenis warna, dapat dimisalkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika. Koordinatnya dinyatakan dalam bentuk x , y , dan z . Misalkan sebuah vektor dituliskan sebagai $r = (x, y, z)$, maka untuk setiap citra, komponen-komponen tersebut digantikan oleh komponen RGB. Dapat dituliskan sebagai berikut untuk citra berwarna = RGB(30,75,255). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB(0,0,0).

2. Citra berskala keabuan

Citra ini memiliki gradasi hitam dan putih hingga menghasilkan warna abu. Biasa juga dikenal dengan istilah *gray scale* atau *gray level*. Pada jenis gambar ini piksel

bergantung dengan intensitasnya. Kelas *array* dalam citra keabuan bisa terdiri dari *single*, *double*, *uint8*, maupun *int16*. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih.

3. Citra biner

Citra biner juga dikenal dengan istilah *binary bilevel image*. Piksel dalam citra biner hanya dapat mengasumsikan dua kemungkinan nilai, 0 atau 1. Dimana nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Gambar biner dinilai paling 'murah' karena menggunakan penyimpanan dan juga persyaratan pemrosesan yang paling sedikit. Contoh gambar biner adalah gambar garis, teks tercetak pada halaman putih, atau siluet. Gambar-gambar ini berisi informasi yang cukup tentang objek dalam gambar dan kita dapat mengenalinya dengan mudah. Dalam *computer vision* citra biner biasa digunakan untuk pengenalan objek, dan pelacakan, Citra tingkat keabuan atau *grayscale* dapat diubah menjadi citra biner dengan proses *thresholding*.

BAB 4

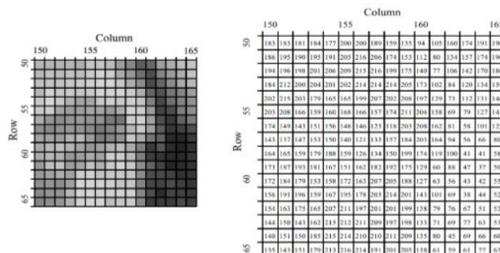
Operasi pada Citra

Dalam aplikasi yang berhubungan dengan gambar visual yang direpresentasikan secara digital, berbagai bentuk pemrosesan umumnya diperlukan sebelum hasilnya siap untuk ditampilkan. Meskipun banyak metode yang digunakan rumit, semuanya berakar pada sejumlah kecil konsep dan teknik dasar. Bab ini membahas operasi-operasi pada citra seperti Operasi Pikel dan Histogram, Operasi Ketetangaan Pikel, Operasi Geometrik, dan Operasi pada Citra Biner.

A. Operasi Pikel dan Histogram

1. Struktur Citra Digital

Seperti yang sudah dijelaskan sebelumnya bahwasannya sebuah gambar digital dapat dijelaskan oleh array dua dimensi dari elemen kecil yang disebut pikel Nilai setiap pikel berhubungan dengan energi yang diterima perangkat pencitraan kita dari sumber radiasi. Salah satu metode umum untuk merepresentasikan citra digital adalah dengan menyimpan setiap nilai pikel sebagai nomor N-bit yang tidak bertanda. Misalnya, gambar *grayscale* menggunakan 8 bit untuk mewakili nilai pikel. Dalam hal ini, nilai setiap pikel dapat bervariasi dari 0 yang mewakili hitam hingga 255 yang terkait dengan putih. Sehingga nilai setiap pikel suatu citra sesuai dengan intensitas warna abu yang berbeda-beda. Gambar 8 menunjukkan suatu citra berskala abu dengan struktur pikselnya.



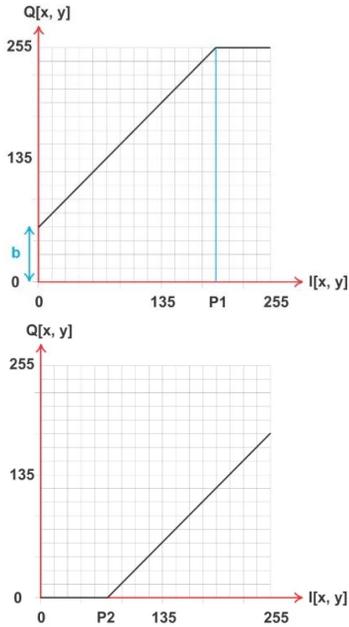
Gambar 8. Citra berskala abu dan pikselnya.

Berdasarkan pembahasan di atas, kita dapat merepresentasikan citra grayscale sebagai fungsi dua dimensi $I [x,y]$ dimana argumen x dan y adalah koordinat bidang yang menentukan piksel tertentu dari citra. Nilai fungsi tersebut menentukan intensitas atau tingkat keabuan citra pada titik tersebut. Hal ini juga berlaku pada citra lainnya seperti citra biner dan RGB.

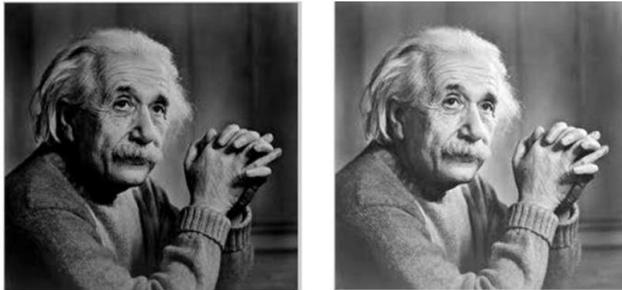
2. Meningkatkan Kecerahan

Operasi peningkatan kecerahan adalah operasi dasar dengan tujuan untuk membuat gambar menjadi lebih terang dengan menaikkan tingkat kecerahannya (*brightness*). Secara matematis, peningkatan kecerahan dilakukan dengan cara mengubah semua nilai piksel citra dengan konstanta. Yaitu dengan menambahkan suatu konstanta β terhadap nilai seluruh piksel. Misalnya, $f(y,x)$ menyatakan nilai piksel pada citra berskala keabuan pada koordinat (y,x) . Maka, citra baru telah meningkat nilai kecerahan semua pikselnya sebesar β terhadap citra asli $f(y,x)$. Demikian pula, kita dapat mengurangi konstanta β dari semua nilai piksel untuk membuat gambar menjadi lebih gelap. Ilustrasi pada Gambar 10 memperlihatkan bagaimana penambahan nilai konstanta pada piksel (kiri) dan sebaliknya bila dilakukan pengurangan (kanan).

Gambar 10 menunjukkan perbedaan kecerahan citra yang sudah ditingkatkan. Kita dapat dengan mudah menyatakan bahwa gambar kanan lebih terang daripada gambar kiri. Namun, jika gambar di sebelah kanan dibuat lebih gelap gambar kiri, maka kita dapat mengatakan bahwa gambar di sebelah kiri lebih terang daripada yang kiri.



Gambar 9. Penambahan dan pengurangan konstanta pada nilai piksel citra.

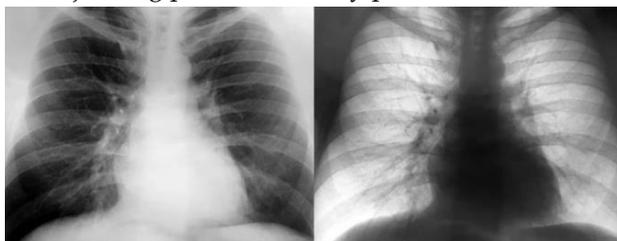


Gambar 10. Perbandingan citra kecerahan yang ditingkatkan.

3. Membalik Citra

Biasanya pembalikan citra dilakukan pada hasil foto rekam medis berupa fotografi rontgen. Citra tersebut disebut dengan citra negatif. Tujuan dilakukannya hal ini adalah untuk mempermudah dan memperjelas objek inti yang ingin

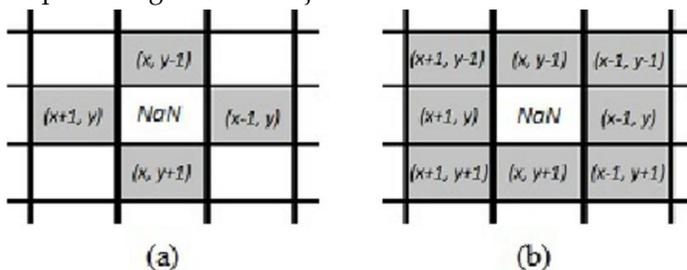
dilihat pada sebuah citra. Contohnya adalah untuk melihat gambar jantung pada citra X-Ray pada Gambar 11.



Gambar 11. Perbandingan citra positif (kiri) dan negatif (kanan) dari foto X-Ray.

B. Operasi Ketetangaan Piksel

Operasi ketetangaan piksel adalah operasi pengolahan citra untuk mendapatkan nilai suatu piksel yang melibatkan nilai piksel-piksel di sekitarnya. Hal ini didasarkan pada kenyataan bahwa setiap piksel pada umumnya tidak berdiri sendiri, melainkan terkait dengan piksel tetangga. karena merupakan bagian suatu objek tertentu di dalam citra.



Gambar 12. Ketetangaan piksel.

Ketetangaan piksel yang umum digunakan adalah 4-ketetangaan dan 8-Ketetangaan. Pada 4-ketetangaan (a) yang merupakan tetangga terdekat piksel P yaitu $\{(x, y - 1), (x + 1, y), (x - 1, y), (x, y + 1)\}$. Sedangkan pada 8-ketetangaan (b), tetangga piksel P yaitu piksel-piksel yang berada disekitar P dengan totalnya sebanyak 8 buah.

1. Aplikasi Ketetangaan pada Citra

Filter yang menggunakan ketetangaan piksel yaitu filter batas, filter pererataan, dan filter median. Sebagai filter atau tapis, operasi ketetangaan piksel berfungsi untuk

menyaring atau paling tidak mengurangi gangguan atau penyimpangan pada citra.

2. Pengertian Konvolusi

Menurut (Kadir & Susanto, 2012) Konvolusi seringkali dilibatkan dalam operasi ketetanggaan piksel. Konvolusi pada citra sering disebut sebagai konvolusi dua-dimensi (2D). Yang mana didefinisikan sebagai proses untuk memperoleh suatu piksel didasarkan pada nilai piksel itu sendiri dan tetangganya, dengan melibatkan suatu matriks yang disebut kernel yang merepresentasikan pembobotan. Wujud kernel umumnya bujur sangkar, tetapi dapat pula berbentuk persegi panjang.

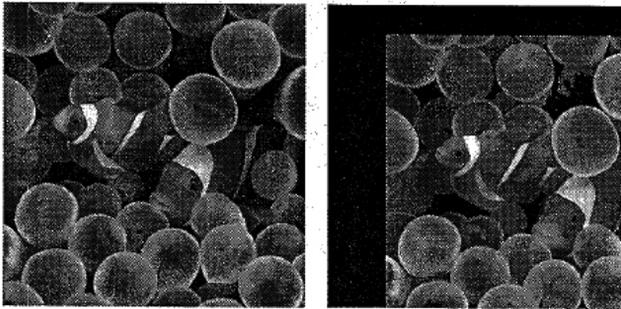
Konvolusi adalah operator matematika yang penting untuk banyak operator dalam image processing. Konvolusi menyediakan cara untuk menggabungkan dua array, biasanya untuk ukuran array yang berbeda, tetapi untuk dimensi array yang sama, menghasilkan array ketiga yang mempunyai dimensi yang sama. Konvolusi dapat digunakan dalam image processing untuk menerapkan operator yang mempunyai nilai output dari piksel yang berasal dari kombinasi linear nilai input piksel tertentu. Konvolusi citra adalah tehnik untuk menghaluskan suatu citra atau memperjelas citra dengan menggantikan nilai piksel dengan sejumlah nilai piksel yang sesuai atau berdekatan dengan piksel aslinya. Tetapi dengan adanya konvolusi, ukuran dari citra tetap sama, tidak berubah.

C. Operasi Geometrik

Operasi geometrik adalah operasi pada citra yang dilakukan secara geometris seperti translasi, rotasi, dan mengubah skala hingga memberikan filter. Pada operasi seperti ini terdapat pemetaan geometrik, yang menyatakan hubungan pemetaan antara piksel pada citra masukan dan piksel citra keluaran.

1. Menggeser Citra

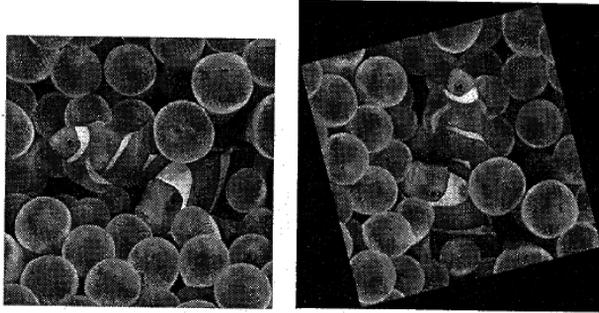
Penggeseran citra mudah disebut juga translasi citra adalah menggeser citra ke arah mendatar atau vertikal. Dengan cara memindahkan setiap elemen piksel citra *input* ke posisi baru pada citra *output* di mana dimensi dari kedua citra pada umumnya adalah sama. Posisi baru dari suatu piksel ditentukan dari nilai variabel translasi (p, q) . Secara umum operasi translasi melakukan perubahan dengan cara menambahkan koordinat awal dengan nilai variabel translasi. Jika ukuran citra *output* diset sama dengan citra *input* maka bila terdapat posisi hasil yang berada di luar batas citra *output*, piksel tersebut tidak dipetakan. Untuk posisi citra *output* yang tidak memiliki nilai pixel diset dengan nilai 0 atau warna hitam. Pada Gambar 13 berikut adalah gambar *input* di sebelah kiri dan gambar *output* di sebelah kanan yang mengalami translasi citra.



Gambar 13. Contoh pergeseran citra

2. Memutar Citra

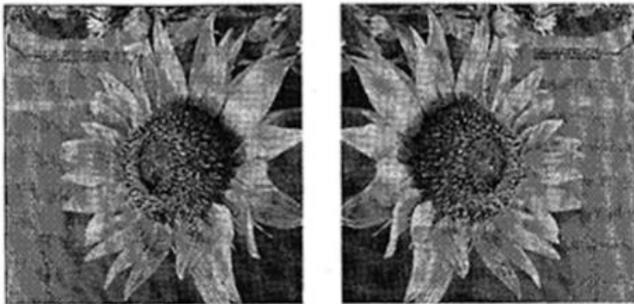
Rotasi merupakan suatu transformasi geometri memindahkan nilai-nilai piksel dari posisi awal menuju ke posisi akhir yang ditentukan melalui nilai variabel rotasi sebesar θ^0 terhadap sudut 0^0 atau garis horizontal dari citra. Sumbu putar θ umumnya memiliki arah putar searah jarum jam dengan garis horizontal. Seperti halnya operasi translasi, hasil perhitungan posisi hasil rotasi dapat memberikan nilai di luar batas citra *output* (apabila ukuran citra *output* sama dengan citra *input*).



Gambar 14. Contoh perputaran citra.

3. Pencerminan Citra

Salah satu teknik dalam pengolahan citra geometri adalah citra cermin. Cermin bayangan adalah pantulan bayangan pada cermin datar dengan ukuran yang sama antara bayangan sumber dan hasil pencerminan. Refleksi atau pencerminan dilakukan dengan memindahkan nilai-nilai piksel pada posisi awal (x_1, y_1) menuju ke posisi baru di (x_2, y_2) pada citra *output* sesuai dengan posisi pencerminan. Posisi pencerminan ada tiga jenis yaitu pencerminan terhadap sumbu x , pencerminan terhadap sumbu y , dan pencerminan terhadap sumbu x dan y .



Gambar 15. Contoh pencerminan citra.

4. Penskalaan Citra

Penskalaan dilakukan dengan memberikan efek memperbesar atau memperkecil ukuran citra *input* sesuai dengan variabel penskalaan citranya. Ukuran baru hasil

penskalaan didapat melalui perkalian antara ukuran citra *input* dengan variabel penskalaan. Penskalaan citra terdiri dari pembesaran dan pengecilan citra, baik itu secara vertical, horizontal maupun keseluruhan.

Seperti yang sudah dijelaskan sebelumnya tipe file citra terbagi mejadi citra raster dan vector. Gambar raster adalah gambar *bitmap* yang terdiri dari piksel individu. Contohnya adalah tipe file JPEG dan PNG. Proses penskalaan gambar raster juga disebut *resampling*, di mana piksel dipetakan ke kisi baru, yang mungkin lebih kecil atau lebih besar dari matriks aslinya. Memperkecil grafik raster dapat menyebabkan sedikit penurunan kualitas gambar karena piksel dipadatkan menjadi kisi yang lebih kecil. Begitu pula meningkatkan grafik raster juga menurunkan kualitas gambar.

Umumnya, memperbesar ukuran gambar digital akan membuatnya terlihat buram. Semakin besar skala gambar, semakin kabur tampilannya. Sedangkan gambar vektor, seperti tipe file SVG atau AI, terdiri dari titik dan garis, dapat ditingkatkan atau diturunkan ukurannya tanpa kehilangan kualitas. Karena piksel tidak dipetakan ke kisi lain, penskalaan gambar vektor hanya memindahkan titik-titik di dalam gambar. Oleh sebab itu gambar vector tampak tajam baik dalam ukuran kecil maupun besar. Untuk alasan ini, logo perusahaan dan ikon aplikasi sering dirancang sebagai grafik vektor. Saat dipublikasikan dalam aplikasi perangkat lunak atau situs *web*, gambar dapat diskalakan ke ukuran tertentu dan kemudian disimpan sebagai grafik raster.



Gambar 16. Contoh penskalaan citra (pembesaran).

5. Efek *ripple* dan *twirl*

Efek gelombang atau biasa disebut dengan *ripple* (riak) adalah aplikasi transformasi citra yang membuat gambar terlihat bergelombang. Efek riak dapat dibuat baik pada arah x maupun y . Di mana φ adalah garis dasar rotasi dan parameter lainnya didefinisikan sebagai transformasi kutub. Efek dari transformasi adalah pusat tetap pada posisi yang sama dan sisa piksel diputar di sekitar pusat dengan sudut rotasi maksimum (φ derajat) di dekat pusat dan menjadi lebih kecil semakin dekat dengan sudut gambar piksel adalah.



Gambar 17. Contoh efek *ripple* pada citra.

Sedangkan efek *twirl* adalah efek olak atau puntiran yang dilakukan dengan memutar citra berdasarkan titik pusat citra, tetapi tidak bersifat linear. Memutar gambar dengan sudut α di tengah atau titik jangkar (x_c, y_c) . Semakin bertambah putaran gambar seiring jarak radial r dari pusat

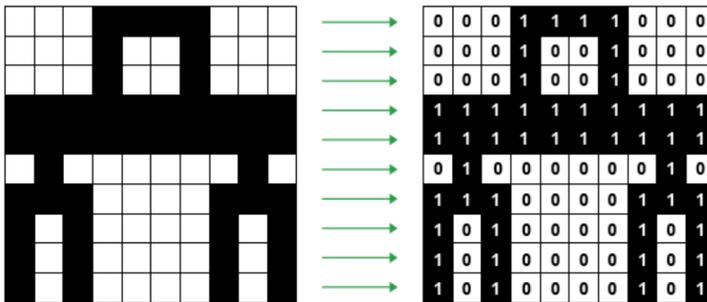
meningkat (hingga r_{maks}), gambar tidak berubah di luar jarak radial r_{maks} .



Gambar 18. Contoh efek *twirl* pada citra.

D. Operasi pada Citra Biner

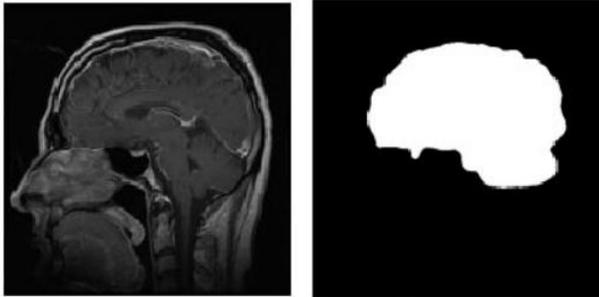
Template citra biner adalah citra yang memiliki hanya dua nilai *graylevel*, 0 dan 1. Dimana hitam = 1, putih = 0, maupun kebalikannya.



Gambar 24. Ilustrasi citra biner.

Adapun pentingnya citra biner yaitu kebutuhan memori untuk setiap piksel sedikit yaitu 1 bit/piksel. Hal ini sangat menguntungkan dalam hal penyimpanan. Selain itu citra biner biasa digunakan dalam merepresentasikan citra hasil pendeteksian tepi (*edge detection*) dan juga dapat menggunakan operasi logika (AND, OR, NOT) sehingga waktu komputasinya kecil. Untuk memisahkan (*segmentasi*) objek dari gambar latar belakangnya. Sedangkan untuk lebih memfokuskan pada

analisis bentuk morfologi, yang dalam hal ini intensitas piksel tidak terlalu penting dibandingkan bentuknya. Setelah objek dipisahkan dari latar belakangnya, properti geometri dan morfologi objek dapat dihitung dari citra biner.



Gambar 25. Segmentasi objek dari *background*.

1. Konversi citra *grayscale* ke citra biner.
 - a. Konversi dari citra hitam-putih ke citra biner dilakukan dengan operasi pengambangan (*thresholding*).
 - b. Operasi pengambangan mengelompokkan nilai derajat keabuan setiap pixel ke dalam 2 kelas, hitam dan putih. Operasi pengambangan terbagi menjadi 3, yaitu pengambangan secara global, lokal, dan adaptif.

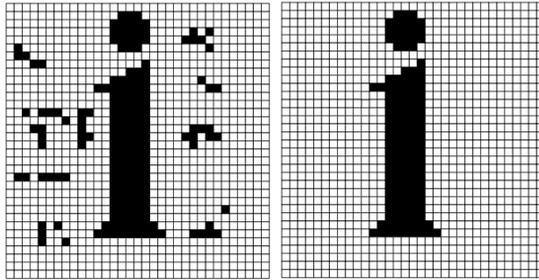
2. Kegunaan citra biner hasil pengambangan.

Citra biner hasil pengambangan berguna untuk memisahkan objek dengan latar belakang pada citra asalnya. Sehingga citra biner menjadi template untuk melakukan segmentasi.

3. Penapis luas.

Seringkali citra biner hasil pengambangan mengandung beberapa daerah yang dianggap sebagai gangguan. Biasanya daerah gangguan itu berukuran kecil. Penapis luas dapat digunakan untuk menghilangkan daerah gangguan tersebut. Misalkan objek yang dianalisis diketahui mempunyai luas yang lebih besar dari T . Maka piksel-piksel dari daerah yang luasnya di bawah T dinyatakan dengan 0.

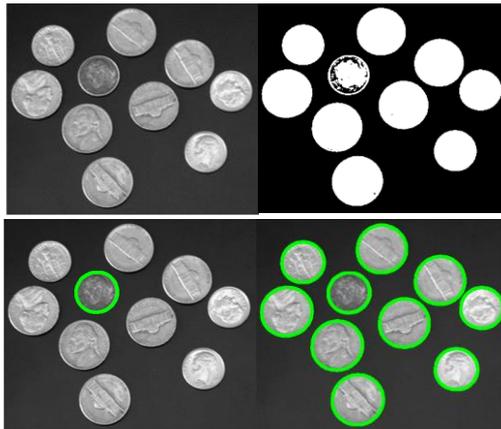
Dengan cara ini, daerah yang berupa gangguan dapat dihilangkan.



Gambar 26. Penapisan pada gangguan citra biner.

4. Penelusuran Batas Objek (*Boundary tracing*).

Setelah citra grayscale disegmentasi menjadi citra biner, maka perbedaan antara objek dengan latar belakang terlihat dengan jelas. Pixels objek berwarna putih sedangkan pixels latar belakang berwarna hitam. Rangkaian pixels yang menjadi batas (*boundary*) antara objek dengan latar belakang dapat diketahui secara keseluruhan (algoritma *boundary following*).



Gambar 27. Plot batas objek citra koin dengan *boundary tracing*.

BAB 5

Morfologi Pengolahan Citra

Operasi morfologi citra merupakan suatu proses yang bertujuan untuk mengubah bentuk objek pada citra asli. Proses tersebut dapat dilakukan pada citra *grayscale* maupun citra biner. Jenis-jenis operasi morfologi di antaranya adalah dilasi, erosi, *closing*, *opening* maupun *Hit-or-Miss transform*. Operasi ini biasa digunakan untuk mengubah struktur bentuk objek yang terkandung dalam citra.

A. Operasi Dilasi

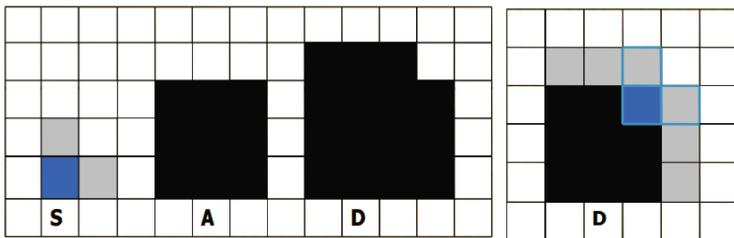
Dilasi merupakan proses penggabungan titik-titik latar (0) menjadi bagian dari objek (1), berdasarkan structuring element S yang digunakan. Rumus dilasi yaitu $D(A, S) = A \oplus S$. Dengan cara untuk setiap titik A letakkan titik poros S pada titik A tersebut kemudian beri angka 1 untuk semua titik (x, y) yang terkena/tertimpa oleh struktur S pada posisi tersebut.

Contoh dilasi =

$$S = \{(0,0), (0,1), (1,0)\}$$

$$= \{\text{poros}, (+0, +1), (+1, +0)\}$$

$$A = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}$$



Gambar 19. Proses dilasi pada saat posisi poros S ada di $(2,2)$.

Tabel 1. Proses dilasi.

Posisi poros $(x(x, y) \in A)$	S_{xy}
(0,0)	$\{(0,0), (1,0), (0,1)\}$
(0,1)	$\{(0,1), (1,1), (0,2)\}$
(0,2)	$\{(0,2), (1,2), (0,3)\}$
...	...
(2,2)	$\{(2,2), (2,3), (3,2)\}$

B. Operasi Erosi

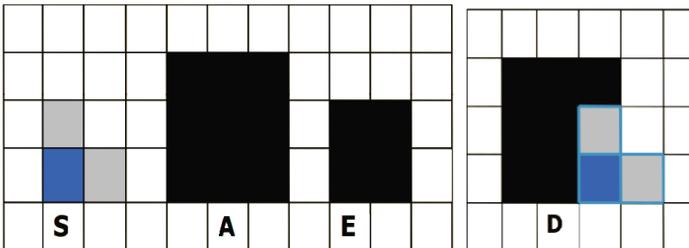
Operasi erosi adalah kebalikan dari operasi dilasi. Erosi merupakan proses penghapusan titik-titik objek (1) menjadi bagian dari latar (0), berdasarkan structuring element S yang digunakan. Rumus erosi yaitu $E(A, S) = A \otimes S$. Dengan cara untuk setiap titik pada A letakkan titik poros S pada titik A tersebut. Jika ada bagian dari S yang berada di luar A , maka titik poros dihapus / dijadikan latar.

Contoh erosi =

$$S = \{(0,0), (0,1), (1,0)\}$$

$$= \{\text{poros}, (+0, +1), (+1, +0)\}$$

$$A = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}$$



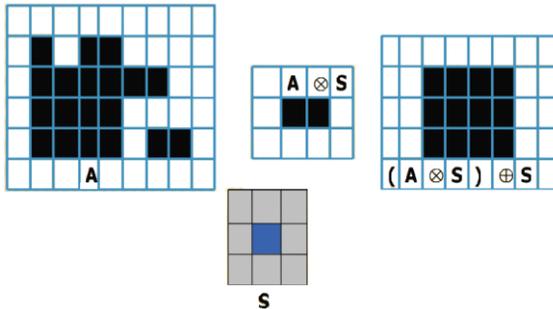
Gambar 20. Proses erosi pada saat posisi poros S ada di (2,2).

Tabel 2. Proses erosi.

Posisi poros ($x(x, y) \in A$)	S_{xy}	Kode
(0,0)	{(0,0), (1,0), (0,1)}	1
(0,1)	{(0,1), (1,1), (0,2)}	1
(0,2)	{(0,2), (1,2), (0,3)}	0
....	
(2,2)	{(2,2), (2,3), (3,2)}	0

C. Proses *Opening*

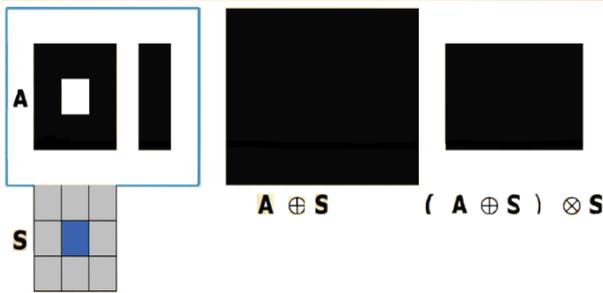
Opening adalah proses erosi yang diikuti dengan dilasi. Efek yang dihasilkan adalah menghilangkan objek-objek kecil dan kurus, memecah objek pada titik-titik yang kurus, dan secara umum membuat *smooth* batas dari objek besar tanpa mengubah area objek secara signifikan. Adapun rumusnya yaitu $A \circ S = (A \otimes S) \oplus S$.



Gambar 21. Proses *opening*.

D. Proses *Closing*

Closing adalah proses dilasi yang diikuti dengan erosi. Efek yang dihasilkan adalah mengisi lubang kecil pada objek, menggabungkan objek-objek yang berdekatan, dan secara umum menjadikan *smooth* batas dari objek besar tanpa mengubah area objek secara signifikan. Adapun rumusnya yaitu $A \bullet S = (A \oplus S) \otimes S$.



Gambar 22. Proses closing.

E. Hit-or-Miss Transform

Suatu structuring element S dapat direpresentasikan dalam bentuk (S_1, S_2) dimana S_1 adalah kumpulan titik-titik objek (hitam) dan S_2 adalah kumpulan titik-titik latar (putih). Dengan rumus $A * S = (A \otimes S_1) \cap (A \otimes S_2)$.

Hit-and-Miss transform $A * S$ adalah kumpulan titik-titik dimana S_1 menemukan match di A dan pada saat yang bersamaan S_2 juga menemukan match di luar A .

Contoh :

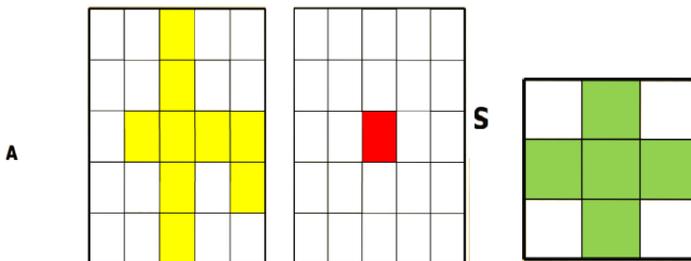
$$S_1 = \{b, e, h\}$$

$$S_2 = \{a, d, g, c, f, i\}$$

a	b	c
d	e	f
g	h	i

$A * S$

→ untuk yang *match* dipertahankan sedangkan yang tidak *match* dihapus



Gambar 23. Proses Hit-or-Miss Transform.

BAB 6

Pengolahan Citra Berwarna

Pengolahan citra berwarna adalah analisis, transformasi, dan interpretasi data visual yang disajikan dalam warna. Ini dapat menghasilkan berbagai hasil dari konversi skala abu-abu dari gambar hitam putih ke analisis rinci informasi yang terkandung dalam foto yang diambil oleh berbagai perangkat digital. Pada bab ini akan diuraikan berbagai ruang warna yang biasa digunakan dalam pengolahan citra digital berwarna.

A. Dasar dan Ruang Warna

Warna adalah informasi visual penting bagi manusia. Cahaya (informasi berwarna) yang dipantulkan dari suatu objek diserap oleh sel kerucut dari sistem visual kita dan akhirnya mengarah pada persepsi warna. Penafsiran penyerapan kerucut ini oleh sistem saraf kita adalah dasar dari persepsi warna kita. Jadi warna adalah representasi persepsi dari reflektansi permukaan suatu objek.

Selain bentuk, tekstur, dan fitur gambar tingkat rendah lainnya, informasi warna merupakan fitur penting yang telah berhasil digunakan di banyak aplikasi pemrosesan gambar seperti pengenalan objek, pencocokan gambar, pengambilan gambar berbasis konten, visi komputer, kompresi gambar berwarna, dan lain sebagainya.

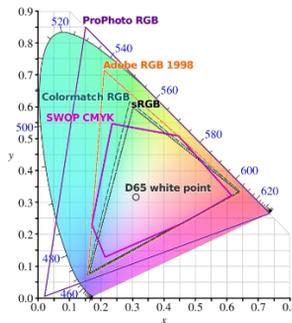
Organisasi warna dalam gambar dalam format tertentu disebut ruang warna. Cara warna direpresentasikan disebut model warna. Setiap gambar menggunakan salah satu dari berbagai macam ruang warna untuk representasi gambar yang efektif. Adapun beberapa ruang warna contohnya adalah RGB, XYZ, HSV/HSL, LAB, LCH, YPbPr, YUV, YIQ, dan lain sebagainya.

Sejumlah ruang warna atau model warna telah diusulkan dan masing-masing memiliki sistem koordinat warna tertentu dan setiap titik dalam ruang warna hanya mewakili satu warna tertentu. Gambar berwarna yang dihasilkan oleh sistem

pencitraan digital, direpresentasikan sebagai merah, hijau, biru dan biasanya disebut gambar RGB. Mereka berguna untuk monitor warna, dan kamera video.

B. Ruang Warna RGB.

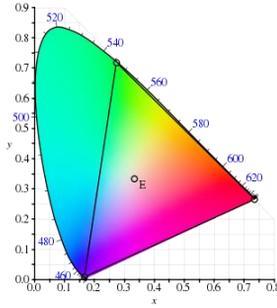
Menggunakan ruang warna RGB, merah, hijau, dan biru dicampur dengan cara yang berbeda untuk membuat kombinasi warna yang berbeda. Alasan penggunaan warna RGB adalah karena mata kita memiliki reseptor warna yang dapat merasakan ketiga warna ini dan kombinasinya dengan cukup efektif. Kita dapat membentuk warna apa saja, secara teoritis, dari ketiga warna ini. Intensitas setiap warna ditentukan dalam rentang 0 hingga 255. Rentang ini disebut kedalaman warna. Ruang warna RGB memiliki dua komponen yaitu kromatisitas titik putih dan kurva koneksi gamma.



Gambar 28. Ruang warna RGB.

C. Ruang Warna XYZ

Warna RGB memiliki ambang saturasi. Mereka tidak bisa melampaui apa yang bisa kita lihat. Ruang warna XYZ membantu kita melampaui ambang batas ini. Warna XYZ dapat digunakan untuk pencocokan warna. Kita dapat memasukkan kode warna dan kemudian mereproduksinya nanti dalam aplikasi yang berbeda, seperti pencetakan. Dengan menggunakan XYZ, kita dapat mengkodekan semua warna yang ada di dunia nyata. Ruang warna ini disebut XYZ karena mengekstrapolasi warna RGB dalam tiga dimensi: x , y , dan z .



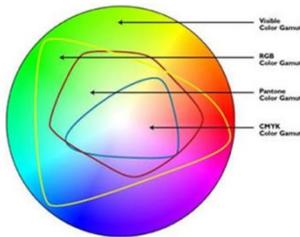
Gambar 29. Ruang warna XYZ

D. Ruang Warna CMY/CMYK

Pada CMY, warna hitam diperoleh jika C, M, dan Y bernilai sama. Namun model warna lain yang menggunakan CMYK (cyan, magenta, yellow, dan black) ditambahkan warna hitam tersendiri sehingga berguna dalam *printer* warna. Alasan ditambahkan warna hitam adalah untuk efisiensi tinta tanpa perlu pencampuran warna lainnya sehingga dapat menghemat biaya. Sebagian besar perangkat *output* termasuk printer warna atau mesin fotokopi menggunakan model warna CMY.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Sama seperti warna aditif primer merah, hijau dan biru, warna primer pigmen di sisi lain adalah magenta, cyan dan kuning dan warna sekunder yang sesuai adalah merah, hijau dan biru. Pigmen primer yang sama (misalnya cyan, magenta, dan kuning) juga dapat menghasilkan hitam. Jadi sistem empat warna cyan (C), magenta (M), kuning (Y), dan hitam (B) membentuk model empat warna.



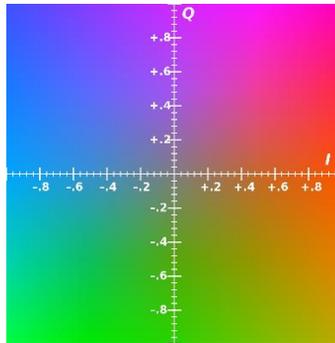
Gambar 30. Ruang warna CMYK

E. Ruang Warna YIQ

Dalam ruang warna ini YIQ juga dikenal dengan nama ruang warna NTSC, informasi luminansi Y mewakili informasi skala abu-abu, sedangkan *hue* (I) dan saturasi (Q) membawa informasi warna. Konversi dari RGB ke YIQ adalah elemen baris pertama jika dijumlahkan menjadi satu dan elemen baris kedua dan ketiga berjumlah 0.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 299 & 587 & 114 \\ 596 & 274 & 322 \\ 211 & 523 & 312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

Jadi pada gambar skala abu-abu, di mana $R = G = B$, komponen warna I dan Q adalah nol. Ruang warna NTSC digunakan di televisi.



Gambar 31. Ruang warna YIQ.

F. Ruang Warna YC_bC_r dan YP_bPr

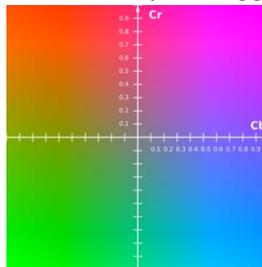
Dalam ruang warna ini, Y adalah komponen bercahaya sedangkan C_b dan C_r memberikan informasi warna. Ruang warna ini digunakan dalam video digital. Informasi dari RGB ke YC_bC_r adalah sebagai berikut:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.00 \\ 112.00 & -93.788 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3)$$

Selain itu ruang warna YP_bPr digunakan dalam elektronik video, seperti pemutar DVD. Yang mana terdiri dari tiga komponen berikut: Y = kabel hijau, P_b = kabel biru, dan P_r

= kabel merah. Ketiga komponen tersebut diturunkan dari ruang warna RGB saja. Y mengacu pada kecerahan; Pb dan Pr adalah dua sinyal warna yang berbeda. Secara umum, saat menggunakan komputer, komponen warna digital diturunkan dari ruang warna RGB. Namun, ketika kita berbicara tentang perangkat elektronik (seperti pemutar DVD), kita perlu menggunakan rekan analog dari ruang warna RGB, yaitu YPbPr.

YCbCr biasa diterapkan dalam kompresi MPEG yang digunakan dalam DVD, TV digital, dan Video CD. Sedangkan YPbPr adalah analog yang menggunakan tiga kabel untuk koneksi, sedangkan YCbCr hanya menggunakan satu kabel.

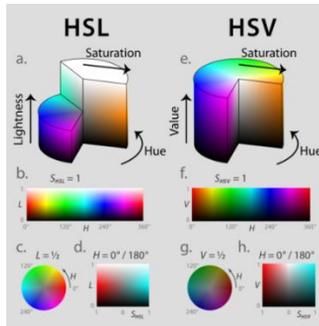


Gambar 32. Ruang warna YCbCr.

G. Ruang Warna HIS, HSV, dan HSL

HIS/HSV/HSL adalah representasi alternatif dari ruang warna RGB. Memiliki komponen hue (H), saturasi (S), intensitas (I), *value* (V), dan *lightness* (L). *Hue* adalah properti yang menggambarkan tiga warna: hijau, merah, dan magenta. Itu juga bisa menjadi campuran dua warna murni: merah dan kuning, dan kuning dan hijau. Saturasi mengukur intensitas gambar. Ini memberitahu kita seberapa jauh warna dari abu-abu. Nilai yang lebih rendah berarti warnanya mendekati abu-abu. *Lightness* mengacu pada intensitas warna sehubungan dengan putih. Ini memberitahu kita seberapa jauh suatu warna dari warna putih. Transformasi RGB ke ruang warna HSC yaitu sebagai berikut:

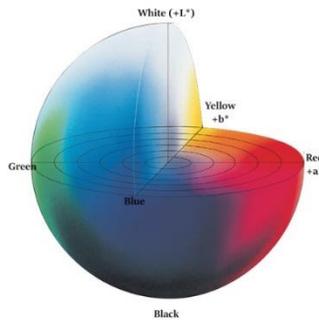
$$H = \tan \left[\frac{3(G - B)}{(R - G) + (R - B)} \right], S = 1 - \frac{\min(R, G, B)}{V}, V = \frac{R + G + B}{3} \quad (4)$$



Gambar 33. Perbandingan ruang warna HSL dan HSV.

H. Ruang Warna CIELAB

CIELAB yang diadopsi sebagai standar internasional pada tahun 1970-an. Properti ruang warna CIELAB ini membuatnya sangat menarik dan berguna untuk analisis warna, dan keunggulan ruang warna CIELAB di atas ruang warna lain telah dibuktikan dalam banyak aplikasi gambar berwarna. Misalnya ruang warna CIELAB telah berhasil digunakan untuk pengelompokan warna.



Gambar 34. Ruang warna LAB.

BAB 7

Segmentasi Citra

Segmentasi citra adalah metode membagi citra digital menjadi subkelompok yang disebut segmen citra, mengurangi kompleksitas citra dan memungkinkan pemrosesan lebih lanjut atau analisis setiap segmen citra. Secara teknis, segmentasi adalah pemberian label pada piksel untuk mengidentifikasi objek, orang, atau elemen penting lainnya dalam citra. Pada bab ini diuraikan teknik-teknik dalam segmentasi citra digital.

A. Pengertian Segmentasi Citra

Segmentasi adalah proses membagi citra menjadi sejumlah daerah homogen yang seragam. Dengan kata lain, segmentasi suatu citra didefinisikan oleh sekumpulan region yang terhubung dan tidak tumpang tindih, sehingga setiap piksel dalam segmen pada citra memperoleh label region unik yang menunjukkan region tersebut. Lebih tepatnya, segmentasi gambar adalah proses pemberian label ke setiap piksel dalam gambar sedemikian rupa sehingga piksel dengan label yang sama memiliki karakteristik tertentu. Tujuan dari segmentasi adalah untuk menyederhanakan dan/atau mengubah representasi suatu citra menjadi sesuatu yang lebih bermakna dan lebih mudah untuk dianalisis.

Segmentasi adalah salah satu elemen terpenting dalam analisis citra otomatis, terutama karena pada langkah ini objek atau entitas lain yang diinginkan diekstraksi dari citra untuk pemrosesan selanjutnya, seperti deskripsi dan pengenalan. Misalnya, citra satelit yang menunjukkan wilayah darat dan laut maka objek-objek di bagian darat dari perlu disegmentasi dengan tepat agar selanjutnya bisa diklasifikasikan.

Setelah mengekstrak setiap segmen, tugas selanjutnya adalah mengekstrak sekumpulan fitur penting seperti tekstur, warna, dan bentuk. Ini adalah entitas terukur yang memberikan ukuran berbagai properti segmen gambar. Beberapa properti tekstur adalah kehalusan, kasar, keteraturan, dll. Sedangkan

deskriptor bentuk umum adalah panjang, lebar, rasio aspek, area, lokasi, keliling, kekompakan, dll. Setiap wilayah tersegmentasi dalam sebuah adegan dapat dicirikan oleh satu set fitur tersebut.



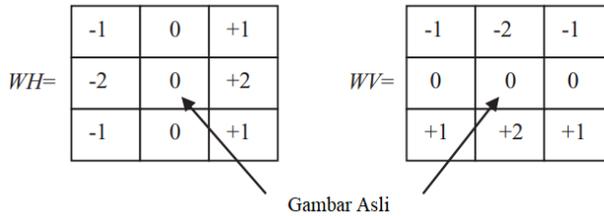
Gambar 35. Segmentasi objek dari *background*.

Akhirnya berdasarkan set. fitur yang diekstraksi ini, setiap objek tersegmentasi diklasifikasikan ke salah satu dari satu set kelas yang bermakna. Dalam citra digital lautan, kelas-kelas ini mungkin kapal atau perahu kecil atau bahkan kapal angkatan laut dan kelas besar, badan air. Masalah segmentasi adegan dan klasifikasi objek adalah dua bidang studi yang terintegrasi dalam visi mesin. Sistem pakar, jaringan semantik, dan sistem berbasis jaringan saraf telah ditemukan untuk melakukan tugas penglihatan tingkat tinggi dengan cukup efisien.

B. Deteksi Tepi

Deteksi tepi mengandung banyak informasi tentang berbagai wilayah dalam suatu gambar. Fitur-fitur ini biasanya disebut sebagai fitur lokal, karena mereka diekstraksi dari properti lokal saja. Tepi pada dasarnya membatasi antara dua wilayah yang berbeda. Deteksi tepi berfungsi untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Umumnya, deteksi tepi menggunakan dua macam detektor, yaitu detektor baris (H_y) dan detektor kolom (H_x). Beberapa contoh yang tergolong jenis ini adalah operator Roberts, Prewitt, Sobel, dan Frei-Chen.

Di antara metode deteksi tepi yang dikenal adalah filter gradien sederhana yang berisi semacam penghalusan gambar. Filter Sobel adalah contoh umum. Ini didefinisikan oleh dua matriks dengan bobot seperti yang ditunjukkan pada Gambar 36.



Gambar 36. Berat filter *Sobel*.

Deteksi tepi dapat dibagi menjadi dua golongan. Golongan pertama disebut deteksi tepi orde pertama, yang bekerja dengan menggunakan turunan atau diferensial orde pertama. Golongan kedua dinamakan deteksi tepi orde kedua, yang menggunakan turunan orde kedua. Sejumlah detektor tepi telah dikembangkan oleh berbagai peneliti. Diantaranya adalah Operator *Robert*, Operator *Sobel*, Operator *Prewitt*, Operator *Kirsch*, Operator tepi *Canny*. Operator berbasis turunan pertama, terdiri dari Operator *Laplacian*, Operator *Laplacian of Gaussian* (LoG), dan Operator *Difference of Gaussian* (DoG).

C. Deteksi Garis

Deteksi garis pada citra dapat diperoleh melalui penggunaan “cadar” (*mask*). Seperti yang terlihat pada Gambar 37, cadar (a) berguna untuk memperoleh garis horizontal, cadar (b) untuk mendapatkan garis yang berorientasi 45°, cadar (c) untuk memperoleh garis tegak, dan cadar (d) untuk mendapatkan garis yang berorientasi -45°.

$$\begin{matrix}
 \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} &
 \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} &
 \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} &
 \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \\
 \text{(a)} & \text{(b)} & \text{(c)} & \text{(d)}
 \end{matrix}$$

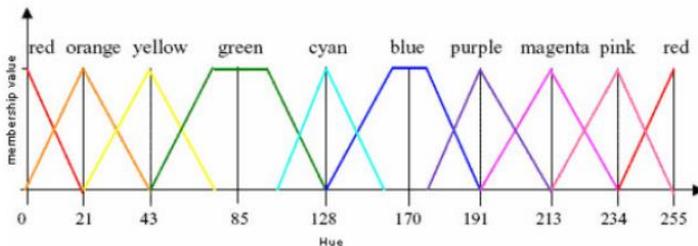
Gambar 37. Empat pasang cadar untuk mendeteksi keberadaan garis

D. *Thresholding*

Dalam pengolahan citra digital, *thresholding* adalah metode paling sederhana dari segmentasi citra. Dari citra *grayscale*, *thresholding* dapat digunakan untuk membuat citra biner. Operasi *thresholding* melibatkan identifikasi satu set ambang batas optimal, berdasarkan mana gambar dipartisi menjadi beberapa wilayah yang berarti. Beberapa model yang sering digunakan yaitu *Bi-level Thresholding*, *Multilevel Thresholding*, dan *Entropy-Based Thresholding*.

E. Segmentasi Warna

Segmentasi warna dapat dilakukan pada ruang warna HLS. Kemudian, dengan berpedoman pada susunan warna sesuai dengan gambar di bawah dilakukan perubahan warna *Hue* yang berdekatan dengan warna yang menjadi pusat dalam fungsi keanggotaan *fuzzy*. Sebagai contoh, warna hijau dengan $H = 80$ akan diubah menjadi 85 dan warna kuning dengan $H = 40$ diubah menjadi 45. Adapun nilai pada komponen L dan S disederhanakan menjadi tiga nilai, yaitu 0, 128, dan 255. Warna yang mungkin timbul ada dua belas buah, yaitu merah, jingga, kuning, hijau, cyan, biru, ungu, magenta, merah muda, hitam, putih, dan abu-abu. Selanjutnya, warna HLS diubah kembali ke RGB.



Gambar 38. Dimensi H.

BAB 8

Restorasi dan Peningkatan Citra

Gambar-gambar yang kita peroleh dari beragam perangkat digital saat diubah dari satu bentuk ke bentuk lain dengan proses seperti pencitraan, pemindaian, atau transmisi maka kualitas gambar keluaran mungkin lebih rendah dari gambar masukan asli. Oleh karena itu, dibutuhkan proses peningkatan kualitas, sehingga gambar keluaran secara visual lebih bermanfaat bagi pengamat manusia dari sudut pandang subjektif. Pemulihan citra atau biasa disebut *enhancement* juga memiliki tujuan lain, yaitu untuk menghilangkan efek degradasi yang mungkin disebabkan oleh sistem pencitraan atau saluran. Meningkatnya kebutuhan untuk mengembangkan sistem otomatis untuk mendeteksi gambar mengharuskan kualitas gambar yang akan diinterpretasikan harus bebas dari noise dan penyimpangan lainnya. Maka dari itu penting untuk melakukan operasi *preprocessing* pada citra.

A. Restorasi Citra

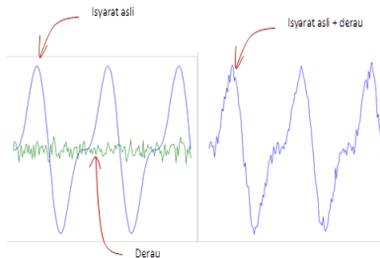
Istilah restorasi mempunyai perbedaan makna dengan peningkatan citra. Peningkatan citra (*image enhancement*) merupakan istilah yang menyatakan usaha untuk membuat gambar agar lebih baik dari sudut pandang pengolahnya. Sedangkan, restorasi citra merupakan proses untuk membuat citra yang kualitasnya turun akibat adanya tambahan derau agar menjadi mirip dengan keadaan aslinya.

Tidak ada sistem pembentukan citra yang sempurna karena keterbatasan fisik yang melekat. Oleh karena itu, gambar tidak identik dengan aslinya bukanlah suatu hal yang asing karena manusia juga membuat kesalahan dalam mengoperasikan sistem pencitraan. Kadang-kadang kita menerima gambar-gambar kabur yang terdegradasi oleh beberapa mekanisme pengambilannya. Salah satu sumber utama degradasi gambar adalah lensa optik dalam kamera digital yang memperoleh informasi visual. Jika kamera tidak fokus dengan tepat maka kita mendapatkan gambar buram.

Faktor lainnya yang dapat menyebabkan degradasi citra yaitu adanya gerakan relatif antara objek dan kamera. Hal ini disebabkan karena kamera mengalami perpindahan impulsif selama interval pengambilan gambar sementara objek statis, Dalam beberapa kasus di atas, diperlukan teknik penyempurnaan gambar yang tepat sehingga gambar yang dihasilkan memiliki kualitas visual yang lebih baik.

B. Noise pada Citra

Noise pada citra disebut juga sebagai derau. Derau merupakan komponen dalam citra yang tidak dikehendaki. Sebagai contoh derau *Gaussian* biasa muncul pada sebarang isyarat. Derau putih (*white noise*) biasa menyertai pada siaran televisi yang berasal dari stasiun pemancar yang lemah. Derau butiran biasa muncul dalam film-film fotografi. Derau yang dinamakan '*salt and pepper*' sering mewarnai citra. Derau *salt* berwarna putih dan derau *pepper* berwarna hitam.



Gambar 39. Derau pada isyarat satu dimensi.

Derau dikelompokkan menjadi 4 yaitu:

1. Derau tambahan (*additive noise*)

Dapat dinyatakan secara matematis seperti berikut:

$$g(.) = f(.) + d(.) \quad (5)$$

Dalam hal ini, $g(.)$ menyatakan isyarat yang telah terkena derau, $f(.)$ menyatakan citra asli, dan $d(.)$ menyatakan derau.

2. Derau perkalian (*multiplicative noise*)

Secara matematis, isyarat yang terkena derau perkalian dapat ditulis seperti berikut:

$$g(.) = f(.) * d(.) \quad (6)$$

3. Derau impuls (*impulse noise*)

Sensor ataupun saluran data terkadang memberikan derau berbentuk biner (0 atau 1). Derau seperti itu dimodelkan sebagai berikut:

$$g(.) = (1 - p) * f(.) + p * d(.) \quad (7)$$

Dalam hal ini, $g(.)$ menyatakan isyarat yang telah terkena derau, $f(.)$ menyatakan citra asli, $d(.)$ menyatakan derau, dan p menyatakan parameter biner yang nilainya berupa 0 atau 1. Berdasarkan rumus di atas, isyarat asli akan hilang saat p bernilai 1.

4. Derau kuantisasi (*quantization noise*)

Noise ini muncul ketika variabel acak kontinu diubah menjadi variabel diskrit atau ketika variabel acak diskrit diubah menjadi variabel dengan level lebih sedikit. Pada citra, derau kuantisasi sering terjadi pada proses akuisisi. Gambar mungkin kontinu pada awalnya, tetapi untuk diproses sebelumnya harus diubah menjadi representasi digital.

C. Jenis *Noise* pada Citra

Adapun jenis-jenis derau yang biasa ditemukan pada citraialah sebagai berikut :

1. *Gaussian*
2. *Garam dan Merica (Salt & Pepper Noise)*
3. *Eksponensial*
4. *Gamma*
5. *Rayleigh*
6. *Uniform*
7. *Periodis*

D. Menghilangkan *Noise* pada Citra

Dalam pengolahan citra digital, untuk menghilangkan derau cara yang umum digunakan adalah dengan melakukan *filtering*. Memfilter data citra adalah proses standar yang digunakan di hampir setiap sistem pemrosesan citra. Filter digunakan untuk menghilangkan *noise* dari gambar dengan

mempertahankan detail yang sama. Pilihan *filter* tergantung pada perilaku *filter* dan jenis data. *Noise* adalah perubahan mendadak dalam nilai piksel dalam sebuah gambar. Jadi ketika akan memfilter gambar, langkah pertama yang dilakukan adalah mengganti nilai setiap piksel dengan rata-rata piksel di sekitarnya. Proses ini akan membuat gambar terlihat lebih halus. Untuk melakukan filterisasi pada citra, berikut jenis-jenis filter yang sering digunakan :

1. Filter Lolos-Rendah
2. Filter Rerata Aritmetik
3. Filter Rerata Harmonik
4. Filter Rerata Kontraharmonik
5. Filter Rerata Yp
6. Filter Median
7. Filter Min
8. Filter Titik Tengah

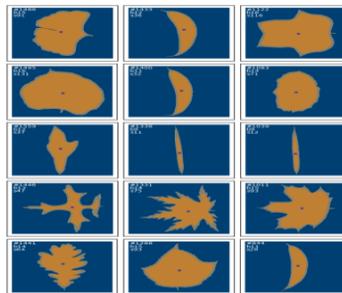
BAB 9

Ekstraksi Fitur Citra

Ekstraksi fitur adalah bagian dari proses pengurangan dimensi, di mana satu set awal data mentah dibagi dan direduksi menjadi grup yang lebih mudah dikelola sehingga ketika ingin mengolahnya akan lebih mudah. Karakteristik terpenting dari kumpulan data besar ini adalah mereka memiliki sejumlah besar variabel. Variabel-variabel ini membutuhkan banyak sumber daya komputasi untuk diproses. Jadi ekstraksi fitur membantu untuk mendapatkan fitur terbaik dari kumpulan data besar tersebut dengan memilih dan menggabungkan variabel ke dalam fitur, sehingga secara efektif mengurangi jumlah data. Fitur-fitur ini mudah diproses, tetapi masih mampu menggambarkan kumpulan data aktual dengan akurasi dan orisinalitas. Pada bab ini dijelaskan fitur-fitur yang diekstraksi pada citra digital.

A. Fitur Bentuk

Fitur bentuk merupakan fitur dasar dalam visual konten pada citra. Dimana setiap gambar dapat dibedakan berdasarkan bentuk dari objek tersebut. Fitur bentuk merupakan suatu fitur yang diperoleh melalui bentuk objek dan dapat dinyatakan melalui kontur, area, dan transformasi, Dalam industri elektronika kadang kala dibutuhkan kemampuan untuk mendeteksi bentuk-bentuk dari PCB (*Printed Circuit Board*) seperti mendeteksi bentuk-bentuk segi empat atau lingkaran untuk memisahkan bagian-bagian dari PCB.



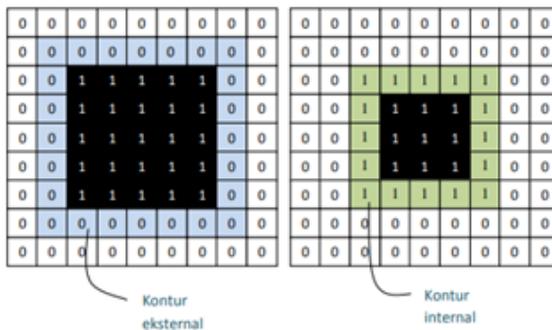
Gambar 40. Ekstraksi fitur berbagai jenis daun.

Bentuk secara kasar dapat didefinisikan sebagai deskripsi suatu objek dikurangi posisi, orientasi, dan ukurannya. Oleh karena itu, fitur bentuk harus invarian terhadap translasi, rotasi, dan skala. Untuk menggunakan bentuk sebagai fitur citra, sangat penting untuk melakukan segmentasi citra untuk mendeteksi batas objek atau wilayah. Adapaun teknik untuk karakterisasi bentuk dapat dibagi menjadi dua kategori. Kategori pertama adalah berbasis batas, menggunakan kontur luar bentuk suatu objek. Kategori kedua adalah berbasis wilayah, menggunakan seluruh bentuk wilayah objek.

Model yang dapat dibangun melalui pengolahan hasil ekstraksi fitur beragam dan sangat bermanfaat. Contohnya adalah bagaimana kamera dapat membedakan objek mobil dan motor, hingga mendeteksi setiap objek yang ada di atas meja atau di sebuah ruangan.

B. Fitur Kontur

Kontur merupakan suatu metode yang digunakan untuk mendapatkan tepi objek. Dalam pengolahan citra terdapat dua istilah kontur yaitu eksternal dan kontur internal.

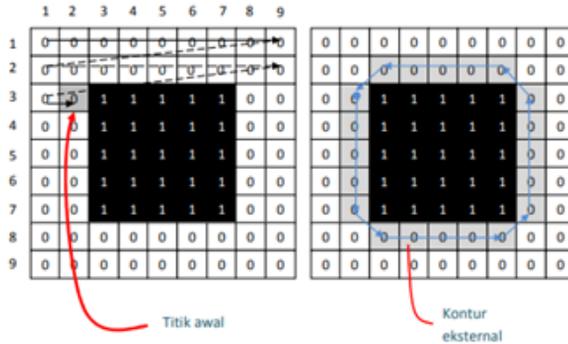


Gambar 41. Kontur eksternal dan kontur internal.

Gambar 41 memberikan ilustrasi tentang perbedaan kedua jenis kontur tersebut. Terlihat bahwa piksel yang menjadi bagian kontur eksternal terletak di luar objek, sedangkan piksel yang menjadi bagian kontur internal terletak di dalam objek itu sendiri.

Untuk memperoleh kontur eksternal, dengan menggunakan pendekatan 8-ketetanggaan, diperoleh hasil

sebagai berikut: (3,2),(4,2),(5,2),(6,2),(7,2),(8,3), (8,4),(8,5), (8,6), (8,7),(7,8), (6,8),(5,8),(4,8),(3,8),(2,7),(2,6),(2,4),(2,3). Proses untuk mendapatkan titik awal (yaitu (3,1)) dilakukan dengan melakukan pemindaian seperti yang diilustrasikan dalam Gambar 40 (kiri). Setelah titik awal ditemukan, penelusuran dilakukan seperti terlihat pada Gambar 40 (kanan).



Gambar 42. Proses penelusuran kontur.

C. Fitur Tekstur

Tekstur adalah fitur gambar yang telah digunakan untuk karakterisasi gambar, dengan aplikasi dalam pengambilan gambar berbasis konten. Karakteristik utama dari tekstur adalah pengulangan pola atau pola di atas suatu wilayah dalam suatu gambar. Tekstur adalah sifat bawaan dari hampir semua permukaan benda, seperti batu bata, kain, kayu, kertas, karpet, awan, pohon, tanah, kulit, dan lain sebagainya. Yang mana berisi informasi penting mengenai pengaturan struktural yang mendasari permukaan dalam gambar. Ketika area kecil dalam sebuah gambar memiliki variasi fitur nada diskrit yang luas, properti dominan dari area tersebut disebut tekstur.

Beberapa fitur tekstur seperti entropi, energi, kontras, dan homogenitas, dapat diekstraksi dari matriks *co-occurrence* $C(ij)$ tingkat keabuan suatu gambar. Matriks ini didefinisikan dengan terlebih dahulu menetapkan vektor perpindahan $d_{x,y} = (\delta x, \delta y)$ dan kemudian menghitung semua pasangan piksel yang dipisahkan oleh perpindahan $d_{x,y}$, dan memiliki tingkat keabuan i dan j . Matriks $C(ij)$ dinormalisasi dengan membagi

setiap elemen dalam matriks dengan jumlah total pasangan piksel. Dengan menggunakan matriks *co-occurrence* ini, metrik fitur tekstur dihitung sebagai berikut.

$$Entropy = - \sum_i \sum_j C(i, j) \log C(i, j), \quad (8)$$

$$Energi = \sum_i \sum_j C^2(i, j), \quad (9)$$

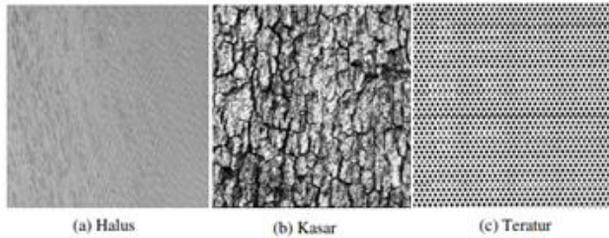
$$Kontras = \sum_i \sum_j (i - j)^2 C(i, j), \quad (10)$$

$$Entropy = \sum_i \sum_j \frac{C(i, j)}{1 + |i - j|}. \quad (11)$$

Pada umumnya, aplikasi tekstur dapat dibagi menjadi dua kategori. Pertama adalah untuk kepentingan segmentasi. Pada proses ini, tekstur dipakai untuk melakukan pemisahan antara satu objek dengan objek yang lain. Kedua adalah untuk klasifikasi tekstur, yang menggunakan fitur-fitur tekstur untuk mengklasifikasi objek. Beberapa contoh aplikasi tekstur disajikan di bawah ini (Tuceryan dan Jain, 1998).

1. Inspeksi secara otomatis pada industri tekstil, pengecatan mobil, pemakaian karpet.
2. Analisis citra medis. Misalnya, tekstur digunakan untuk klasifikasi penyakit paru-paru, diagnosis leukemia, dan pembedaan tipe-tipe sel darah putih.
3. Analisis penginderaan jarak-jauh. Misalnya, tekstur dipakai untuk kepentingan klasifikasi area tanah.

Kulkarni (1994) mendefinisikan tekstur sebagai hubungan mutual antara nilai intensitas piksel-piksel yang bertetangga yang berulang di suatu area yang lebih luas daripada jarak hubungan tersebut. Namun, penjelasan seperti itu pun masih menyisakan kesulitan untuk mengenali pengulangan yang terjadi pada citra. Ada suatu pengulangan yang terkadang sulit dijabarkan tetapi mudah ditangkap oleh mata, seperti yang terdapat pada Gambar 43(a) dan Gambar 43(b). Hal ini berbeda dengan Gambar 43(c). Citra yang disebut memiliki sifat pengulangan yang paling sulit dilihat dari ketiga gambar.



Gambar 43. Berbagai citra yang memiliki sifat tekstur yang berbeda-beda.

D. Fitur Warna

Warna adalah salah satu fitur visual yang paling banyak digunakan dalam pengambilan gambar berbasis konten karena memiliki daya tarik yang tinggi. Meskipun kita hanya dapat melihat tingkat abu-abu dalam jumlah terbatas, mata kita mampu membedakan ribuan warna dan komputer bahkan dapat mewakili jutaan warna yang dapat dibedakan dalam praktiknya. Warna telah berhasil diterapkan untuk mengambil gambar, karena memiliki korelasi yang sangat kuat dengan objek yang mendasari dalam sebuah gambar. Meskipun kita dapat menggunakan ruang warna apapun untuk menghitung histogram warna, seperti HSV, HLS, dan ruang warna CIE telah ditemukan menghasilkan warna yang lebih baik sebagai dibandingkan dengan ruang RGB. Karena ruang warna ini secara visual (atau perseptual) seragam dibandingkan dengan RGB, ruang warna ini ternyata lebih efektif untuk mengukur kesamaan warna di antara gambar. Contoh sederhana deteksi citra dengan memanfaatkan fitur warna adalah membedakan paprika berwarna merah dan hijau.

E. Topologi

Sebuah citra digital dapat diwakili oleh satu atau lebih properti topologi, yang biasanya mewakili bentuk geometris dari sebuah citra. Karakteristik menarik dari sifat topologi adalah ketika perubahan dilakukan pada gambar itu sendiri, seperti peregangan, deformasi, rotasi, penskalaan, translasi, atau transformasi dimana sifat gambar ini tidak berubah. Akibatnya,

sifat topologi bisa sangat berguna dalam karakterisasi gambar dan dapat digunakan sebagai ciri khas dari konten gambar untuk digunakan dalam pengambilan gambar berbasis konten.

Salah satu sifat topologi citra digital dikenal dengan istilah bilangan Euler. Bilangan Euler biasanya dihitung dalam citra biner. Namun, dapat juga diperluas untuk mengkarakterisasi gambar grayscale dan mendefinisikan vektor bilangan Euler dari bidang biner dari citra tersebut. Bilangan Euler didefinisikan sebagai selisih antara jumlah komponen yang terhubung dan jumlah lubang pada citra biner.

BAB 10

Pengantar *Computer Vision*

Computer Vision, sering disingkat CV, didefinisikan sebagai bidang studi yang berupaya mengembangkan teknik untuk membantu komputer melihat dan memahami konten gambar digital seperti foto dan video. Pada bab ini akan diuraikan pembahasan untuk memperkenalkan visi komputer dan cara kerjanya. bab

A. Pendahuluan

Perkembangan teknologi komputer pada dewasa ini telah mengalami kemajuan, termasuk dalam bidang teknologi citra digital. *Computer vision* membuat komputer dapat melihat dan mengenali suatu citra layaknya manusia. Dalam mengimplementasikan teknik interaksi visual untuk komputer tentunya membutuhkan salah satu cabang bidang ilmu komputer, salah satunya adalah *computer vision* yang akhir-akhir ini menjadi salah satu isu yang hangat diperbincangkan di dalam lingkungan ilmu komputer.

Computer Vision merupakan sebuah sistem otomatis yang digunakan untuk melakukan analisa citra dan video oleh komputer untuk memperoleh informasi dan pemahaman dari suatu obyek. Visi komputer biasanya dikatakan secara umum merupakan kemampuan komputer untuk menduplikasikan mata manusia yang baik dalam melakukan penangkapan atau perekaman citra, analisa citra, ataupun dalam pemahaman citra untuk diterapkan di dalam sebuah mesin ataupun komputer.

B. Pengertian *Computer Vision*

Visi komputer adalah sub-bidang *Deep Learning* dan *Artificial Intelligence* di mana manusia mengajar komputer untuk melihat dan menafsirkan dunia di sekitar mereka. Meskipun makhluk hidup seperti manusia dan hewan secara alami dapat melihat sejak dilahirkan, membantu mesin untuk memahami lingkungan yang sama melalui penglihatan tetap menjadi

tantangan. Dalam penerapannya terdapat kategori *vision* yang dibagi menjadi :

1. *Low-Level Vision*

Berada pada kategori paling awal tidak membuat visi level rendah berkurang efektivitasnya. Hal yang mendasari visi ini adalah bagaimana komputer memahami gambar sebagai objek digital seperti yang sudah dijelaskan dalam bab sebelumnya. Tugas mesin dalam tahap ini contohnya adalah menghilangkan derau pada gambar, memberi filter, mempertajam gambar, *thresholding*, klasifikasi tekstur permukaan dan lainnya.

2. *Intermediate-Level Vision*

Pada kategori *intermediate* perkembangan kemampuan visi yaitu dapat mendekteksi garis, tepi, dan pola. Pengenalan pola yang cukup sering ditemukan adalah analisis pola batas, pola tepi, lingkaran dan elips hingga pencocokan pola dari beberapa objek.

3. *3-D Vision* dan Gerakan

Pada kategori ini mesin diajarkan untuk menjadi sudut pandang pertama dalam melihat dunia layaknya manusia. Di dalam dimensi-3 mesin dilatih untuk dapat megambil informasi dari objek bergerak. Komputer dapat mengenali suatu benda bervolume dengan mengidentifikasi bentuk dan bayangan yang dihasilkan oleh pemberian cahaya ke permukaan benda. Bahkan perkembangan *3D Vision* sudah berkembang sangat pesat dan digunakan dalam kehidupan sehari-hari. Contohnya adalah bagaimana kamera pemantau milik kepolisian dan pemerintahan dapat melacak plat kendaraan saat berada di jalanan. Dapat diartikan bahwa deteksi *real-time* dengan kamera bukanlah hal yang asing. Namun demikian, masih banyak tantangan di masa depan yang akan ditempuh untuk membuat mesin menjadi lebih 'pintar' dalam mengambil keputusan dari data yang sudah diperolehnya.

C. Cara Kerja *Computer Vision*

Computer Vision adalah ekstraksi otomatis informasi dari gambar. Informasi dapat berarti apa saja mulai dari model 3D, posisi kamera, deteksi dan pengenalan objek hingga pengelompokan dan pencarian konten gambar. Intinya, tugas utama visi komputer adalah tentang bagaimana membuat komputer memahami gambar digital serta data visual dari dunia nyata. Dalam menganalisis informasi dari *input* tersebut untuk membuat keputusan melibatkan beberapa proses yaitu sebagai berikut.

1. Pengolahan sinyal/citra

Pengolahan citra digital adalah manipulasi dan interpretasi digital dari citra dengan bantuan komputer. *Input* dari pengolahan citra adalah gambar, sedangkan *output*-nya adalah citra hasil pengolahan. Citra secara umum merupakan suatu gambar, foto ataupun berbagai tampilan dua dimensi yang menggambarkan suatu visualisasi objek. Citra dapat diwujudkan dalam bentuk tercetak ataupun digital. Citra digital adalah larik angka-angka secara berdimensi-2 yang tersimpan dalam suatu bentuk larik (*array*) angka digital yang merupakan hasil kuantifikasi dari tingkat kecerahan masing-masing piksel penyusun citra tersebut.

Operasi-operasi pada pengolahan citra diterapkan bila:

- a. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra (*image enhancement*). Contoh : perbaikan kontras gelap/terang, perbaikan tepian objek, penajaman, pemberian warna semu, dll.
- b. Adanya cacat pada citra sehingga perlu dihilangkan/diminimumkan (*image restoration*). Contohnya penghilangan kesamaran (*deblurring*) citra tampak kabur karena pengaturan fokus lensa tidak tepat / kamera goyang, penghilangan *noise*.

- c. Elemen dalam citra perlu dikelompokkan, dicocokkan atau diukur (*image segmentation*). Operasi ini berkaitan erat dengan pengenalan pola.
- d. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek (*image analysis*). Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya.
- e. Sebagian citra perlu digabung dengan bagian citra yang lain (*image reconstruction*). Contohnya beberapa foto *rontgen* digunakan untuk membentuk ulang gambar organ tubuh.
- f. Pemampatan citra (*image compression*). Contohnya suatu file citra berbentuk BMP berukuran 258 KB dimampatkan dengan metode JPEG menjadi berukuran 49 KB. Biasanya dilakukan untuk menghemat penyimpanan pada *server* yang menerima banyak gambar *input*, ataupun saat transmisi gambar agar lebih cepat terkirim.
- g. Menyembunyikan data rahasia (berupa teks/gambar) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang (*steganografi* dan *watermarking*).

2. Klasifikasi Citra

Klasifikasi objek pada citra secara umum adalah masalah utama dalam *Computer Vision* yang sejak dahulu dicari solusinya. Klasifikasi citra merupakan salah satu proses analisis data secara numeris bertujuan untuk identifikasi kenampakan atau pengenalan pola. Pengenalan pola objek (*spectral pattern recognition*) adalah mengevaluasi informasi objek berdasarkan ciri citra penginderaan jauh sebagai dasar interpretasi citra digital. Pengenalan pola sistem meliputi kategorisasi pixel citra dengan basis hubungan spasial antar piksel-piksel tersebut.

Salah satu metode *Artificial Intelligence* untuk pemrosesan citra digital *Convolutional Neural Network* (CNN) yaitu jenis khusus dari Neural Network satu bidang dari *Deep*

Learning yang dapat digunakan untuk klasifikasi citra objek. CNN merupakan algoritma *Artificial Intelligence* yang populer pada saat ini. Model arsitektur pada metode CNN juga dapat dikembangkan dengan *transfer learning*. CNN model menggunakan *pretrained* atau sebelumnya sudah dilakukan proses pelatihan oleh orang lain atau peneliti dengan menggunakan *dataset* yang sangat besar salah satunya menggunakan dataset *imagenet*.

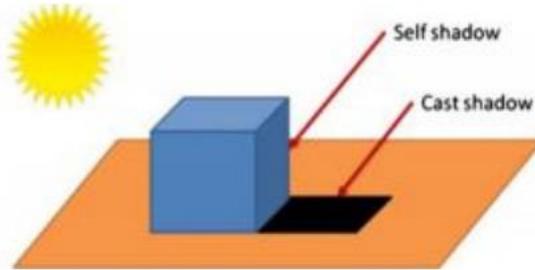
3. Pengenalan Bentuk dan Bayangan

a. Bentuk

Bentuk merupakan variabel kualitatif yang memberikan konfigurasi atau kerangka suatu objek. Bentuk adalah gambar yang mudah dikenali contohnya adalah gedung sekolah pada umumnya berbentuk huruf I, L dan U atau persegi panjang, roda berbentuk lingkaran, planet berbentuk bola, dan lain sebagainya.

b. Bayangan

Bayangan sering menjadi kunci pengenalan yang penting bagi beberapa objek dengan karakteristik tertentu. Bayangan bersifat menyembunyikan detail atau objek yang berada di daerah gelap. Dengan adanya bayangan, objek primer akan menjadi lebih jelas. Contohnya lereng terjal tampak lebih jelas dengan adanya bayangan, begitu juga cerobong asap dan menara, tampak lebih jelas dengan adanya bayangan. Foto-foto yang sangat condong biasanya memperlihatkan bayangan objek yang tergambar dengan jelas. Bayangan dapat dibagi menjadi dua kelas yaitu *self shadow* dan *cast shadow*. Kedua banyanya tersebut memiliki tingkat kecerahan yang berbeda. Ilustrasi *self shadow* dan *cast shadow* dapat dilihat pada Gambar 44.



Gambar 44. Ilustrasi *self shadow* dan *cast shadow*

Self shadow merupakan daerah bayangan pada objek itu sendiri. *Self shadow* juga biasanya memiliki tingkat kecerahan yang lebih tinggi daripada *cast shadow* karena bagian tersebut mendapatkan lebih banyak pencahayaan sekunder dari benda-benda disekitarnya. *Cast shadow* terdiri dari dua bagian yaitu umbra dan penumbra. Umbra adalah bagian dari *cast shadow* dimana cahaya langsung sepenuhnya diblokir atau terhalangi oleh objek. Sedangkan penumbra adalah bagian dari *cast shadow* dimana cahaya langsung hanya terhalangi secara parsial atau sebagian.

4. *Filtering, Contrast, dan Shading*

a. *Filtering*

Merupakan salah satu teknik pengolahan citra yang banyak digunakan. *Filtering* menggunakan prinsip pengolahan citra sederhana dimana setiap band pada sebuah citra dianggap sebagai sebuah matriks. Selanjutnya, operasi matematika dilakukan pada matriks tersebut untuk memperoleh atau menguatkan fitur tertentu yang ada pada citra tersebut. Operasi *Image Filtering* sering digunakan sebagai langkah awal dalam pengolahan (*pre-processing*) citra, misalnya untuk menghilangkan *noise* atau memperkuat fitur dengan ciri tertentu pada citra.

Image Filtering bekerja dengan cara menjalankan sebuah '*kernel*' atau matriks persegi dengan ukuran tertentu pada seluruh citra secara bertahap. Inilah

mengapa operasi kernel pada seringkali disebut sebagai '*moving window*'.

b. *Contrast*

Metode yang dilakukan dengan menambah atau mengurangi kontras (pencahayaan) agar citra tersebut lebih tajam dari citra aslinya. Kontras adalah rentang antara titik tergelap dan titik terterang dalam suatu gambar. Rincian halus muncul menjadi terbaik bila ada kontras tinggi di antara pixel yang berdekatan warna objek yang berdekatan, intensitas, posisi cahaya dan distorsi sistemnya bisa menyembunyikan sisi dan batas karena ada kontras yang tidak sempurna. Dalam kasus semacam itu, kontras dapat diperbaiki dengan cara pemrosesan. Yang mana dapat dilakukan dengan mengidentifikasi rentang nilai intensitas piksel yang ditekan atau dinaikkan. Dengan kata lain, nilai *threshold* atas dan bawah sudah di-set dan setiap piksel yang jatuh pada rentang di atas akan dimodifikasi. Modifikasi dilakukan dengan menambah, mengurangi, mengalikan, dan membagi nilai-nilai piksel dengan konstan tergantung kepada penghalusan yang dibutuhkan.

c. *Shading*

Shading adalah proses pengolahan dengan cara memberikan atau mengulaskan warna lebih tua, dimana warna tersebut akan membuat efek mengecilkan daerah yang diulaskan pewarna tersebut. Prinsip yang mendasari teknik ini adalah pemodelan reflektansi objek di tempat kejadian sebagai fungsi dari sudut datang dan munculnya cahaya dari permukaannya. Bentuk dari teknik *shading* tidak menyimpulkan kedalaman tetapi normal permukaan permukaan dan dengan demikian membentuk kelas teknik rekonstruksi permukaan yang sama sekali baru. Jelas bahwa bentuk dari teknik *shading* tidak dapat menyimpulkan jarak absolut.

BAB 11

Perkembangan *Computer Vision*

Saat ini, visi komputer adalah sebuah divisi kecerdasan buatan yang menangani pengajaran komputer bagaimana 'melihat' gambar dan objek 2D dan 3D. Didukung oleh pembelajaran mendalam, visi komputer mahir dalam mengidentifikasi dan memproses data visual dalam jumlah besar dengan cepat dan akurat dan membuat keputusan atau rekomendasi berdasarkan informasi tersebut. Pada bab ini diuraikan bagaimana perkembangan *computer vision* dan penerapannya dalam kehidupan sehari-hari.

A. Sejarah Perkembangan *Computer Vision*

Ketika visi komputer pertama kali dimulai pada awal 1970-an, yang dipandang sebagai komponen persepsi visual dari usaha ambisius ilmuwan untuk meniru kecerdasan manusia dan memberi robot perilaku cerdas layaknya manusia. Pada saat itu, itu diyakini oleh beberapa orang awal. Dengan ditemukannya kecerdasan buatan di tahun 1950-an, semakin mudah untuk pula mengimplementasikan *computer vision* dalam penelitian. Menurut (Szeliski, 2022) berikut adalah perkembangan topik penelitian di bidang visi komputer:

Tahun	Topik
1970	Pemrosesan gambar digital, Blok dunia dan pelabelan garis Silinder umum, Pengenalan pola Korespondensi stereo.
1980	Gambar intrinsik, Aliran optik, Struktur dari gerakan, Piramida gambar, Bentuk dari shading/tekstur/fokus, Pemodelan berbasis fisik, Regularisasi.
1990	Markov bidang acak, Filter Kalman, Pemrosesan data rentang 3D, Invarian proyektif, Faktorisasi, Visi berbasis fisika.

2000	Potongan grafik, Penyaringan partikel, Segmentasi berbasis energi, Pengenalan dan deteksi wajah, Pemodelan dan rendering berbasis gambar.
2010	Sintesis tekstur dan pengecatan, Fotografi komputasi, Pengenalan berbasis fitur, Pengenalan kategori, Pembelajaran mesin (<i>Machine Learning</i>).
2020	Memodelkan dan melacak manusia, Segmentasi semantik, SLAM dan VIO, <i>Deep Learning</i> , <i>Vision</i> dan bahasa.

Tabel 3. Perkembangan *computer vision* berdasarkan topik penelitian 6 dekade terakhir.

B. Penerapan *Computer Vision*

Penerapan visi komputer pada kehidupan manusia beragam pada berbagai aspek. Hal ini dikarenakan efektivitas bantuan mesin dalam menganalisis citra baik 2D dan 3D terbukti dan terjamin hasilnya. Berbagai aspek krusial yang membutuhkan pembuatan keputusan penting bahkan sudah menerapkan bantuan alat pendeteksi citra, seperti contohnya dalam bidang kesehatan dengan adanya klasifikasi sel tumor dari hasil rontgen. Berikut adalah beberapa contoh penerapan *computer vision* dalam berbagai aspek :

1. Pemrosesan gambar biometrik dan biomedis

Sistem identifikasi biometrik berguna dalam beberapa aplikasi seperti aplikasi komersial dan penegakan hukum, terutama dalam identifikasi kriminal, sistem keamanan, *videophone*, verifikasi kartu kredit, KTP, foto untuk identifikasi pribadi, dll. Pengenalan wajah manusia, sidik jari, tanda tangan, dan banyak gambar biometrik lainnya merupakan bidang penelitian penting di bidang visi komputer. Demikian pula ada berbagai jenis modalitas pencitraan biomedis seperti *X-Ray*, *Computed Tomography* (CT), *Magnetic Resonance Imaging* (MRI), gambar *ultrasound*, dan banyak lainnya yang digunakan di bidang medis untuk

diagnosis dan pengobatan penyakit. Berikut adalah contoh penerapan yang sering ditemukan:

- a. Pengenalan pola biometrik, contohnya adaah seleksi fitur, ekstraksi fitur wajah bagian depan, ekstraksi fitur wajah bagian samping, maupun identifikasi wajah.
- b. Pengenalan wajah menggunakan *Eigenfaces* / *Fisherfaces*.
- c. Verifikasi tanda tangan.
- d. Analisis citra biomedis, yang terdiri dari analisis citra mikroskopis maupun makroskopis.
- e. Citra *X-Ray*. Pada bidang kesehatan visi komputer untuk citra rontgen bervariasi dari identifikasi penyakit jantung, identifikasi penyakit tulang, identifikasi tulang rusuk, dan banyak lainnya.
- f. Analisis citra *X-Ray* dental.

2. Analisis pemandangan multispektral penginderaan jarak jauh

Informasi mengenai sumber daya alam suatu negara sangat berguna untuk tujuan perencanaan. Sumber daya termasuk sumber daya pertanian (misalnya, persediaan makanan seperti beras, gandum, sayuran, dll.), sumber daya hidrologi (misalnya, badan air seperti sungai, kanal, kolam, dll.), sumber daya mineral (misalnya, tambang logam, batu bara, dll), sumber daya hutan, sumber daya geologi dan strategis, dan berbagai sumber daya lainnya. Informasi tersebut dapat dengan mudah diekstraksi dari gambar penginderaan jauh dengan memanfaatkan data citra satelit dan udara untuk memperkirakan sumber daya yang tersedia di suatu negara. Untuk analisis pemandangan penginderaan jauh, gambar objek di permukaan bumi ditangkap oleh sensor di satelit penginderaan jauh atau oleh pemindai multispektral yang ditempatkan di pesawat terbang. Gambar-gambar ini kemudian dikirim ke stasiun bumi untuk diproses lebih lanjut. Berikut contoh pemanfaatan penginderaan jarak jauh.

- a. Sensor dan citra satelit. Citra satelit sangat bermanfaat pada perkembangan teknologi dan aspek lainnya. Baik objek yang ada di bumi maupun objek luar angkasa dapat dilihat dengan bantuan satelit dan teleskop canggih. Dengan adanya penginderaan jauh tersebut sangat bermanfaat dalam penelitian dan pendidikan umat manusia.
- b. Beragam citra dari objek-objek di bumi. Citra yang diambil jauh dari permukaan bumi dapat dengan mudah mendeteksi wilayah perairan, wilayah vegetasi (untuk bercocok tanam), tanah, maupun objek buatan manusia. Sehingga tanpa harus melakukan perjalanan udara berkilo-kilometer jauhnya, informasi tersebut dapat dengan mudah dipanen dari data citra yang disimpan oleh satelit.
- c. Klasifikasi pendekatan berbasis pengetahuan. Selain dapat melihat objek yang ada di bumi data citra juga dapat diklasifikasikan untuk membedakan objek natural dan *artificial*. Hal ini dapat dilihat nyata dalam persebaran penduduk di suatu wilayah untuk mengukur tingkat kepadatan huni suatu lokasi.

3. Deteksi dan pelacakan objek bergerak

Deteksi dan klasifikasi objek bergerak merupakan bidang penelitian penting dalam visi komputer cukup *advance*. Bidang ini menjadi sangat penting karena fakta bahwa dunia visual kita yang dinamis dan kita terus-menerus menemukan adegan video yang berisi sejumlah besar objek bergerak. Untuk mengelompokkan, mendeteksi, dan melacak objek-objek ini dari rangkaian gambar video mungkin merupakan tantangan paling penting yang dihadapi para ahli saat ini. Sistem deteksi objek bergerak sangat bermanfaat dalam kehidupan, misalnya saja aplikasi dalam pengawasan manusia, sistem keamanan, pemantauan lalu lintas, visi industri, pengawasan pertahanan, dan lain sebagainya adalah bukti penerapannya. Prinsip penting

dalam deteksi yaitu adanya bayangan. Untuk melacak objek 3D dalam lingkungan sekitar dibutuhkan kuantitas cahaya yang memadai untuk mendapatkan hasil deteksi yang maksimal. Walaupun begitu deteksi pada wilayah gelap bukanlah hal yang tidak mungkin karena perangkat kamera digital masa kini sudah dilengkapi fitur *night vision*.

BAB 12

Machine Learning dalam Computer Vision

Bayangkan kita sedang duduk di teras rumah, mengamati apa yang terjadi di sekitar rumah. Ada dua sistem di tubuh kita yang sedang bekerja yaitu mata yang bertindak sebagai sensor dan menciptakan representasi pemandangan, sementara sistem kognitif memahami apa yang dilihat oleh mata. Kita mungkin melihat orang yang berlari tergesa-gesa, dedaunan yang tertiuip angin, beberapa gerakan lainnya dan menyadari bahwa angin telah bertiup dengan kencang pertanda akan turun hujan. Proses penarikan kesimpulan bahwa hari akan hujan dapat terjadi karena adanya proses pemahaman terhadap visi.

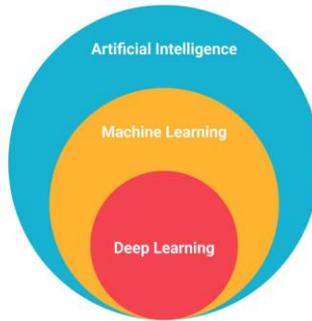
Di lain sisi, *Computer Vision* mencoba meniru kemampuan penglihatan manusia dengan menyediakan metode untuk pembentukan gambar dan persepsi mesin. Peniruan sistem sensorik manusia difokuskan pada perangkat keras dan pada desain dan penempatan sensor seperti kamera. Pendekatan modern untuk meniru sistem kognitif manusia terdiri dari metode pembelajaran mesin (ML) yang digunakan untuk mengekstrak informasi dari gambar. Sebelum masuk ke dalam pembahasan machine learning, terlebih dahulu mari kita kenali dasarnya yaitu *Artificial Intelligence*.

A. Pengantar *Artificial Intelligence* (AI)

Dalam buku ini sering digunakan istilah "*Artificial Intelligence*", "*Machine Learning*", dan "*Deep Learning*". Ketiga hal tersebut terdengar serupa namun tidak sama. Berikut pengertian singkatnya menurut (Kou, Ganju, & Kasam, 2019).

1. *Artificial Intelligence* (AI) memberi mesin kemampuan untuk meniru perilaku manusia. Deep Blue IBM adalah contoh AI yang dapat dikenali.
2. *Machie Learning* (ML) adalah cabang AI di mana mesin menggunakan teknik statistik untuk belajar dari informasi dan pengalaman sebelumnya. Tujuannya adalah agar mesin mengambil tindakan di masa depan berdasarkan pengamatan pembelajaran dari masa lalu.

3. *Deep Learning* (DL) adalah sub-bidang pembelajaran mesin di mana *jaringan* saraf yang dalam dan berlapis-lapis digunakan untuk membuat prediksi, terutama unggul dalam *computer vision*, *speech recognition*, pemahaman bahasa alami, dan sebagainya.



Gambar 45. Hubungan AI, ML, dan DL.

Pendekatan *deep learning* adalah kelas algoritma *machine learning* yang menggunakan banyak lapisan unit pemrosesan nonlinier untuk representasi dan transformasi. Setiap lapisan menggunakan output dari lapisan sebelumnya sebagai input dan representasi hierarkis dapat diperoleh dengan tingkat abstraksi yang berbeda. Algoritma ini dapat dilatih dengan cara yang diawasi atau tidak, dan aplikasinya mencakup analisis pola (*unsupervised*) dan klasifikasi (*supervised*). Saat ini, perkembangan *deep learning* yang pesat telah dipercepat oleh tiga alasan utama, yaitu data yang sangat besar, kemampuan komputasi yang kuat, dan algoritma-algoritma baru. Di masa depan, semakin banyak *domain* akan difasilitasi oleh *deep learning* dan sebaliknya memberikan kontribusi kembali ke teknologi pembelajaran mendalam (Jiang, Hadid, Pang, Granger, & Feng, 2019).

Berikut di mana *deep learning* telah digunakan dalam permasalahan *computer vision* (Brownlee, 2019).

1. Klasifikasi Gambar
2. Klasifikasi Gambar Dengan Lokalisasi
3. Deteksi Objek
4. Segmentasi Objek

5. Transfer *Style* Gambar
6. Pewarnaan Gambar
7. Rekonstruksi Gambar
8. Gambar *Super-Resolution*
9. Sintesis Gambar
10. Dan lain sebagainya

B. Pengantar *Machine Learning*

Machine Learning adalah praktik pemrograman komputer untuk belajar dari data. Dalam contoh di atas, program akan dengan mudah dapat menentukan apakah yang diberikan penting atau "*spam*". *Machine Learning* berisi sebuah algoritma yang bersifat generik (umum) dimana algoritma tersebut dapat menghasilkan sesuatu yang menarik atau bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik. Pada intinya, algoritma yang generik tersebut ketika diberikan sejumlah data maka ia dapat membangun sebuah model atau inferensi dari data tersebut.

Machine learning adalah ilmu pengembangan algoritma dan model secara statistik yang digunakan sistem komputer untuk menjalankan tugas tanpa instruksi eksplisit, mengandalkan pola dan inferensi sebagai gantinya. Istilah *machine learning* pada dasarnya menjelaskan proses komputer dalam mempelajari data. Oleh karena itu, kita pasti akan terus bersinggungan dengan data ketika mempelajari *machine learning*. Sistem komputer menggunakan algoritma *machine learning* untuk memproses data historis berjumlah besar dan mengidentifikasi pola data. Hal ini memungkinkannya untuk memprediksi hasil yang lebih akurat dari set data input yang diberikan.

Machine learning sendiri merupakan salah satu cabang dari disiplin dalam kecerdasan buatan (*artificial intelligent*) yang membahas pembangunan sistem berdasarkan data. Program *machine learning* mengikuti cara belajar manusia, yakni belajar dari contoh dan masalah sebelumnya. *Machine learning* akan mempelajari pola dari contoh yang dianalisis untuk menentukan

jawaban dari pertanyaan-pertanyaan berikutnya. Memang, tidak semua masalah bisa dipecahkan dengan program *machine learning*, namun seringkali algoritma yang bersifat kompleks ternyata bisa dipecahkan dengan mudah oleh *machine learning*. Beberapa contoh program berbasis *machine learning* yang digunakan dalam kehidupan sehari-hari diantara adalah:

1. Pada bidang bisnis, *machine learning* sangat penting membantu bisnis dengan cara mendorong pertumbuhan, membuka aliran pendapatan baru, dan menyelesaikan masalah-masalah yang menantang. Data adalah penggerak yang penting di balik pengambilan keputusan bisnis, tetapi secara tradisional, perusahaan telah menggunakan data dari berbagai sumber, seperti umpan balik pelanggan, karyawan, dan keuangan.
2. Pada bidang manufaktur, *machine learning* dapat mendukung pemeliharaan prediktif, kontrol kualitas, dan riset inovatif dalam sektor manufaktur. Teknologi *machine learning* juga membantu perusahaan meningkatkan solusi logistik, termasuk aset, rantai pasokan, dan manajemen inventaris.
3. Perkembangan sensor dan perangkat sandangan telah menghasilkan volume data kesehatan yang signifikan. Pada bidang kesehatan, *machine learning* dapat menganalisis informasi dan mendukung para dokter mendiagnosis serta merawat pasien secara *real time*. Para peneliti sedang mengembangkan solusi mendeteksi tumor, kanker dan mendiagnosis penyakit mata, yang memengaruhi hasil kondisi kesehatan manusia secara signifikan.
4. Proyek *machine learning* pada bidang layanan keuangan meningkatkan analitik dan regulasi risiko. Teknologi *machine learning* dapat memungkinkan para investor untuk mengidentifikasi peluang baru dengan cara menganalisis pergerakan pasar saham, mengevaluasi pengelola investasi global, atau melakukan kalibrasi portofolio keuangan. Selain itu, teknologi *machine learning* dapat membantu mengidentifikasi klien pinjaman berisiko tinggi dan mengurangi tanda-tanda penipuan.

5. Pada perusahaan hiburan, *machine learning* bermanfaat untuk memberi informasi terkait konten yang sesuai permintaan dan kebutuhan konsumen. Algoritma *machine learning* melakukan *deployment* untuk mendesain trailer serta iklan lainnya, memberikan rekomendasi konten yang dipersonalisasi kepada konsumen.

C. Klasifikasi *Machine Learning*

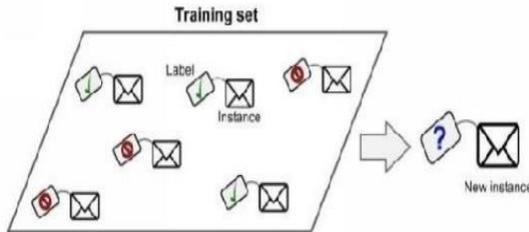
Machine learning dapat dikelompokkan berdasarkan bagaimana cara belajar sehingga dapat melakukan tugasnya. Terdapat pembagian *Machine learning* menjadi beberapa kelompok sebagai berikut:

1. *Supervised Learning*

Secara bahasa, *supervised learning* adalah pembelajaran terarah/terawasi. Dalam jenis sistem *Machine learning* ini, data yang dimasukkan ke dalam algoritma, dengan solusi yang diinginkan, disebut sebagai "label". *Supervised Learning* merupakan sebuah pemodelan dimana algoritmanya dapat membangkitkan suatu fungsi yang memetakan input ke output yang diinginkan. Pada *Supervised Learning* data yang memiliki label sehingga tujuan pengolahan tersebut adalah mengelompokkan data ke dalam kelompok yang sudah ada. Algoritma ini memerlukan data berlabel untuk membangun sebuah model yang tingkat akurasinya bisa ditingkatkan dari waktu ke waktu. Semakin banyak model tersebut mengolah data, maka tingkat keakurasiannya juga akan semakin tinggi.

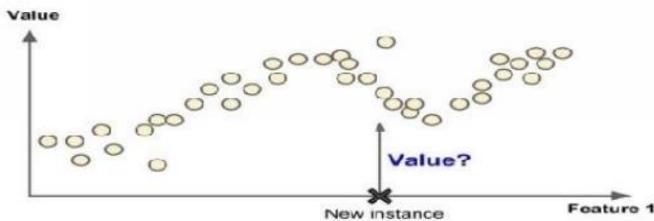
Dalam algoritma *supervised learning*, terdapat dua variabel, yaitu variabel input yang biasa disebut variabel X dan variabel output yang biasa disebut variabel Y. Proses pengolahan data yang dilakukan jika menggunakan *Supervised Learning* memerlukan *data training*, *data training* sendiri digunakan dalam memprediksi maupun mengklasifikasi data. Tujuan algoritma *supervised learning* adalah untuk mempelajari fungsi pemetaan dari variabel X ke variabel Y. Rumus umum pemetaan variabel X dan Y adalah $Y = f(X)$.

Tujuan akhir dari algoritma supervised learning adalah untuk memperkirakan fungsi pemetaan (f) agar kita dapat memprediksi variabel Y ketika kita memiliki data input (variabel X) yang baru.



Gambar 46. Contoh pelatihan dengan menggunakan banyak e-mail disaat yang sama dengan kelas masing-masing.

Supervised learning mengelompokkan berbagai tugas dari klasifikasi. program di atas merupakan contoh yang bagus karena telah dilatih dengan menggunakan banyak email disaat yang sama dengan kelas masing-masing . Contoh lain adalah untuk memprediksi nilai numerik seperti harga sebuah apartemen, diberikan *serangkaian* fitur (lokasi, jumlah kamar, fasilitas) yang disebut prediktor; jenis tugas ini disebut regresi.



Gambar 47. Contoh memprediksi nilai numerik dari harga sebuah apartemen.

Adapun contoh-contoh yang termasuk algoritma yang termasuk dalam kategori *Supervised learning* sebagai berikut:

- a. *K-Nearest Neighbors* (KNN)
- b. *Linear regression* (Regresi Linear)

- c. *Feature Extraction*
- d. *Naïve Bayes*
- e. *Neural Network*
- f. *Support Vector Machines (SVM)*
- g. *Logistic Rregression (Regresi logistik)*
- h. *Decision Ttrees dan Random Forests*

2. *Unsupervised Learning*

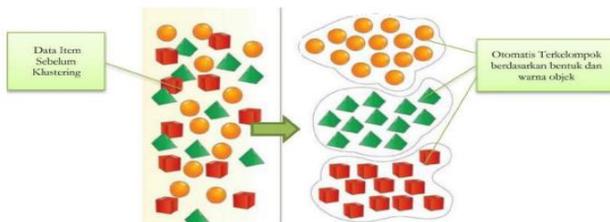
Berbeda dengan *supervised learning*, *Unsupervised Learning* adalah salah satu tipe algoritma machine learning yang digunakan untuk menarik kesimpulan dari dataset. Metode ini hanya akan mempelajari suatu data berdasarkan kedekatannya saja atau yang biasa disebut dengan clustering. Metode unsupervised learning yang paling umum adalah analisis cluster, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data.

Pada algoritma unsupervised learning, data tidak memiliki label secara eksplisit dan model mampu belajar dari data dengan menemukan pola yang implisit. Sangat berbeda dengan supervised learning, unsupervised learning merupakan jenis learning yang hanya mempunyai variabel input tapi tidak mempunyai variabel output yang berhubungan. Tujuan dari *Machine Learning* ini adalah untuk memodelkan struktur data dan menyimpulkan fungsi yang mendeskripsikan data tersebut.

Cara kerja algoritma ini yaitu mencari pola tersembunyi (eksplisit) dari dataset yang diberikan. *Unsupervised machine learning* bekerja dengan menganalisis data yang tidak berlabel untuk menemukan pola tersembunyi dan menentukan korelasinya. Pendekatan ini tidak menggunakan data training dan data test untuk melakukan prediksi maupun klasifikasi dengan tujuan mengelompokkan objek yang hampir sama dalam suatu area tertentu. Salah satu contoh implementasi *unsupervised learning* adalah clustering. Dengan teknologi clustering,

sebuah perusahaan dapat mengidentifikasi customer segmen yang berpotensi untuk menjual produk mereka. Perusahaan dapat mengidentifikasi customer segmen yang paling mungkin menggunakan layanan atau membeli produk mereka. Perusahaan juga dapat mengevaluasi segmen pelanggan, lalu memutuskan untuk menjual produk guna memaksimalkan keuntungan mereka

Sebagai contoh, pada Gambar 48 berisi dua jenis item yang berbeda. Kedua item tersebut akan dipisah menjadi beberapa kategori tergantung data. Komputer hanya mengetahui fitur-fitur yang akan digunakan untuk membedakan kedua item tersebut yaitu warna dan bentuk. Dengan menggunakan algoritma *clustering*, komputer akan dapat membagi item-item menjadi dua kelompok tanpa harus diberi pengetahuan. Algoritma akan bekerja untuk membagi menjadi beberapa kelompok dengan melihat isi data masing-masing item. Contoh algoritma *Unsupervised learning* sederhana dan sering digunakan adalah *K-means*.



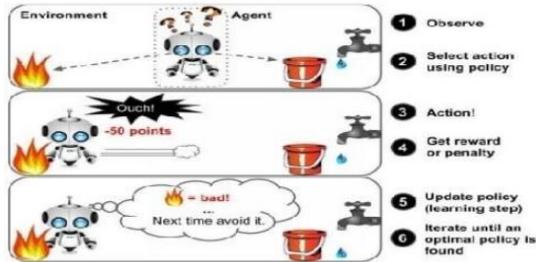
Gambar 48. Kedua item yang dipisah menjadi beberapa kategori tergantung data.

Berikut adalah algoritma yang termasuk dalam kategori *unsupervised learning*:

- Clustering* : *K-means* dan *Hierarchical Cluster Analysis*
- DBSCAN
- Fuzzy C-Means
- Association rule learning* : *Eclat*, *apriori*
- Self-Organizing Map*
- Visualization and dimensionality reduction* : *kernel PCA*, *t-distributed PCA*

3. Reinforcement Learning

Reinforcement learning adalah jenis lain dari sistem *Machine learning*. Agen "sistem AI" akan mengamati lingkungan, melakukan tindakan yang diberikan, dan kemudian menerima imbalan sebagai balasannya. Dengan tipe ini, agen harus belajar dengan sendirinya.



Gambar 49. Contoh Agen "sistem AI" yang mengamati lingkungan, melakukan tindakan yang diberikan, dan kemudian menerima imbalan sebagai balasannya.

Machine learning dapat dikategorikan berdasarkan proses pembelajaran dilakukan secara bertahap (*Batch Learning*) atau secara langsung (*on the fly Online Learning*). Prinsip dasar dari *online learning* adalah menghasilkan sebuah model yang melakukan proses pembelajaran data baru secara realtime atau mendekati *realtime*. Sedangkan *Batch Learning*, data *training* akan dipecah-pecah menjadi beberapa bagian lalu setiap bagiannya akan dipelajari secara terpisah pada waktu yang berbeda.

Machine Learning juga dikelompokkan berdasarkan bagaimana cara kerjanya yaitu sebagai berikut:

a. Instance Based Learning

Instance based learning adalah sebuah kelompok algoritma *Machine learning* yang bekerja dengan membandingkan data *testing* dengan data yang telah dipelajari pada proses *training*. Algoritma kelompok ini tidak membuat sebuah generalisasi melainkan ke arah perbandingan dengan data yang disimpan di memori.

Contoh yang termasuk ke kelompok ini adalah *K-Nearest Neighbors*, *kernel machines*, dan *RBF network*.

b. *Model Based Learning*

Model based learning adalah kebalikan dari *instance based* dimana menggunakan memori untuk melakukan pemecahan masalah, algoritma ini membuat sebuah model yang bersifat generic

Dari klasifikasi *machine learning* yang sudah dijabarkan di atas, dapat dilihat perbandingan sederhana dalam Tabel 3 berikut ini:

Kriteria	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definisi	Belajar dengan menggunakan data berlabel	Dilatih menggunakan data yang tidak berlabel tanpa panduan apa pun.	Bekerja untuk berinteraksi dengan lingkungan
Tipe Input Data	Data berlabel	Data tidak berlabel	Data sudah didefinisikan sebelumnya
Tipe Pemasalahan	Regresi, Klasifikasi, dan mempelajari pola input label	Asosiasi dan Clustering	Eksploitasi atau Eksplorasi
Pembangunan Model	Model dibangun dan dilatih sebelum pengujian	Model dibangun dan dilatih sebelum pengujian	Model ini dilatih dan diuji secara bersamaan
Pengawasan	Pengawasan Ekstra	Tidak diawasi	Tidak diawasi
Algoritma	Linear Regression, Logistic Regression, SVM, KNN, Naïve Bayes, Random Forest, Neural Networks	K - Means, Hierarchical clustering, C - Means, Apriori, Association Rule, DBSCAN	Q - Learning, SARSA, Markov Decision Process
Tujuan	Menemukan persamaan pemetaan pada data input dan labelnya	Menemukan fitur serupa dalam data input yang mendasari untuk diklasifikasi ke dalam kelas.	Mempelajari serangkaian tindakan dengan mengikuti metode trial-and-error
Aplikasi	Evaluasi risiko, Prakiraan penjualan, Deteksi gambar, Prediksi pertumbuhan populasi	Segmentasi pelanggan, Sistem rekomendasi, Deteksi anomaly, Target pasar	Mobil kendali otomatis, Gaming, robot penyedot debu, Layanan kesehatan, Natural Language Processing (NLP)

Tabel 4. Perbandingan 3 klasifikasi Machine Learning.

BAB 13

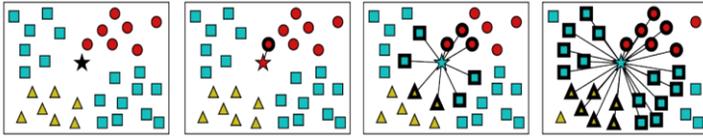
Teknik dan Algoritma dalam *Computer Vision*

Sebelumnya pada Bab 12 telah dijelaskan klasifikasi *machine learning* yang dibagi menjadi pembelajaran diawasi dan pembelajaran tidak diawasi. Pada bab ini akan dirangkum berbagai algoritma yang sering digunakan dalam visi komputer. Mulai dari metode *deep learning* hingga *Convolutional Neural Network* (CNN).

A. *K-Nearest Neighbors*

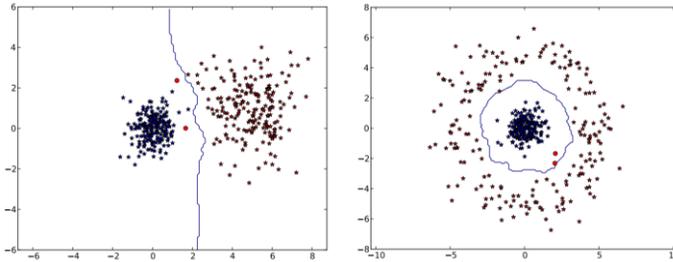
Menurut (Solem, 2012) salah satu metode klasifikasi yang paling sederhana dan paling sering digunakan adalah *K-Nearest Neighbor Classifier* (kNN). Algoritma ini hanya membandingkan objek (misalnya vektor fitur) untuk diklasifikasikan dengan semua objek dalam set pelatihan dengan label kelas yang diketahui dan memungkinkan pemilihan k terdekat untuk kelas mana yang akan ditetapkan dengan *voting*. Metode ini memiliki kinerja yang baik tetapi memiliki sejumlah kelemahan. Sama seperti algoritma *clustering K-means*, jumlah k perlu dipilih dan nilainya akan mempengaruhi kinerja. Selanjutnya, metode tersebut membutuhkan seluruh set pelatihan untuk disimpan dan jika set ini besar maka pencarian akan menjadi lambat. Untuk set pelatihan besar, beberapa bentuk *binning* biasanya digunakan untuk mengurangi jumlah perbandingan yang dibutuhkan. Sisi positifnya, tidak ada batasan ukuran jarak yang digunakan dan Algoritma ini juga dapat diparalelkan dengan mudah.

K-Nearest Neighbors adalah teknik non-parametrik yang sangat sederhana, yaitu teknik yang tidak melibatkan bentuk analitik parameter rendah untuk distribusi yang *mendasarinya*. Sebagai gantinya, semua contoh pelatihan dipertahankan, dan pada waktu evaluasi k tetangga "terdekat" ditemukan dan kemudian dirata-rata untuk menghasilkan *output* (Szeliski, 2022).



Gambar 50. Klasifikasi *K-Nearest Neighbors* menentukan objek tetangga terdekat dengan objek bintang.

Berikut adalah contoh sederhana penerapan KNN pada gambar 2D. Untuk setiap contoh warna menunjukkan label kelas (biru dan merah). Titik yang diklasifikasikan dengan benar ditunjukkan *dengan* bintang dan titik yang salah diklasifikasikan dengan lingkaran. Seperti tampak pada Gambar 51 di bawah, kedua buah lingkaran merah tidak berada pada klasifikasi yang tepat yaitu di luar kurva batas keputusan pengklasifikasian objek merah berupa bintang merah.



Gambar 51. Klasifikasi data berdimensi-2 dengan *K-Nearest Neighbors*.

Adapun masing-masing kelebihan dan kekurangan KNN dalam *klasifikasi* dirangkum dalam Tabel 5 berikut (MLNerds, How does KNN algorithm work ? What are the advantages and disadvantages of KNN?, 2019):

Kelebihan	Kekurangan
Sederhana untuk diterapkan dan mudah untuk dipahami	Kompleksitas prediksi tinggi untuk set data besar sehingga tidak cocok untuk set data besar, karena seluruh data pelatihan

	diproses untuk setiap prediksi
Dapat mempelajari batasan keputusan non-linier saat digunakan untuk klasifikasi dan regresi. Dapat muncul dengan batas keputusan yang sangat fleksibel dengan menyesuaikan nilai k	Kompleksitas prediksi yang lebih tinggi dengan dimensi yang lebih tinggi. Pada dasarnya kompleksitas prediksi dalam <i>supervised learning</i> menjadi lebih tinggi untuk data dimensi yang lebih tinggi
Tidak ada waktu pelatihan untuk klasifikasi/regresi. Algoritma KNN tidak memiliki langkah pelatihan eksplisit dan semua pekerjaan terjadi selama prediksi	Karena KNN mengharapkan titik objek menjadi dekat di semua dimensi, KNN mungkin tidak mempertimbangkan titik yang benar-benar dekat dalam beberapa dimensi, meskipun lebih jauh dalam beberapa dimensi
Terus berkembang dengan data baru. Data baru dapat terus ditambahkan ke dalam kumpulan data, prediksi disesuaikan tanpa harus melatih kembali model baru	KNN sensitif terhadap outlier sehingga satu contoh label yang salah dapat mengubah batas kelas. Hal ini bisa menjadi masalah yang lebih besar untuk dimensi yang lebih besar pula
Ada banyak metrik jarak yang dapat dipilih. Beberapa metrik jarak populer yang digunakan adalah <i>Euclidean</i> , <i>Manhattan</i> , <i>Minkowski</i> , jarak <i>hamming</i> dan lain sebagainya	

Tabel 5. Kelebihan dan Kekurangan KNN Classifier.

B. Naïve Bayes

Walaupun sederhana *Naïve Bayes* tergolong dalam pengklasifikasi yang kuat. Pengklasifikasi *Bayes* adalah pengklasifikasi probabilistik berdasarkan penerapan teorema *Bayes* untuk probabilitas bersyarat. Asumsinya adalah bahwa semua fitur independen dan tidak terkait satu sama lain yang menjadi latar belakang mengapa *classifier* ini disebut "naïf". Pengklasifikasi *Bayes* dapat dilatih dengan sangat efisien karena model yang dipilih diterapkan pada setiap fitur secara independen. Terlepas dari asumsi sederhananya, *Bayes classifier* telah berhasil dalam aplikasi praktis, khususnya untuk penyaringan *spam e-mail*. Manfaat lain dari pengklasifikasi ini adalah bahwa setelah model dipelajari, tidak ada data pelatihan yang perlu disimpan sehingga hanya parameter model yang diperlukan. Pengklasifikasi dibangun dengan mengalikan probabilitas kondisional individu dari setiap fitur untuk mendapatkan probabilitas total kelas. Kemudian kelas dengan probabilitas tertinggi dipilih.

Kelebihan dan kekurangan *Naïve Bayes Classifier* dapat dilihat pada rangkumn Tabel 6 di bawah ini (Akkaya & Çolakoglu, 2019):

Kelebihan	Kekurangan
Model yang cepat dan fleksibel memberikan hasil yang sangat andal	<i>Records</i> data yang besar diperlukan untuk mencapai hasil yang baik
Bekerja dengan baik dengan set data besar	Menunjukkan kinerja yang lebih rendah dari pengklasifikasi lain sesuai dengan jenis masalah
Tidak perlu menghabiskan banyak waktu untuk pelatihan	
Memberikan kinerja perataan yang lebih baik dengan menghilangkan spesifikasi yang tidak signifikan	

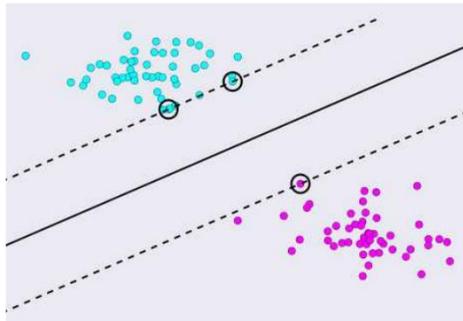
Tabel 6. Kelebihan dan Kekurangan *Naïve Bayes Classifier*.

C. Support Vector Machine (SVM)

Support Vector Machines (SVM) adalah tipe pengklasifikasi kuat yang sering memberikan hasil mutakhir untuk banyak masalah klasifikasi. Dalam bentuknya yang paling sederhana, SVM menemukan *hyperplane* pemisah linier (bidang dalam ruang dimensi yang lebih tinggi) dengan *kemungkinan* pemisahan terbaik antara dua kelas. Fungsi keputusan untuk vektor fitur x adalah sebagai berikut

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b,$$

di mana \mathbf{w} adalah *hyperplane* normal dan b adalah konstanta offset. Tingkat nol dari fungsi ini kemudian idealnya memisahkan dua kelas sehingga satu kelas memiliki nilai positif dan negatif. Parameter w dan b ditemukan dengan *memecahkan* masalah optimasi pada set pelatihan vektor fitur berlabel x_i dengan label $y_i \in \{-1,1\}$ sehingga *hyperplane* memiliki pemisahan maksimal antara dua kelas.



Gambar 52. SVM menemukan permukaan keputusan linier (*hyperplane*) yang memaksimalkan margin ke contoh pelatihan terdekat, yang disebut *support vectors*.

Berikut kelebihan *dan* kekurangan dari *Support Vector Machine* (SVM) yang dirangkum dalam Tabel 7 (K, 2019).

Kelebihan	Kekurangan
SVM bekerja relatif baik ketika ada margin pemisahan yang jelas antar kelas	Tidak cocok untuk dataset besar
Lebih efektif di ruang dimensi tinggi	SVM tidak berkinerja baik ketika kumpulan data memiliki lebih banyak kelas target <i>noise</i> yang tumpang tindih
Efektif dalam kasus di mana jumlah dimensi lebih besar dari jumlah sampel	Jika jumlah fitur untuk setiap titik data melebihi jumlah sampel data pelatihan, SVM akan berkinerja buruk
SVM relatif hemat memori	Karena <i>classifier</i> bekerja dengan meletakkan titik data di atas dan di bawah <i>hyperplane</i> maka tidak ada penjelasan probabilistik untuk klasifikasi

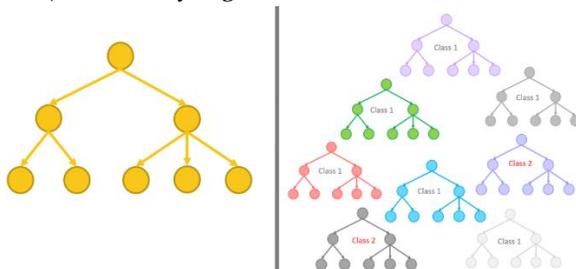
Tabel 7. Kelebihan dan Kekurangan *Support Vector Machine*.

D. Decision Tree dan Random Forest

Decision Tree adalah algoritma pembelajaran mesin yang sangat populer. Pohon Keputusan memecahkan masalah pembelajaran mesin dengan mengubah data menjadi representasi pohon. Setiap simpul internal dari representasi pohon menunjukkan atribut dan setiap simpul daun menunjukkan label kelas. Algoritma ini dapat digunakan untuk menyelesaikan masalah regresi dan klasifikasi.

Berbeda dengan sebagian besar teknik *supervised learning* yang telah kita bahas dalam bab ini, pohon keputusan (*decision tree*) melakukan urutan operasi yang lebih sederhana, seringkali hanya melihat pada elemen fitur individual sebelum memutuskan elemen mana yang akan dilihat selanjutnya

(Hastie, Tibshirani, & Friedman, 2009). Sementara pohon keputusan telah digunakan dalam pembelajaran mesin statistik selama beberapa dekade, di lain sisi *random forest* baru mulai mendapatkan daya tarik dalam visi komputer kurang lebih satu dekade yang lalu. *Decision tree*, seperti SVM, adalah pengklasifikasi diskriminatif (atau *regressor*), karena algoritma ini tidak pernah secara eksplisit membentuk model probabilistik (generatif) dari data yang diklasifikasi.



Gambar 53. *Decision Tree* (kiri) dan *Random Forest* (kanan).

Adapun kelebihan dan kekurangan dari *Decision Tree* adalah sebagai berikut (K, 2019):

Kelebihan	Kekurangan
Dibandingkan dengan algoritma lain, pohon keputusan membutuhkan lebih sedikit upaya untuk persiapan data selama <i>preprocessing</i>	Perubahan kecil pada data dapat menyebabkan perubahan besar pada struktur pohon keputusan yang menyebabkan ketidakstabilan
Tidak memerlukan normalisasi data	Terkadang perhitungan bisa jauh lebih kompleks dibandingkan dengan algoritma lain
Tidak memerlukan penskalaan data	Pohon keputusan seringkali melibatkan waktu yang lebih lama untuk melatih model
<i>Missing values</i> dalam data tidak mempengaruhi proses membangun pohon	<i>Training</i> model juga relatif mahal karena kompleksitas

keputusan sampai batas tertentu	dan waktu yang dibutuhkan lebih banyak
model sangat intuitif dan mudah dijelaskan kepada tim teknis serta pemangku kepentingan (<i>stakeholders</i>)	Algoritma <i>Decision Tree</i> tidak memadai untuk menerapkan regresi dan memprediksi nilai kontinu

Tabel 8. Kelebihan dan Kekurangan *Decision Tree*.

E. *Clustering*

Salah satu hal paling sederhana yang dapat kita lakukan pada data sampel adalah mengelompokkannya ke dalam kumpulan berdasarkan aspek kesamaan. Oleh karena itu pada bagian ini akan dikenalkan beberapa metode pengelompokan dan menunjukkan bagaimana menggunakannya untuk *clustering* citra dalam menemukan kelompok gambar yang serupa. *Clustering* dapat digunakan untuk pengenalan (*recognition*), untuk membagi kumpulan data gambar dan juga untuk organisasi dan navigasi.

1. *K-Means Clustering*

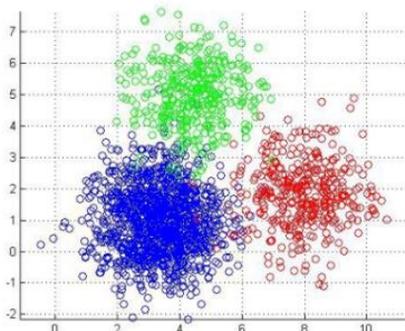
Menurut (Solem, 2012), *K-Means* adalah algoritma pengelompokan yang sangat sederhana yang mencoba untuk mempartisi data *input* dalam k cluster. *K-Means* bekerja dengan menyempurnakan estimasi awal *centroid* kelas secara iteratif sebagai berikut:

- a. Inisialisasi *centroid* $\mu_i, i = 1 \dots k$, secara acak atau dengan tebakan tertentu.
- b. Tetapkan setiap titik data ke kelas c_i dari *centroid* terdekatnya.
- c. Perbarui *centroid* sebagai rata-rata semua titik data yang ditetapkan ke kelas itu.
- d. Ulangi bagian b dan c sampai konvergen.

K-means mencoba untuk meminimalkan total varians dalam kelas

$$V = \sum_{i=1}^k \sum_{\mathbf{x}_j \in C_i} (\mathbf{x}_j - \boldsymbol{\mu}_i)^2$$

Di mana \mathbf{x}_j adalah vektor data. Algoritma di atas adalah penyempurnaan *heuristik* yang berfungsi dengan baik untuk sebagian besar kasus tetapi tidak menjamin bahwa solusi terbaik telah ditemukan. Untuk menghindari efek memilih inisialisasi *centroid* yang buruk, algoritma sering dijalankan beberapa kali dengan *centroid* inisialisasi yang berbeda. Kemudian solusi dengan varians terendah V akhirnya bisa dipilih.



Gambar 54. *K-Means Clustering.*

Berikut adalah kelebihan dan kekurangan menggunakan *cluster K-Means* (Google Developers, 2022).

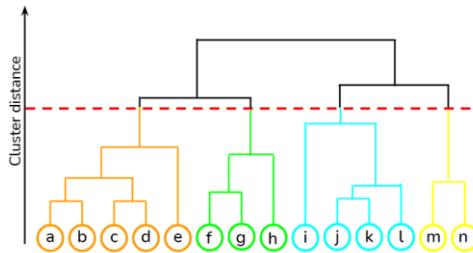
Kelebihan	Kekurangan
Relatif sederhana untuk diterapkan	Memilih k secara manual
Skala ke kumpulan data besar	Menjadi tergantung pada nilai-nilai awal
Menjamin konvergensi	Saat mengelompokkan data dengan berbagai ukuran dan kepadatan K-Means memiliki masalah <i>clustering</i>

Dapat menghangatkan posisi <i>centroids</i>	<i>Centroid</i> dapat terseret karena adanya <i>outlier</i>
Mudah beradaptasi dengan contoh baru	Ketika jumlah dimensi meningkat, ukuran kesamaan berbasis jarak konvergen ke nilai konstan antara setiap contoh yang diberikan
Generalisasi ke cluster dengan bentuk dan ukuran yang berbeda, seperti <i>cluster</i> elips	

Tabel 9. Kelebihan dan Kekurangan *K-Means Clustering*.

2. Hierarchical Clustering

Pengelompokan hierarkis adalah algoritma pengelompokan sederhana yang digunakan untuk membangun pohon kesamaan berdasarkan jarak berpasangan. Algoritma dimulai dengan mengelompokkan dua objek terdekat (berdasarkan jarak antara vektor fitur) dan membuat simpul "rata-rata" di pohon dengan dua objek sebagai anak. Kemudian pasangan terdekat berikutnya ditemukan di antara objek yang tersisa juga termasuk simpul rata-rata, dan seterusnya. Pada setiap node jarak antara dua anak akan disimpan. *Cluster* kemudian dapat diekstraksi dengan melintasi pohon ini dan berhenti di node dengan jarak yang lebih kecil beberapa ambang batas yang kemudian menentukan ukuran kelompok.



Gambar 55. Hierarchical Clustering.

Pengelompokan hierarkis memiliki beberapa manfaat. Misalnya, struktur pohon dapat digunakan untuk memvisualisasikan hubungan dan menunjukkan bagaimana

klaster terkait. Sebuah vektor fitur yang baik akan memberikan pemisahan yang bagus di pohon. Manfaat lain adalah bahwa pohon dapat digunakan kembali dengan ambang batas *cluster* yang berbeda tanpa harus menghitung ulang pohon. Kekurangannya adalah bahwa seseorang perlu memilih ambang batas jika *cluster* yang sebenarnya diperlukan. Selain itu klaster jenis ini juga tidak bekerja dengan baik pada sejumlah besar data atau kumpulan data yang sangat besar. *Hierarchical Clustering* memberikan hasil terbaik dalam beberapa kasus saja (Sultana, 2020).

Kelebihan dan kekurangan penguasaan klaster hirarkis dirangkum dalam Tabel 10 di bawah ini.

Kelebihan	Kekurangan
Tidak perlu menentukan jumlah <i>cluster</i> di awal	Tidak bekerja dengan baik pada dataset yang besar
Dapat memperoleh jumlah <i>cluster</i> yang diinginkan dengan memotong Dendogram pada level yang tepat	Semua pendekatan untuk menghitung kemiripan antar klaster memiliki kelemahannya masing-masing
Mudah untuk menentukan jumlah cluster hanya dengan melihat Dendogram	Saat keputusan dibuat untuk menggabungkan dua <i>cluster</i> , maka selanjutnya tidak dapat dibatalkan
	Sensitif terhadap <i>noise</i> dan <i>outlier</i>

Tabel 10. Kelebihan dan Kekurangan *Hierarchical Clustering*.

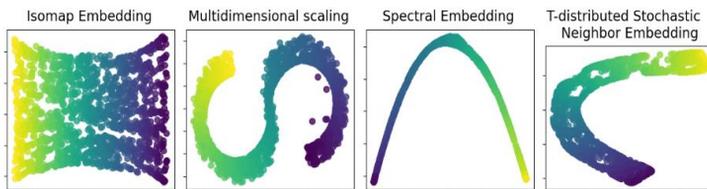
F. *Manifold Learning*

Pembelajaran *Manifold* adalah pendekatan pengurangan dimensi non-linier. Algoritma untuk tugas ini didasarkan pada gagasan bahwa dimensi banyak kumpulan data hanya tinggi secara artifisial. Dalam banyak kasus, data yang dianalisis tidak berada di subruang linier global, tetapi hidup di *manifold* berdimensi lebih rendah. Oleh karena itu *digunakan*

pembelajaran *manifold* karena sistem ini mengekstrak lipatan berdimensi lebih rendah dalam ruang berdimensi lebih tinggi, sistem ini juga dikenal sebagai teknik belajar berlipat ganda (López, Bhonsale, Peeters, & Impe, 2020).

Pembelajaran *Manifold* dapat dianggap sebagai upaya untuk menggeneralisasi kerangka kerja linier seperti PCA (*Principal Component Analysis*) agar peka terhadap struktur non-linier dalam data. Tipikal *Manifold Learning* termasuk pada pembelajaran tidak terawasi dimana ia mempelajari *struktur* data berdimensi tinggi dari data itu sendiri, tanpa menggunakan klasifikasi yang telah ditentukan sebelumnya.

Beberapa pendekatan algoritma yang telah dikembangkan untuk *Manifold Learning* meliputi penskalaan multidimensi, *Isomap*, Penyematan Linier Lokal, *Hessian Eigenmaps*, *Laplacian Eigenmaps*, penyelarasan ruang tangen lokal, Pengurangan Dimensi dengan mempelajari Pemetaan Invarian, LLE yang dimodifikasi, *t-distributed Stochastic Neighbor Embedding* (t-SNE), dan UMAP (Scikit Learn, 2022).



Gambar 56. Beberapa pendekatan algoritma *Manifold Learning*.

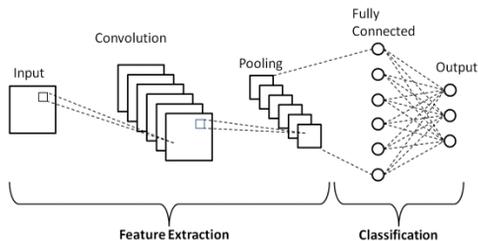
G. Convolutional Neural Network (CNN)

Pada dasarnya CNN termasuk ke dalam *Neural Network*, yaitu salah satu arsitektur *deep learning* jaringan saraf yang bisa dikatakan terinspirasi dari otak manusia. Jaringan saraf convolutional (CNN) ini pertama kali diperkenalkan sekitar tahun 1998 dan sejak itu telah berkembang dengan pesat (Kapur, 2017). CNN adalah varian dari jaringan saraf tradisional, di mana tidak semua *perceptron* terhubung satu sama lain. Di CNN, koneksi antar *perceptron* jarang terjadi. Selain itu, setiap lapisan di CNN dapat berperilaku dengan cara yang berbeda.

CNN terbukti sebagai algoritma pembelajaran terbaik untuk memahami informasi yang tersedia ke dalam citra dan dianggap sebagai model ideal untuk berbagai tugas terkait citra seperti segmentasi, klasifikasi, penandaan, deteksi dan lainnya. Sama halnya seperti otak manusia, CNN terdiri dari banyak sel komputasi yang disebut sebagai 'neuron'. Sel-sel ini melakukan operasi sederhana dan berinteraksi satu sama lain untuk membuat keputusan. Ada berbagai model CNN yang tersedia yang dapat dilatih ulang menggunakan kumpulan data khusus *domain*. Selain itu model CNN melakukan pembelajaran ujung ke ujung dan menyediakan kumpulan kelas objek beranotasi yang ada dalam kumpulan gambar yang diberikan. Studi menunjukkan bahwa, model CNN yang sudah cukup dilatih mampu mencapai kinerja tingkat manusia dalam tugas visi komputer. CNN tidak terbatas pada masalah *supervised learning* saja, tetapi *deep* CNN dapat dimanfaatkan potensinya dalam mengekstraksi informasi penting dari data skala besar yang tidak berlabel (Patel & Patel, 2020).

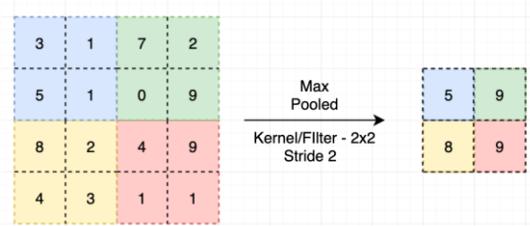
1. Lapisan Jaringan CNN

CNN adalah varian yang banyak digunakan dari Jaringan Syaraf Tiruan (JST). Nama CNN diberikan berdasarkan jenis lapisan tersembunyi yang dikandungnya, terdiri dari beberapa blok bangunan dasar. Ada lapisan *convolutional*, lapisan *pooling*, lapisan yang terhubung penuh, dan lapisan normalisasi lapisan tersembunyi yang digunakan oleh CNN. Pada Gambar 57, klasifikasi gambar umum ditampilkan, lapis demi lapis menggunakan arsitektur CNN. Ekstraksi fitur dari citra dilakukan pada dua lapisan pertama yaitu lapisan *convolutional* dan *pooling*.



Gambar 57. Lapisan arsitektur dalam CNN.

Selanjutnya adalah lapisan *Pooling* yang dikenal sebagai proses diskritisasi berbasis sampel. Tujuan utama dari *pooling* adalah untuk menurunkan *sampel* representasi *input*. Ada 2 jenis mekanisme *pooling* yang biasa digunakan yang dikenal sebagai *max* dan *min pooling*. Seperti namanya, *max pooling* memilih nilai maksimum dari wilayah yang ditentukan dan *min pooling* memilih nilai minimum dari wilayah yang ditetapkan.



Gambar 58. Contoh *max pooling* pada CNN.

Sedangkan lapisan yang terhubung penuh pada dasarnya merupakan lapisan konvolusi dengan filter ukuran 1x1. Setiap unit dalam lapisan yang terhubung penuh, terhubung erat ke semua unit lapisan sebelumnya. Dalam CNN lapisan yang terhubung penuh biasanya ditempatkan di ujung arsitektur.

2. Arsitektur CNN

Sejak tahun 1989, CNN telah banyak dibenahi dan diperbaiki. Perbaikan ini berisi penyetulan dan pengoptimalan parameter, regularisasi, kedalaman arsitektur, dan banyak lagi. Sebagian besar perbaikan yang paling *mempengaruhi* kinerja CNN yaitu penataan ulang unit pemrosesan, kedalaman jaringan dan eksploitasi spasial. Bagian ini membahas berbagai arsitektur CNN populer dan ringkasan dari semua arsitektur yang disajikan dalam Tabel 11 (Patel & Patel, 2020) (Khan, Rahmani, Shah, & Bennamoun, 2018) (Gurucharan, 2022).

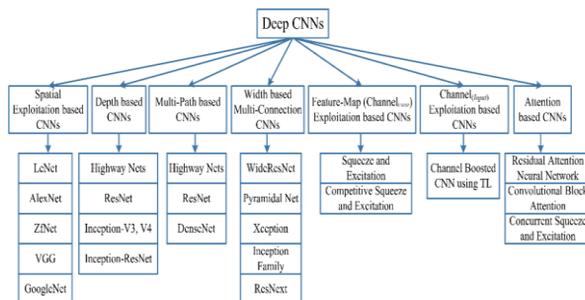
Model	Fitur	Aplikasi	Blok Bangunan CNN
LeNet-5 (1998)	Salah satu CNN paling awal dengan mengekstraksi fitur spasial, Tanh adalah fungsi aktivasi yang digunakan	Pengenalan karakter tulisan tangan	7 lapisan dengan 5 konvolusi, 2 lapisan <i>pooling</i> , 2 lapisan yang terhubung sepenuhnya
AlexNet (2012)	Jaringan <i>Deep CNN</i> pertama, meningkatkan generalisasi untuk gambar. Jumlah parameter yang cocok yang jauh lebih besar	Pengenalan karakter tulisan tangan China	8 lapisan dengan 5 konvolusi (<i>max pooling layers</i>), dan 3 lapisan yang terhubung sepenuhnya
NiN (2013)	Menggabungkan lapisan yang terhubung sepenuhnya di antara hasil lapisan konvolusi sangat	Meningkatkan diskriminabilitas model-model sebelumnya	11 lapisan dengan 9 lapisan konvolusi (<i>max pooling layers</i>), lapisan

	membantu dalam pelatihan jaringan		<i>dropout</i> , dan 2 lapisan yang terhubung sepenuhnya
VGGnet (2014)	Memiliki model yang sederhana dan penggunaan kernel konvolusi berukuran kecil	Ekstraksi fitur, deteksi dan pengenalan objek	16 lapisan konvolusi diakhiri dengan 2 lapisan yang terhubung sepenuhnya sebagai <i>output</i>
GoogLeNet (2015)	Model populer pertama yang menggunakan arsitektur yang lebih kompleks dengan beberapa cabang jaringan	<i>Pre-trained ImageNet</i> dengan mengklasifikasi citra ke dalam 1.000 kategori objek	22 lapisan dengan <i>Modul Inception</i>
ResNet-50 (2015)	Model yang sederhana namun lambat dilatih	<i>Spoofing</i> otomatis, pendeteksian, <i>speech recognition</i>	50 lapisan konvolusi diakhiri dengan 1 lapisan yang

			terhubung sepenuhnya
FractalNet (2016)	Menghindari representasi fitur yang berlebihan dengan menggunakan "path-dropout" untuk regularisasi jaringan	Tugas klasifikasi CIFAR dan <i>ImageNet</i>	50 lapisan konvolusi dan 2 lapisan <i>max pooling</i>

Tabel 11. Ringkasan Model Arsitektur CNN.

Beberapa desain CNN yang sukses yang dibangun menggunakan blok bangunan dasar telah dirangkum dalam Tabel 11. Tatanan arsitektur tersebut merupakan desain yang unggul, populer dan memiliki peringkat di ajang *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*. Masih banyak desain-desain arsitektur lainnya yang ikut berevolusi. Berikut pada Gambar 59 adalah hierarki arsitektur CNN dalam 7 kategori berbeda (Khan, Sohail, Zahoor, & Qureshi, 2020).



Gambar 59. Hierarki arsitektur CNN yang dalam 7 kategori.

3. *Tools dan Libraries*

Tingginya minat para akademisi maupun komunitas peneliti dan penggiat *deep learning* menjadi latar belakang pesatnya pengembangan *framework deep learning*. Mulai dari Universitas Berkeley California, Universitas New York, Universitas Toronto, Universitas Montreal hingga kelompok industri seperti *Google* dan *Microsoft* dan lain sebagainya ikut berpartisipasi dalam *pengembangan*. Motivasi utama untuk mengembangkan *libraries* ini adalah untuk menyediakan lingkungan pengembangan yang lebih efisien dan ramah bagi para peneliti untuk merancang dan mengimplementasikan *deep neural networks*. Tidak sedikit dari *tools* dan praktik merupakan *open source* yang sangat bermanfaat dan diminati oleh komunitas. Adapun *framework deep learning* yang dikembangkan dan didukung secara luas termasuk di antaranya, seperti *Caffe*, *TensorFlow*, *MatConvNet*, *Torch7*, *Theano*, *Keras*, *Lasagne*, *Marvin*, *Chainer*, *DeepLearning4J*, *MXNet*, dan lainnya (Khan, Rahmani, Shah, & Bennamoun, 2018).

BAB 14

Computer Vision Dataset

Untuk menyiapkan model pembelajaran mesin dan algoritma AI untuk proyek visi komputer, kita memerlukan data. Salah satu tantangan yang dihadapi oleh perusahaan yang mengerjakan proyek CV adalah mendapatkan cukup data yang tepat dan berkualitas tinggi untuk dapat melatih algoritma mereka. Pada bab ini menjelaskan pentingnya *dataset* dalam visi komputer dan bagaimana cara mendapatkannya.

A. Membuat Dataset

Di sebagian besar proyek *Machine Learning*, tahap pertama yang dilakukan adalah mengumpulkan data. Pengumpulan data dapat dilakukan dengan berbagai cara. Contohnya seperti memasang kamera di persimpangan lalu lintas, menghubungkan ke katalog digital untuk mendapatkan foto suku cadang mobil, membeli arsip citra satelit, mengumpulkan hasil *scan*, dan masih banyak lainnya. Proses pengumpulan data dapat berupa kegiatan logistik (penyimpanan kamera lalu lintas), aktivitas teknis (membangun konektor perangkat lunak ke *database* katalog), ataupun komersial (membeli arsip gambar). Kesuksesan proyek *Machine Learning* bergantung baik pada kualitas dan kuantitas data. Oleh karena itu bisa jadi, selain mengumpulkan data, kita juga perlu mengumpulkam metadata. Metadata adalah data tentang data tentang konteks di mana gambar-gambar ini diperoleh. Misalnya, cuaca pada saat foto persimpangan lalu lintas diambil, atau *file properties* dari *file* citra yang tersimpan di komputer. Berikut ini adalah penjabaran sumber citra yang dapat dikumpulkan menjadi *dataset*.

1. Foto

Foto adalah salah satu sumber data gambar yang paling umum. File ini dapat diperoleh dari media sosial dan halaman *internet*, maupun foto yang diambil oleh kamera otomatis, contohnya CCTV. Hal yang perlu kita pertimbangkan saat mengumpulkan gambar adalah

penempatan kamera dan ukuran serta resolusi gambar. Gambar harus membingkai apa pun yang kita fokuskan. Misalnya, kamera yang dipasang untuk memotret persimpangan lalu lintas harus memiliki pandangan yang tidak terhalang ke seluruh persimpangan. Atau teleskop bintang harus diposisikan sedemikian rupa agar tidak menghadap ke bangunan-bangunan. Secara intuitif, model pembelajaran mesin memiliki akurasi tinggi dengan melatihnya pada gambar beresolusi tertinggi pula. Oleh karena itu kita harus berusaha mengumpulkan data dengan resolusi tertinggi yang kita bisa. Namun, resolusi gambar yang tinggi memiliki juga beberapa kelemahan, diantaranya:

- a. Gambar yang lebih besar membutuhkan model yang lebih besar, sehingga akan memakan lebih banyak waktu untuk proses *training*.
- b. Kapasitas memory (RAM) yang terbatas bisa menjadi masalah dalam pengolahan gambar beresolusi tinggi.
- c. Gambar beresolusi tinggi, terutama yang diambil di ruang terbuka cenderung memiliki lebih banyak *noise*.
- d. Mengumpulkan citra beresolusi tinggi membutuhkan waktu yang lebih lama dan biaya yang lebih besar.
- e. Semakin tinggi kualitas semakin besar pula ukuran file, hal ini akan mempengaruhi aksesibilitas terhadap transmisi file.

Uraian di atas membuat kita mempertimbangkan penggunaan citra beresolusi tinggi menjadi sesuatu yang sia-sia. Pada dasarnya, menggunakan gambar dengan kualitas yang baik sangat disarankan bila memiliki *budget* yang besar. Efek samping seperti *noise* nantinya dapat diperbaiki menggunakan *pre-processing* gambar.

2. Pencitraan

Banyak instrumen yang dapat membuat gambar ruang 2D atau 3D. Contohnya adalah *X-Ray*, *MRI*, *radar*, spektroskop, dan masih banyak lainnya. *X-Ray* sendiri adalah

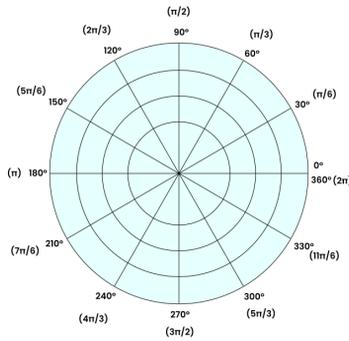
proyeksi objek 3D dan dapat diperlakukan sebagai gambar skala abu-abu (*grayscale*). Sementara foto-foto biasa berisi tiga saluran (merah, hijau, biru), gambar-gambar ini hanya memiliki satu saluran.



Gambar 60. Citra *grayscale* dari X-Ray dada manusia.

3. *Polar Grid*

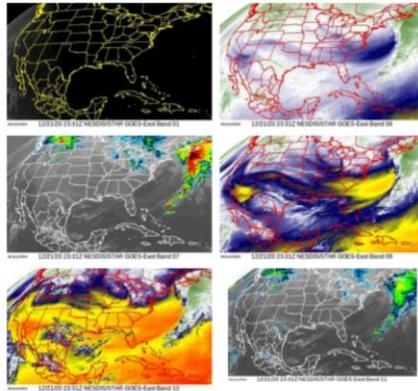
Atau bisa juga disebut sebagai jaringan kutub, adalah informasi koordinat yang dihasilkan alat seperti radar dan *ultrasound*. Kita dapat menggunakan data 2D ini sebagai gambar *input*, ataupun mengubahnya menjadi sistem koordinat *Cartesian* sebelum menggunakannya sebagai *input* untuk model pembelajaran mesin. Penggunaan data *polar grid* sebagai gambar *input* untuk model ML lebih dianjurkan, juga dengan menyertakan jarak setiap piksel dari pusat (atau ukuran piksel) sebagai masukan tambahan untuk model ML. Karena setiap piksel memiliki ukuran yang berbeda, cara termudah untuk memasukkan informasi ini adalah dengan memperlakukan ukuran piksel sebagai saluran tambahan. Dengan cara ini kita dapat memanfaatkan semua data gambar tanpa kehilangan informasi melalui transformasi koordinat.



Gambar 61. Visualisasi *polar grid*.

4. Satelit

Saat bekerja dengan citra satelit disarankan untuk bekerja dalam tampilan satelit asli atau *grid* yang dikoreksi paralaks daripada memetakan ulang gambar ke koordinat Bumi. Jika menggunakan data peta yang diproyeksikan, coba lakukan pembelajaran mesin dalam proyeksi data asli. Perlakukan gambar yang dikumpulkan dari lokasi yang sama pada waktu yang kira-kira sama, tetapi pada panjang gelombang yang berbeda (Lakshmanan, Görner, & Gillard, 2021).



Gambar 62. Gambar satelit yang diambil di waktu yang sama pada 6 saluran yang berbeda.

B. *Pre-processing Dataset*

Pra-pemrosesan data memainkan peran penting dalam pembelajaran mesin. Misalkan kita diberi gambar objek apa pun yang akan pelajari dan semua gambar disejajarkan secara vertikal. Selama fase pembelajaran kita melewati gambar objek yang sama secara horisontal, ada kemungkinan program tidak akan dapat mendeteksi objek apa itu. Hal tersebut adalah satu contoh dari banyak masalah yang dapat dihadapi dalam *training data*. Istilah yang lebih formal yang digunakan untuk menggambarkan kondisi data “bersih” adalah konsistensi data, dan invarian terhadap translasi, rotasi, dan skala. Dengan data yang seragam berarti di mana pun objek berada dalam gambar, pada orientasi apa, dan seberapa kecil atau besar objek tersebut, program akan dapat mendeteksi objek dengan benar. Adapun teknik-teknik dari *preprocessing data* yaitu sebagai berikut.

1. Mengubah warna citra menjadi *grayscale* untuk mengurangi kompleksitas perhitungan. Hal ini dilakukan karena pada beberapa objek, warna tidak diperlukan untuk mengenali dan menafsirkan sebuah gambar. Skala abu-abu bisa cukup baik untuk mengenali objek tertentu. Karena gambar berwarna mengandung lebih banyak informasi daripada gambar hitam putih, gambar tersebut dapat menambah kerumitan yang tidak perlu dan memakan lebih banyak ruang di memori. Dengan mengubahnya menjadi skala abu-abu akan mengurangi jumlah piksel yang perlu diproses.
2. Standarisasi Citra. Salah satu kendala penting yang ada di beberapa algoritma pembelajaran mesin, seperti CNN adalah kebutuhan untuk mengubah ukuran gambar dalam kumpulan data ke dimensi terpadu. Artinya gambar kita harus diproses sebelumnya dan diskalakan agar memiliki lebar dan tinggi yang identik sebelum dimasukkan ke algoritma pembelajaran.
3. Pembesaran Gambar. Teknik pra-pemrosesan umum lainnya melibatkan penambahan kumpulan data yang ada. Penskalaan, rotasi, dan transformasi *affine* lainnya adalah tipikal. Hal ini dilakukan untuk memperbesar kumpulan

data Anda dan mengekspos jaringan saraf (*neural network*) ke berbagai variasi gambar yang dimiliki. Dengan demikian model dapat mengenali objek ketika mereka muncul dalam bentuk apa pun.

4. Teknik lainnya yang dapat digunakan untuk menyiapkan gambar untuk melatih model pembelajaran mesin seperti menghapus warna latar belakang dari gambar untuk mengurangi noise. Atau kebutuhan lainnya untuk mencerahkan atau menggelapkan gambar. Singkatnya, penyesuaian apa pun yang perlu diterapkan ke kumpulan data digolongkan sebagai pra-pemrosesan data.

C. *Splitting Dataset*

Dataset dari gambar dan label pada umumnya dibagi menjadi tiga bagian, untuk pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*). Dalam beberapa proyek sederhana untuk mencari akurasi misalnya cukup membagi data menjadi data latih dan uji. Proporsi *splitting* tidak ditentukan secara spesifik, tetapi pada umumnya proporsi yang digunakan yaitu 80:10:10. Rasio yang tepat bergantung pada data, tetapi rasio 70:20:10 cukup optimal untuk kumpulan data kecil. *Dataset training* adalah kumpulan contoh yang disajikan ke model sehingga jumlahnya harus paling banyak. Hilangnya *dataset* pelatihan pada akhir pelatihan biasa terjadi. Oleh karena itu, kita harus menggunakan kumpulan data contoh yang belum ditunjukkan ke model selama proses pelatihannya. Itulah tujuan dari *dataset* validasi.

Jika kita hanya melatih satu model dalam satu kali pelatihan, kita hanya membutuhkan *dataset* pelatihan dan validasi (umumnya dengan pembagian 80:20). Namun, kemungkinan besar kita akan mencoba kembali pelatihan dengan rangkaian *hyperparameter* yang berbeda dengan tujuan seperti ingin mengubah *learning rate*, mengurangi *dropout*, ataupun menambahkan beberapa *layer* ke model. Semakin banyak *hyperparameter* yang dioptimalkan terhadap *dataset* validasi, semakin banyak pula keterampilan model pada *dataset*

validasi yang dimasukkan ke dalam struktur model itu sendiri. Dengan demikian, dataset validasi tidak lagi menjadi cerminan tentang bagaimana model akan tampil ketika diberikan data baru yang masih bersih.

Memisahkan kumpulan data di awal setiap *training* bukanlah ide yang baik. Jika kita melakukan ini, setiap eksperimen akan memiliki set data pelatihan dan validasi yang berbeda, sehingga melenceng dari tujuan untuk mempertahankan set data *testing* yang benar-benar independen. Sebagai gantinya, kita harus membagi satu kali, dan kemudian melanjutkan menggunakan set data pelatihan dan validasi yang sama untuk semua eksperimen penyetelan *hyperparameter* kita. Oleh karena itu, kita harus menyimpan file pelatihan, validasi, dan pengujian dan menggunakannya secara konsisten sepanjang siklus hidup model.



Gambar 63. Berbagai proporsi *splitting dataset*.

BAB 15

Deteksi dan Pencocokan Fitur

Deteksi objek adalah salah satu topik yang paling terkenal dan banyak diteliti di bidang *Computer Vision* dan *Machine Learning*. Topik ini telah menarik banyak peneliti yang bekerja di berbagai bidang visi komputer, robotika, pencitraan medis, teknik mesin, dan telekomunikasi. Deteksi objek adalah metodologi dalam Pembelajaran Mesin yang berfokus pada pelokalan dan pengenalan objek yang berbeda dalam gambar dan video. Setiap objek memiliki fitur tersendiri yang membantu membedakannya dalam bingkai foto atau video, selain deteksi fitur mesin juga melakukan pencocokan fitur atau *features matching*. Dalam bab ini dirangkum berbagai metodologi dalam deteksi dan pencocokan fitur.

A. Deteksi Tepi

Deteksi tepi adalah teknik pemrosesan gambar yang digunakan untuk mengidentifikasi titik-titik dalam gambar digital dengan diskontinuitas untuk mengamati perubahan tajam dalam kecerahan gambar. Titik-titik di mana kecerahan gambar sangat bervariasi ini disebut tepi (atau batas) gambar. Ada berbagai metode dalam *edge detection*, dan berikut ini adalah beberapa metode yang paling umum digunakan:

1. *Prewitt Edge Detection*

Metode ini merupakan detektor tepi yang umum digunakan terutama untuk mendeteksi tepi horizontal dan vertikal pada citra. Adapun langkah-langkah dalam operator ini yaitu

- a. Membaca data gambar
- b. Mengubah ke citra *grayscale*
- c. Mengubah ke dalam bentuk format ganda
- d. Mendefinisikan *mask* atau *filter*
- e. Mendeteksi tepi sepanjang *X-axis*
- f. Mendeteksi tepi sepanjang *Y-axis*
- g. Menggabungkan tepi yang sudah dideteksi X dan Y
- h. Menampilkan semua gambar

2. Sobel Edge Detection

Metode ini menggunakan *filter* yang lebih menekankan pada bagian tengah *filter*. Ini adalah salah satu detektor tepi yang paling umum digunakan dan membantu mengurangi kebisingan dan memberikan perbedaan, memberikan respons tepi secara bersamaan. Perbedaan antara *Sobel* dan *Prewitt* adalah bahwa di *operator Sobel* koefisien *mask* dapat disesuaikan sesuai dengan kebutuhan asalkan mengikuti semua sifat *mask* turunan.

3. Laplacian Edge Detection

Filter Laplacian adalah *filter* turunan yang digunakan untuk mengekstrak tepi vertikal dan horizontal dari suatu gambar. Berbeda dengan detektor tepi *Sobel*, detektor tepi *Laplacian* hanya menggunakan satu kernel yang menghitung turunan orde kedua dalam satu lintasan. Tahapan dalam metode ini yaitu

- a. Menentukan fungsi untuk *filter*
- b. Membuat *mask*
- c. Menentukan fungsi untuk mengulangi *filter* di atas *mask*
- d. Membuat fungsi untuk memeriksa nilai-nilai nol
- e. Membuat fungsi untuk merangkup keseluruhan proses

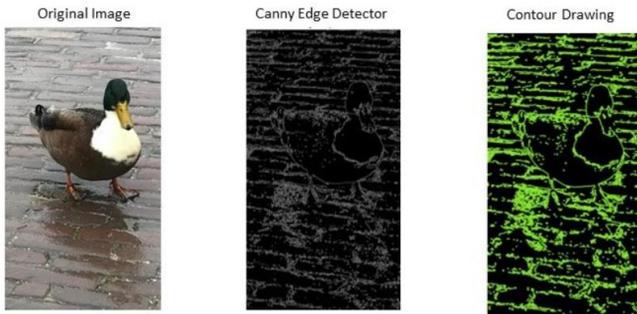
4. Canny Edge Detection

Metode ini adalah yang paling umum digunakan karena sangat efektif dan kompleks dibandingkan dengan banyak metode lainnya. *Canny edge detector* adalah algoritma multi-tahap yang digunakan untuk mendeteksi atau mengidentifikasi berbagai tepi. Tahapan yang dilalui yakni:

- a. Mengubah citra menjadi *grayscale*.
- b. Mengurangi *noise*.
- c. Menghitung gradient.
- d. *Non-maximum suppression*.
- e. *Thresholding* ganda.
- f. *Hysteresis edge tracking*.

B. Deteksi Kontur

Kontur adalah kurva tertutup yang menghubungkan titik-titik dalam gambar dengan warna atau intensitas yang sama. Karena didasarkan pada intensitas piksel, kinerjanya lebih baik dengan gambar biner, terutama dengan latar belakang hitam dan objek putih. Kontur mendeteksi sosok tertutup tidak seperti deteksi tepi di mana hanya batas objek yang digariskan. Oleh karena itu, kontur dapat divisualisasikan sebagai hubungan tepi untuk membentuk suatu objek. Kontur dapat diterapkan pada beberapa deteksi objek sesuai dengan ukuran dan intensitas, memotong objek, dan sebagainya. Kontur banyak digunakan dalam topologi serta aplikasi medis (Computation, 2021).



Gambar 64. Perbandingan deteksi tepi *Canny* dan deteksi kontur pada gambar yang sama.

C. Teknik Segmentasi

Segmentasi gambar adalah proses memecah gambar menjadi wilayah yang lebih kecil yang secara individual menyimpan informasi yang berarti dan membantu kita memahami keseluruhan isi gambar. Sebagai contoh, mari kita lihat Gambar 65 berikut. Di sebelah kiri dapat dilihat gambar asli dan di sebelah kanan ada gambar tersegmentasi yang sesuai. Seperti yang kita lihat, algoritma berhasil mengelompokkan bagian-bagian gambar yang serupa. Seperti seluruh *background* dengan semak-semak dikelompokkan menjadi warna hijau tua. Rumput kuning dan rumput di depan binatang itu diwarnai

menjadi satu. Tidak semua teknik akan bekerja di semua situasi, beberapa teknik segmentasi akan dibahas pada bagian ini.



Gambar 65. Perbandingan gambar asli dan hasil segmentasi.

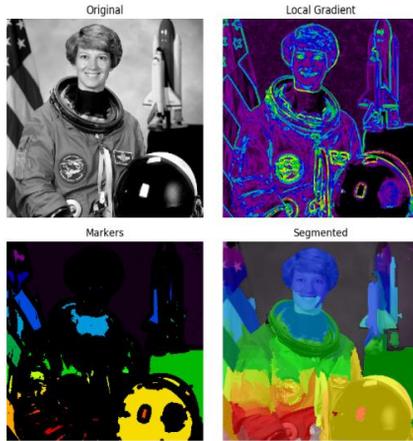
1. *Watershed Algorithm*

Algoritma *Watershed* digunakan untuk segmentasi pada beberapa citra yang kompleks karena jika kita menerapkan *thresholding* dan deteksi kontur yang sederhana maka tidak akan dapat memberikan hasil yang tepat. Algoritma DAS (Daerah Aliran Sungai) didasarkan pada penggalian latar belakang dan latar depan tertentu dan kemudian menggunakan penanda akan membuat DAS berjalan dan mendeteksi batas yang tepat. Algoritma ini umumnya membantu dalam mendeteksi objek yang bersentuhan dan tumpang tindih dalam gambar.

Untuk penanda, dapat ditentukan pengguna seperti mengklik secara manual dan mendapatkan koordinat untuk penanda dan juga menggunakan beberapa algoritma yang ditentukan seperti ambang batas atau operasi morfologi apapun.

Langkah-langkah algoritma adalah sebagai berikut:

- a. Baca gambar yang ingin disegmentasikan.
- b. Ubah menjadi skala abu-abu (hanya jika belum dalam skala abu-abu).
- c. Ubah nilai piksel gambar menjadi *unsigned int*.
- d. Hitung gradien lokal dari gambar.
- e. Terapkan algoritma DAS.



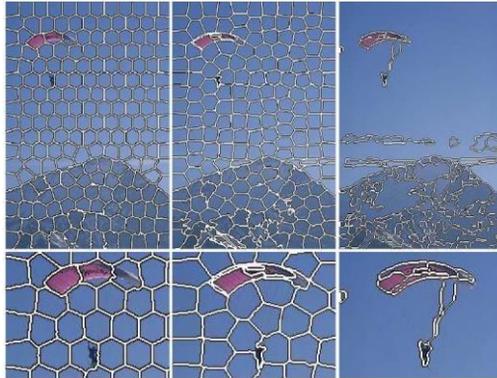
Gambar 66. Hasil tahapan Algoritma *Watershed*.

2. *Superpixels*

Superpiksel mengelompokkan piksel yang serupa dalam warna dan properti tingkat rendah lainnya. Dalam hal ini, superpiksel mengatasi dua masalah yang melekat pada pemrosesan gambar digital: pertama, piksel hanyalah hasil diskritisasi dan kedua, tingginya jumlah piksel dalam gambar besar mencegah algoritma menjadi layak secara komputasi.

Dalam visi komputer, superpiksel memberikan representasi data gambar yang ringkas dengan mengelompokkan piksel yang mirip secara persepsi. Sebagai cara untuk secara efektif mengurangi jumlah gambar primitif untuk pemrosesan selanjutnya, piksel super telah diadopsi secara luas dalam masalah penglihatan seperti deteksi *saliency*, deteksi objek, pelacakan, dan segmentasi semantik. Namun, superpiksel belum diadopsi secara luas di DNN untuk pengurangan dimensi. Salah satu alasan utama adalah bahwa, operasi konvolusi standar dalam jaringan saraf convolutional (CNNs) didefinisikan pada *grid* gambar biasa. Sementara beberapa upaya telah dilakukan untuk memodifikasi arsitektur dalam untuk menggabungkan superpiksel, melakukan konvolusi pada kisi superpiksel

yang tidak teratur tetap menjadi tantangan (Yang, Sun, Jin, & Zhou, 2020) (Santos & Oliveira, 2018).



Gambar 67. Contoh penerapan segmentasi *superpixels*.

3. *Graph cut* yang dinormalisasi

Merupakan salah satu teknik segmentasi gambar yang paling populer saat ini. Penjelasan paling sederhana dari teknik pemotongan grafik adalah bahwa setiap piksel dalam gambar diperlakukan sebagai simpul. Sebuah graf adalah himpunan simpul V dan tepi E yang menghubungkan berbagai node; $G = \{V, E\}$. Graf berbobot adalah graf yang bobotnya terkait dengan setiap tepi sedangkan graf terhubung adalah graf di mana setiap pasangan node terhubung. Dalam pengelompokan, graf berbobot dibagi menjadi himpunan-himpunan (kelompok) yang terpisah-pisah di mana dengan beberapa ukuran kesamaan dalam suatu kelompok tinggi dan di seluruh kelompok itu rendah. Pemotongan graf adalah teknik pengelompokan di mana tingkat ketidakmiripan antara dua kelompok ini dihitung sebagai berat total tepi yang dihilangkan antara 2 bagian.

BAB 16

Object Detection & Recognition

Deteksi dan pengenalan objek adalah tugas visi komputer penting yang digunakan untuk mendeteksi contoh objek visual dari kelas tertentu misalnya, manusia, hewan, mobil, atau bangunan dalam gambar digital seperti foto atau bingkai video. Tujuan dari deteksi objek adalah untuk mengembangkan model komputasi yang menyediakan informasi paling mendasar yang dibutuhkan oleh aplikasi visi komputer. Pengenalan objek adalah hal yang penting karena proses ini membentuk dasar dari banyak tugas visi komputer lainnya, misalnya, segmentasi instan, teks gambar, pelacakan objek, dan banyak lagi. Contoh sederhana pengenalan objek seperti, pengenalan hewan, deteksi kendaraan, penghitungan jumlah orang, deteksi wajah, deteksi teks, hingga pengenalan plat nomor. Pada bab ini akan dibahas bagaimana visi komputer dapat mengenali objek dan tahapan proses yang dilaluinya.

A. Klasifikasi Citra

Ada banyak metode untuk klasifikasi dan pengenalan citra, seperti metode pengenalan statistik, metode pengenalan struktur, metode jaringan saraf (*neural network*), metode pengenalan himpunan *fuzzy*, SVM, dan metode pencocokan *template*. Berbagai metode yang berbeda ini memiliki kelebihan dan kekurangannya masing-masing. Hal penting dari pemrosesan citra dan *robot vision* adalah ekstraksi fitur dan teknologi klasifikasi yang dapat memproses informasi visual secara efisien, memperoleh informasi yang dibutuhkan dalam kehidupan dan produksi industri. Klasifikasi gambar adalah tugas mengkategorikan dan menetapkan label ke kelompok piksel atau vektor dalam gambar tergantung pada aturan tertentu.

Teknik klasifikasi citra terutama dibagi menjadi dua kategori yaitu *supervised* dan *unsupervised*. Teknik klasifikasi tanpa pengawasan adalah metode yang sepenuhnya otomatis yang tidak memanfaatkan data pelatihan. Artinya algoritma

machine learning digunakan untuk menganalisis dan mengelompokkan kumpulan data yang tidak berlabel dengan menemukan pola atau grup data tersembunyi tanpa perlu campur tangan manusia. Dengan bantuan algoritma yang sesuai, karakterisasi tertentu dari suatu gambar dikenali secara sistematis selama tahap pemrosesan gambar. Pengenalan pola dan pengelompokan gambar adalah dua metode klasifikasi gambar yang paling umum digunakan dalam tahap klasifikasi. Dua algoritma populer yang digunakan untuk klasifikasi gambar tanpa pengawasan adalah K-mean dan ISODATA (Boesch, 2022).

Sedangkan metode klasifikasi citra yang diawasi menggunakan sampel referensi yang diklasifikasikan sebelumnya untuk melatih pengklasifikasi dan selanjutnya mengklasifikasikan data baru yang tidak diketahui. Oleh karena itu, teknik klasifikasi terawasi adalah proses pemilihan sampel data pelatihan secara visual di dalam citra dan mengalokasikannya ke kategori yang telah dipilih sebelumnya, termasuk vegetasi, jalanan, sumber daya air, dan bangunan. Hal ini dilakukan untuk membuat ukuran statistik yang akan diterapkan pada gambar secara keseluruhan.

Klasifikasi visi komputer bekerja dengan cara sebuah komputer menganalisis gambar dalam bentuk piksel. Gambar diproyeksikan sebagai *array* matriks dengan ukuran matriks bergantung pada resolusi gambar. Sederhananya, klasifikasi gambar dalam tampilan komputer adalah analisis data statistik ini menggunakan algoritma. Dalam pemrosesan citra digital, klasifikasi citra dilakukan dengan mengelompokkan piksel secara otomatis ke dalam kategori tertentu, yang disebut "kelas".

Klasifikasi pengenalan citra secara umum dibagi menjadi beberapa langkah dimulai dengan melakukan *input* citra uji, melakukan pra-pemrosesan citra, melakukan segmentasi citra, mengekstraksi fitur citra, melakukan pengenalan citra, dan mengeluarkan hasil yang sudah diklasifikasikan. Metode yang umum digunakan untuk memproses citra termasuk *Deep CNN*

(DCNN), *Bag of Features* (BOF), *K-Means*, algoritma kopling SVM (KM-SVM), dan lainnya (Liu, Bao, & Pang, 2019).

Dibandingkan dengan pendekatan *computer vision* tradisional dalam pemrosesan gambar yang digunakan di masa lalu, *deep learning* hanya membutuhkan pengetahuan teknik alat pembelajaran mesin. Tidak perlu keahlian di bidang visi mesin tertentu untuk membuat sebuah fitur. Bagaimanapun, *deep learning* memerlukan pelabelan data manual untuk menginterpretasikan sampel yang baik dan buruk, yang dikenal sebagai anotasi gambar. *Deep CNN* (DNN) dan CNN telah menunjukkan hasil yang sangat baik dalam tugas visi komputer, terutama dalam klasifikasi gambar. *Convolutional Neural Network* (CNN, atau ConvNet) adalah jenis khusus jaringan saraf multi-layer yang terinspirasi oleh mekanisme sistem optik dan saraf manusia. Hanya ada beberapa tahap pra-pemrosesan yang diperlukan saat menggunakan CNN dalam klasifikasi. Karena CNN mengembangkan dan mengadaptasi filter gambar secara mandiri (Meißner, 2018).

Penerapan klasifikasi gambar menjadi penting di berbagai industri, penggunaannya populer di bidang perawatan medis, manufaktur industri, *smart city*, asuransi, dan bahkan eksplorasi ruang angkasa. Salah satu alasan pesatnya penerapan adalah jumlah data visual yang terus bertambah dan kemajuan pesat dalam teknologi komputasi canggih. Ada banyak contoh aplikasi klasifikasi citra. Penggunaan yang cukup populer meliputi:

1. Inspeksi otomatis dan kontrol kualitas

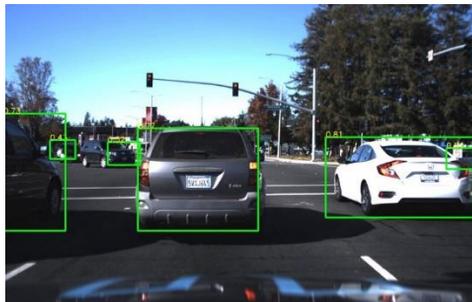
Seiring waktu, manufaktur menjadi lebih kompleks dan efisien, dan menjadi jelas bahwa produk tidak dapat sepenuhnya sesuai dengan sketsa karena selalu ada penyimpangan tergantung pada rincian pemeriksaan. Dalam upaya untuk mengatasi masalah ini, batas toleransi menjadi lebih umum, dan inspektur manusia digantikan oleh sistem visi komputer. Dengan meningkatnya persaingan, tekanan untuk proses yang lebih efisien, lebih sedikit limbah, biaya lebih rendah, dan kualitas produk yang lebih tinggi

meningkat di semua sektor. Hal ini menyebabkan munculnya manajemen kualitas, otomatisasi, hingga paradigma Industri 4.0 memanfaatkan transformasi digital dari industri manufaktur. Persyaratan ini membebani kemampuan operator manusia untuk mengikuti tugas inspeksi dan kemungkinan kesalahan manusia meningkat. (Babic, Farahani, & Wuest, 2021).

2. Pengenalan objek pada mobil tanpa pengemudi

Adanya klasifikasi dan pengenalan objek pada mobil *self driving* sehingga dapat diketahui di mana kendaraan lain berada di jalan dan mampu mengantisipasi ke mana mobil bergerak. Selain itu dapat pula menentukan seberapa jauh posisinya, ke arah mana mobil bergerak, dan seberapa laju pergerakannya. Sama halnya seperti yang kita lakukan dengan mata kita sendiri saat kita mengemudi secara manual. Deteksi dan pelacakan objek adalah konsep utama dalam visi komputer tingkat lanjut (Sawhney, 2021).

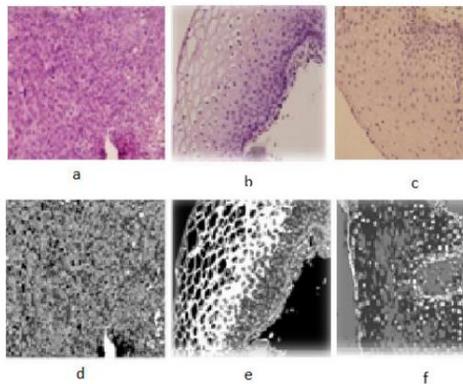
Melalui algoritma yang disebut YOLO (You Only Look Once) dilakukan deteksi objek waktu nyata (*realtime*), sempurna untuk kendaraan kendali otomatis. YOLO sangat cepat, menggunakan 24 lapisan konvolusi, dan dapat memproses hingga 155 *frame* per detik. Hal ini membuatnya mudah diimplementasikan ke dalam mobil *self-driving* (Wen, 2019).



Gambar 68. Contoh penerapan YOLO pada mobil *self driving*.

3. Deteksi sel kanker dalam *slide* patologi

Perkembangan klasifikasi penyakit dalam sudah berkembang sangat pesat. Terutama dalam penanganan penyakit kanker yang kasusnya kian bertambah sepanjang tahun, peneliti *machine learning* dan medis pun berkolaborasi untuk melatih pembelajaran mesin untuk mediagnosa kanker berdasarkan citra. Kanker yang dapat diklasifikasi begitu banyak jenisnya mulai dari organ dalam seperti serviks hingga organ luar seperti kulit. Dengan dilakukannya klasifikasi dan deteksi dini pada penyakit kanker ditujukan untuk dapat dilakukannya penanganan secara dini (Rahmadwati, 2014).



Gambar 69. Citra dari (a) serviks normal, (b) pra kanker dan (c) ganas. Gambar (d), (e) dan (f) adalah citra hasil yang telah disegmentasi menggunakan *K-means clustering*.

4. Pengenalan wajah untuk keamanan

Banyak orang yang sudah akrab dengan teknologi pengenalan wajah melalui *Face ID* yang digunakan untuk membuka kunci *smartphone* sebagai contohnya. Biasanya, pengenalan wajah tidak bergantung pada *database* foto yang sangat besar untuk menentukan identitas individu melainkan hanya mengidentifikasi dan mengenali satu orang sebagai pemilik tunggal perangkat, sambil membatasi akses ke orang lain.

Selain membuka kunci ponsel, pengenalan wajah bekerja dengan mencocokkan wajah orang yang berjalan melewati kamera khusus, dengan gambar orang dalam daftar pantauan. Daftar pantauan dapat berisi gambar siapa saja, termasuk orang yang tidak dicurigai melakukan kesalahan, dan gambar tersebut dapat berasal dari mana saja. Sederhananya pengenalan wajah bekerja dengan mengidentifikasi wajah yang spesifik untuk dikenali. Penerapan ini banyak sekali manfaatnya seperti dalam penegakan hukum, kontrol pengawasan di bandara, perbankan, layanan kesehatan, pelacakan kehadiran pekerja, menemukan orang yang hilang, pengenalan supir untuk mobil-mobil perusahaan dan masih banyak lainnya (AO Kaspersky Lab, 2022).



Gambar 70. Penerapan *face recognition* di bandara.

5. Pemantauan lalu lintas dan deteksi kemacetan

Deteksi dan manajemen kemacetan lalu lintas jalan telah menjadi tantangan dan beberapa negara telah memunculkan konsep dan ide baru untuk mendeteksi kemacetan. Saat ini, jumlah kendaraan yang meningkat hingga tingkat yang mengkhawatirkan sedangkan keterbatasan perbaikan infrastruktur jalan belum mampu mengatasi masalah tersebut. Selama dekade ini, banyak konsep dan teknologi inovatif telah dikembangkan sehubungan dengan pengumpulan data lalu lintas waktu nyata dan menggunakannya untuk memperoleh pengetahuan tentang berbagai aspek arus lalu lintas. Baik informasi kuantitatif maupun kualitatif dapat dikumpulkan

dengan menggunakan teknik-teknik tersebut. Data kuantitatif meliputi kecepatan lajur rata-rata, panjang antrian, waktu perjalanan, waktu tunggu, dan lainnya. Sedangkan data kualitatif antara lain mendeteksi dan mengkarakterisasi kemacetan di berbagai lokasi kejadian (Mandal, et al., 2011).



Gambar 71. *Traffic management* otomatis.

6. Pemetaan penggunaan lahan

Machine learning juga telah berkembang ke titik yang memungkinkan penggunaan visi komputer dan pembelajaran mendalam dalam analisis dan klasifikasi gambar permukaan lahan. Ilmuwan riset lingkungan menggunakan citra satelit dan udara untuk memahami pola penggunaan lahan, khususnya untuk memahami dampak perubahan iklim dan ekspansi populasi manusia terhadap sumber daya alam di Bumi. Menyaring citra menjadi data yang dapat ditindaklanjuti yakni dalam bentuk peta tutupan lahan saat ini membutuhkan anotasi manual yang ekstensif. Pekerjaan ini menggunakan bantuan *computer vision* untuk mempercepat proses ini, memungkinkan ilmuwan lingkungan dan analis geospasial menghabiskan lebih sedikit waktu menggambar poligon dan lebih banyak waktu untuk perencanaan konservasi (esri, 2020).



Gambar 72. Perubahan perkebunan pisang di Queensland utara dari tahun 2015 hingga 2018.

B. Pencarian Gambar

Surfing dan *browsing* di internet untuk mencari informasi tidak hanya sebatas memasukkan teks berupa kata kunci. Kini kita juga dapat mencari informasi melalui inputan gambar untuk mencari tahu informasi yang ada di dalam gambar tersebut. Pencarian gambar atau biasa disebut *image search* adalah teknologi yang menggunakan gambar dan bukan kata-kata untuk menetapkan kriteria pencarian dan menemukan hasil yang sesuai. Selain *image search* ada istilah lainnya yaitu *visual search*.

Pencarian visual berada di bawah payung apa yang dikenal sebagai "pencarian sensorik", yang mencakup pencarian melalui teks, suara, dan penglihatan. Meskipun pencarian visual dan gambar didasarkan pada objek yang sama yaitu citra, perbedaan penting terletak pada inputan yang digunakan. Untuk melakukan pencarian gambar dilakukan *input* kata kunci, sedangkan dengan pencarian visual pengguna menggunakan gambar untuk melakukan pencarian (Mohanadasan, 2020). Sebagai contoh, *image search* adalah saat seseorang ingin membeli "baju batik pekalongan", maka cukup memasukkan kata kunci model, warna, motif yang diinginkan. Sedangkan *visual search* dengan memilih gambar baju batik Pekalongan yang disukai dan mengunggahnya ke dalam *search engine* maka dengan kombinasi warna atau desain batik yang serupa dan

perangkat lunak pencarian visual akan menyaring hasil pencarian yang sesuai.

Computer vision dan pemrosesan informasi multimedia telah membuat kemajuan ekstrim dalam dekade terakhir dan banyak tugas di beberapa domain dapat dilakukan dengan tingkat akurasi yang sama seperti jika dilakukan oleh manusia, atau lebih baik. Hal ini adalah bukti nyata dari manfaat tersedianya sejumlah besar data yang digunakan untuk pelatihan. Mengingat bahwa kita dapat secara otomatis memproses beberapa bentuk informasi visual lebih baik daripada manusia (Smeaton, 2021).

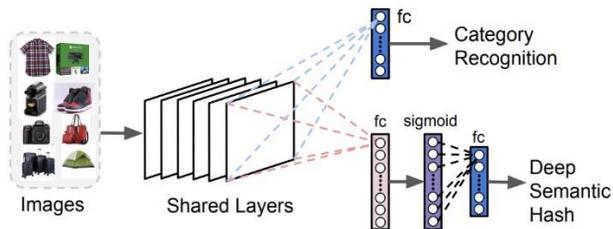
Penggunaan pencarian visual terbaik dan paling produktif adalah dalam industri *e-commerce* di mana pelanggan dapat mencari produk atau item pakaian favorit mereka menggunakan pencarian visual. Pelanggan dapat menemukan produk serupa yang cocok dengan bentuk, warna, *style* dan pola. Teknologi pencarian visual menampilkan produk sesuai relevansi yang paling populer di mata pelanggan. Beberapa pencari visual yang cukup populer contohnya adalah *Google Lens*, *Pinterest Lens*, *Bing Visual Search*, *Snapchat Camera Search*, *Amazon StyleSnap*, dan *eBay Visual Search*



Gambar 73. *Visual search* dengan *eBay* menampilkan gambar yang cocok dengan gambar berkotak merah.

Pada Gambar 73 di atas adalah contoh bagaimana *visual search* dimanfaatkan dalam laman belanja *online*. Saat pembeli

mengunggah gambar kueri untuk pencarian visual, jaringan saraf *eBay* melakukan dua proses penting yaitu pengenalan kategori dengan memprediksi kategori produk apakah yang diterima, Setelah itu proses kedua yaitu melihat kesamaan atau kemiripan. *Neural network* menghasilkan *hash* semantik untuk gambar. Dengan menggunakan nomor untuk menghitung skor kesamaan terhadap semua *hash* produk dalam kategori yang diprediksi. Hinga akhirnya produk dengan nilai kemiripan terbesar akan ditampilkan dari hasil kueri (Williams, 2018).



Gambar 73. Skema *training eBay neural network*.

C. Face Recognition

1. Permasalahan pada *face recognition*

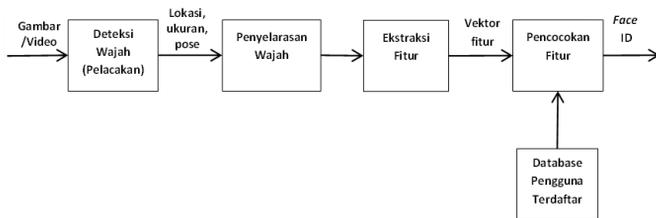
Seperti yang sudah dibahas sebelumnya pada subbab Klasifikasi Citra, pengenalan wajah adalah masalah mengidentifikasi dan memverifikasi orang dalam foto dengan wajah mereka. Mengenali wajah seseorang adalah hal yang cukup menantang karena objek yang diidentifikasi oleh mesin memiliki inkonsistensi. Misalnya saat berada di bawah cahaya yang berbeda-beda, atau ketika wajah berubah karena usia atau terhalang oleh aksesoris dan rambut wajah (Brownlee, 2019). Selain itu pose juga dapat menyebabkan variasi dramatis dalam gambar wajah. Pada dasarnya, ini disebabkan oleh struktur geometris 3D yang kompleks dari kepala manusia. Rotasi kepala yang kaku menghasilkan oklusi diri sehingga beberapa sisi wajah tidak akan terlihat. Pada saat yang sama, bentuk dan posisi bagian yang terlihat dari citra wajah juga bervariasi secara nonlinier dari pose ke

pose. Akibatnya, keragaman penampilan yang disebabkan oleh pose biasanya lebih besar daripada yang disebabkan oleh identitas wajah itu sendiri. Akibatnya algoritma pengenalan wajah umum selalu gagal ketika berhadapan dengan gambar pose yang berbeda (Jiang, Hadid, Pang, Granger, & Feng, 2019).

2. Deteksi wajah otomatis

Saat mesin diberikan gambar statis atau video dari suatu adegan, maka proses yang dilakukan yaitu mengidentifikasi atau memverifikasi satu orang atau lebih yang ada di dalam adegan menggunakan *database* wajah yang sudah sebelumnya disimpan. Pada Gambar 74 dapat dilihat alur proses pengenalan wajah sering digambarkan sebagai suatu proses yang pertama melibatkan empat langkah (Brownlee, 2019), yaitu sebagai berikut:

- a. Deteksi wajah dengan menemukan satu atau lebih wajah dalam gambar dan tandai dengan kotak pembatas.
- b. Pnyelarasan wajah yaitu menormalkan wajah agar konsisten dengan *database*, seperti geometri dan fotometrik.
- c. Ekstraksi fitur, digunakan untuk tugas pengenalan (*recognition*).
- d. Pengenalan wajah. Dengan melakukan pencocokan wajah dengan satu atau lebih wajah yang dikenal dalam *database* yang telah disiapkan.



Gambar 74. Alur proses pengenalan wajah.

3. Teknik deteksi

Sebelum pengenalan wajah dapat diterapkan pada gambar umum, lokasi dan ukuran wajah harus ditemukan terlebih dahulu. Pada prinsipnya, algoritma pengenalan wajah dapat diterapkan pada setiap piksel dan skala, tetapi proses seperti itu memakan lebih banyak waktu dalam praktiknya. Selama empat dekade terakhir, berbagai macam algoritma deteksi wajah instan telah dikembangkan. Saat ini teknik deteksi wajah dapat diklasifikasikan menjadi 4 kategori yaitu, berbasis pengetahuan, berbasis fitur, berbasis *template*, atau berbasis penampilan (Bernstein, 2020) (Szeliski, 2022).

- a. Teknik berbasis pengetahuan atau metode berbasis aturan, menggambarkan wajah berdasarkan aturan. Tantangan dari pendekatan ini adalah sulitnya menghasilkan aturan yang terdefinisi dengan baik.
- b. Teknik berbasis fitur mencoba untuk menemukan lokasi fitur gambar yang berbeda seperti mata, hidung, dan mulut, dan kemudian memverifikasi apakah fitur ini berada dalam pengaturan geometri yang masuk akal.
- c. Pendekatan berbasis *template* seperti *Active Appearance Models* (AAMs), dapat menangani berbagai variasi pose dan ekspresi. Biasanya memerlukan inisialisasi yang baik di dekat wajah asli sehingga pendekatan ini tidak cocok sebagai detektor wajah cepat (*realtime*).
- d. Pendekatan berbasis penampilan memindai *patch* persegi panjang kecil yang tumpang tindih (berlapis) dari suatu gambar dengan banyak objek wajah saat mencari kandidat wajah yang sesuai. Untuk menangani variasi skala, gambar biasanya diubah menjadi piramida sub-oktaf dan pemindaian terpisah dilakukan pada setiap tingkat. Sebagian besar pendekatan berbasis penampilan sangat bergantung pada pengklasifikasi pelatihan menggunakan set wajah berlabel dan wajah non-label.



Gambar 74. Hasil deteksi wajah dengan *manifold learning* (Rowley, Baluja, & Kanade, 1998).

4. Metode deteksi

Selama 20 tahun terakhir, banyak algoritma pengenalan wajah yang telah diusulkan. Di antara beberapa teknik, metode sub-ruang atau disebut metode berbasis penampilan memberikan kontribusi yang paling signifikan terhadap perkembangan teknik deteksi dan pengenalan wajah. Berdasarkan jenis transformasi citra yang digunakan, metode subruang dapat dibagi lagi menjadi metode linier dan nonlinier. Semua teknik pengenalan wajah yang ada dapat diklasifikasikan menjadi lima jenis berdasarkan cara mereka mengidentifikasi wajah (Datta, Datta, & Banerjee, 2016).

a. Pengenalan wajah berbasis sub-ruang

Secara umum, metode sub-ruang menggunakan serangkaian gambar wajah pelatihan untuk menghitung ruang koordinat di mana gambar wajah dikompresi ke dimensi yang lebih sedikit, sambil mempertahankan varians maksimum di setiap arah sub-ruang ortogonal. Gambar wajah, yang serupa dalam konfigurasi keseluruhan, tidak didistribusikan secara acak dalam ruang gambar yang sangat besar ini dan dengan demikian dapat dijelaskan oleh sub-ruang dimensi yang relatif rendah. Ide utamanya adalah untuk menemukan vektor yang paling baik menjelaskan distribusi gambar wajah

dalam seluruh ruang gambar. Vektor-vektor dengan dimensi tereduksi ini mendefinisikan subruang dari citra wajah, yang disebut sebagai ruang wajah.

b. Pengenalan wajah berbasis *neural network*

Solusi nonlinier lebih lanjut untuk masalah pengenalan wajah adalah dengan memanfaatkan jaringan saraf. Keuntungan dari jaringan syaraf tiruan dalam klasifikasi dibandingkan yang linier adalah bahwa metode ini dapat mengurangi kesalahan klasifikasi di antara kelas-kelas tetangga. Secara umum, struktur jaringan sangat bergantung pada bidang aplikasinya, sehingga konteks yang berbeda menghasilkan jaringan yang sangat berbeda.

Kelas jaringan saraf yang dikenal sebagai peta pengorganisasian diri (SOM) juga diuji dalam pengenalan wajah, untuk mengeksplorasi sifat khusus mereka. Jaringan saraf berbasis keputusan probabilistik digunakan dalam deteksi wajah, pelokalan mata, dan juga dalam pengenalan wajah. Fleksibilitas jaringan ini karena struktur hierarkinya dengan fungsi basis nonlinier dan skema penugasan kredit yang kompetitif. Pendekatan berbasis jaringan saraf menemui masalah ketika jumlah kelas meningkat. Namun, tidak cocok untuk tugas pengenalan gambar model tunggal, karena beberapa gambar model per-orang diperlukan untuk melatih sistem.

c. Pendekatan dengan *Support Vector Machine* (SVM)

Pengenalan wajah merupakan masuk ke dalam permasalahan kelas K , dimana K adalah jumlah individu yang diketahui. Jadi, mesin vektor pendukung (SVM) yang berhubungan dengan metode klasifikasi biner dapat diterapkan pada situasi seperti itu, dengan merumuskan kembali masalah pengenalan wajah dan menafsirkan kembali *output* dari pengklasifikasi SVM. Masalah dirumuskan sebagai masalah perbedaan ruang, yang

memodelkan ketidaksetaraan antara dua citra wajah. Dalam ruang perbedaan, pengenalan wajah merupakan masalah dua kelas yang menunjukkan ketidakmiripan antara wajah orang yang sama dan ketidakmiripan antara wajah orang yang berbeda. Dengan memodifikasi interpretasi permukaan keputusan yang dihasilkan oleh SVM, metrik kesamaan dapat dihasilkan antara wajah yang dipelajari dari contoh perbedaan antara wajah.

5. *Dataset untuk face recognition*

Seperti yang kita ketahui bahwa data adalah hal yang sangat penting bagi pembelajaran mesin. Dengan sumber data yang berkualitas, maka akan menghasilkan model yang bagus pula. Sama halnya dengan dataset untuk *face recognition* pada *computer vision*. Untuk membandingkan berbagai teknik deteksi yang sudah dijelaskan sebelumnya maupun untuk menilai seberapa baik metode tersebut bekerja, beberapa *database* gambar wajah telah dikembangkan. Beberapa *database* wajah dihasilkan oleh kelompok penelitian berbeda yang memberikan variasi citra sebanyak mungkin. Setiap *database* dirancang untuk mengatasi tantangan spesifik yang mencakup berbagai skenario.

Database wajah yang tersedia untuk umum berisi gambar wajah dengan berbagai macam pose, sudut iluminasi, gerakan, oklusi wajah dan warna iluminasi. Seluruh rangkaian gambar, serta anotasi dan hasil eksperimen, ditempatkan di domain publik dan tersedia untuk diunduh melalui web di seluruh dunia. Berikut contohnya:

- a. ORL *dataset*
- b. OULU *physics dataset*
- c. XM2VTS *dataset*
- d. Yale *dataset*
- e. Yale-B *dataset*
- f. MIT *dataset*
- g. PIE *dataset*
- h. UMIST *dataset*

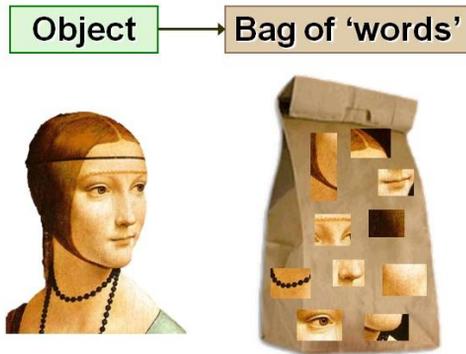
- i. PURDU AR *dataset*
- j. FERET *dataset*
- k. Faces in the Wild
- l. MegaFace
- m. WIDER FACE
- n. YouTube Faces (YTF)

D. Deteksi Visual 3D

Deteksi dan pelacakan multi-objek 3D adalah teknologi yang memungkinkan kita untuk mendeteksi *instance* dari kelas tertentu. Ada banyak aplikasi seperti robotika, kendali otomatis, dan *augmented reality*. Istilah 'kotak pembatas' sering digunakan untuk mendefinisikan wilayah yang diinginkan dan kemudian digunakan untuk mengklasifikasikan ke dalam kategori masing-masing.

Pendeteksian target berdimensi-3 bekerja dengan menemukan semua *Region of Interest* (RoI) pada citra dan menentukan posisi serta kategorinya. Karena penampilan, bentuk dan sikap yang berbeda dari berbagai objek, serta gangguan pencahayaan, pelindung dan faktor lainnya selama pencitraan, deteksi objek terus menjadi masalah yang menantang di bidang visi komputer (Wang & Ye, 2020).

Tugas visi komputer dalam analisis bentuk 3D memerlukan ekstraksi fitur bentuk 3D pula seperti pengambilan bentuk, pencocokan, segmentasi, estimasi pose manusia, pelacakan, hingga pemahaman suatu adegan/kejadian. Fitur 3D dapat diekstraksi melalui deskriptor bentuk global atau lokal. Deskriptor bentuk global adalah pendekatan umum untuk klasifikasi atau pengambilan bentuk yang dapat dianggap sebagai pemetaan dari ruang objek 3D ke ruang vektor dengan dimensi terbatas. Sedangkan, deskriptor bentuk lokal lebih banyak digunakan karena tidak hanya dapat menyediakan fitur lokal untuk pencocokan bentuk atau segmentasi bentuk berbasis titik, tetapi juga membangun fitur global dengan deskripsi bentuk global tambahan seperti *Bag-of-Words* (Jiang, Hadid, Pang, Granger, & Feng, 2019).



Gambar 75. Ilustrasi *Bag-of-Words*.

Teknik pengenalan 3D yang paling banyak digunakan adalah algoritma *Iterative Closest Point (ICP)*, yang bergantian antara menemukan titik terdekat yang cocok antara dua permukaan yang disejajarkan dan kemudian memecahkan masalah orientasi absolut 3D. Beberapa teknik, menggunakan bidang singgung permukaan lokal untuk membuat perhitungan ini lebih akurat dan untuk mempercepat konvergensi. Untuk mempercepat penentuan titik terdekat, dan juga untuk membuat perhitungan jarak ke permukaan lebih akurat, salah satu dari dua set titik dapat diubah menjadi fungsi jarak bertanda. Kelemahannya, algoritma ICP dan variannya hanya dapat menemukan keselarasan optimal lokal antara permukaan 3D.

E. Deteksi Video dan *Motion*

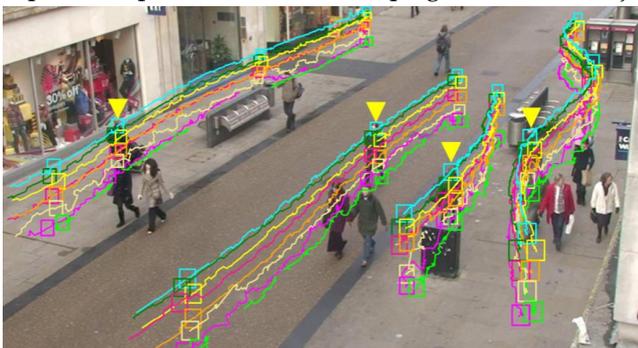
Deteksi *motion* dapat didefinisikan sebagai proses mendeteksi perubahan fisik pada posisi suatu objek sehubungan dengan latar belakangnya. Secara umum gerak dapat dideteksi menggunakan berbagai modalitas seperti Inframerah, energi frekuensi radio, suara, getaran dan magne. Di antara semua pendekatan ini, pendekatan berbasis kamera yang paling banyak digunakan dalam pengembangan teknik berbasis visi komputer untuk mendeteksi objek bergerak. Penerapan deteksi gerak berbasis visi komputer seperti di bidang pengawasan video cerdas, pemantauan lalu lintas, deteksi peristiwa, pelacakan orang dan analisis perilaku dan masih banyak lagi.

Deteksi objek bergerak adalah proses mengekstrak objek yang diinginkan dengan mengekstraknya dari lingkungan sekitarnya. Untuk mengekstrak latar depan dari lingkungan, video yang diambil kamera terlebih dahulu didekomposisi menjadi bingkai dan kemudian piksel dari bingkai ini diklasifikasikan menjadi dua set, satu milik latar belakang dan yang lainnya milik latar depan (Manchanda & Sharma, 2016).

Sebuah video tidak lain adalah serangkaian gambar. Video juga mengubah arsitektur model *deep learning* karena harus mempertimbangkan fitur temporal. Klasifikasi video adalah tugas pelabelan video dengan kategori. Sebuah kategori dapat berupa bingkai atau keseluruhan video. Klasifikasi video dapat berupa pelabelan objek yang ada dalam video atau melabeli tindakan yang terjadi dalam video.

Berikut adalah langkah-langkah penting bagaimana proses mesin dapat memahami objek bergerak (Viras, 2019):

1. Video disegmentasi ke dalam bingkai, sehingga sebuah video diubah menjadi gambar melalui pembuatan *frame* per satuan waktu.
2. Dua gambar/bingkai (A & B) yang diambil secara berurutan.
3. Mengubah gambar A & B menjadi skala abu-abu (*grayscale*).
4. Menghitung perbedaan antara dua gambar skala abu-abu.
5. Jika, perbedaan signifikan terdeteksi antara *frame* A & B, dapat disimpulkan bahwa beberapa gerakan telah terjadi.



Gambar 76. *Motion tracking* objek bergerak.

BAB 17

Teknik Deteksi Objek

Teknik untuk deteksi objek termasuk ke dalam pendekatan berbasis *machine learning* dan pendekatan berbasis *deep learning*. Dalam bab ini berisi pembahasan mengenai pendekatan yang digunakan dalam metodologi deteksi objek berbasis pembelajaran mendalam. Metodologi *Deep Learning* mampu melakukan deteksi objek dengan algoritma yang dirancang untuk melakukan deteksi objek didasarkan pada dua pendekatan. Yaitu deteksi objek satu tahap dan deteksi objek dua tahap. Detektor satu tahap memiliki kecepatan inferensi yang tinggi dan detektor dua tahap memiliki lokalisasi dan akurasi pengenalan yang tinggi. Detektor satu tahap memiliki kecepatan inferensi yang tinggi dan detektor dua tahap memiliki lokalisasi dan akurasi pengenalan yang tinggi.

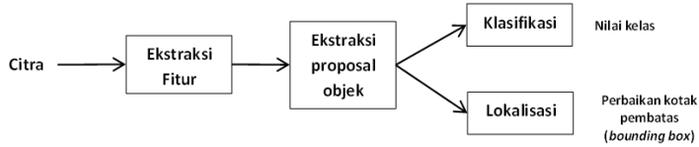
A. Detektor Objek Dua Tahap dan Satu Tahap

1. Detektor Objek Dua Tahap (*Two-stage Detector*)

Model deteksi objek memprediksi kotak pembatas (*bounding box*), satu untuk setiap objek yang ditemukan, serta probabilitas klasifikasi untuk setiap objek. Deteksi objek dua tahap mengacu pada penggunaan algoritma yang memecah pernyataan masalah deteksi objek menjadi dua tahap berikut:

- Mendeteksi kemungkinan wilayah objek.
- Mengklasifikasikan gambar di wilayah tersebut ke dalam kelas objek.

Algoritma dua langkah yang populer seperti Fast-RCNN dan Faster-RCNN biasanya menggunakan Jaringan Proposal Wilayah yang mengusulkan wilayah yang diinginkan yang mungkin berisi objek (Bandyopadhyay, 2022).



Gambar 77. Proses deteksi 2 tahap.

Dibandingkan dengan detektor satu tahap, detektor dua tahap memiliki keuntungan sebagai berikut (Lu, Li, Li, & Yan, 2020):

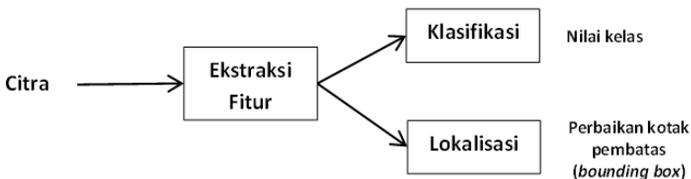
- a. Dengan mengambil sampel kumpulan proposal wilayah yang jarang, detektor dua tahap menyaring sebagian besar proposal negatif, sedangkan detektor satu tahap secara langsung menghadap semua daerah pada gambar dan memiliki masalah ketidakseimbangan kelas jika tidak ada desain khusus yang diperkenalkan.
- b. Karena detektor dua tahap hanya memproses sejumlah kecil proposal, kepala jaringan (untuk klasifikasi dan regresi proposal) dapat lebih besar dari detektor satu tahap, sehingga fitur yang lebih lengkap akan diekstraksi.
- c. Detektor dua tahap memiliki fitur berkualitas tinggi dari proposal sampel dengan menggunakan operasi RoI (*Region of Interest*) *Align* yang mengekstrak fitur lokasi yang konsisten dari setiap proposal. Namun, proposal wilayah yang berbeda dapat berbagi fitur yang sama dalam pendeteksi satu tahap dan representasi proposal yang kasar dan implisit secara spasial dapat menyebabkan ketidakselarasan fitur yang parah.
- d. Detektor dua tahap meregresi lokasi objek dua kali (sekali pada setiap tahap) dan kotak pembatas lebih disempurnakan daripada metode satu tahap.

No.	Model	Tahun
1.	R-CNN (<i>Region-based CNN detector</i>)	2014
2.	Fast R-CNN (versi lebih cepat dari R-CNN)	2015
3.	Faster R-CNN+++	2016
4.	Faster R-CNN (menggunakan <i>novel</i> RPN)	2017
5.	Mask R-CNN (ekstensi dari Faster R-CNN)	2017
6.	FPN (<i>Feature Pyramid Network</i>)	2017
7.	Faster R-CNN G-RMI	2017
8.	<i>Deformable</i> R-FCN	2017
9.	<i>Cascade</i> R-CNN	2018
10.	<i>Relation Networks for Object Detection</i>	2018
11.	DCNv2+Faster R-CNN	2018
12.	PANet	2018
13.	<i>Deeply Supervised Salient Object Detection with Short Connections</i>	2020

Pada Tabel 12 di atas dapat dilihat bagaimana perkembangan *detector* 2 tahap berkembang dari waktu ke waktu (Lohia, Kadam, Joshi, & Bongale, 2021). Untuk saat ini, detektor dua lebih unggul dalam akurasi pendeteksian. Dalam detektor jenis ini, proposal wilayah jarang dihasilkan pada tahap pertama dan kemudian diregresi lebih lanjut dan diklasifikasikan pada tahap kedua.

2. Detektor Objek Satu Tahap (*One-stage Detector*)

Di lain sisi, detektor satu tahap hanya membutuhkan satu lintasan melalui jaringan saraf dan memprediksi semua kotak pembatas sekaligus. Teknik ini jauh lebih cepat dan sangat cocok untuk perangkat seluler. Contoh paling umum dari pendeteksi objek satu tahap adalah dengan menggunakan YOLO, SSD, SqueezeDet, dan DetectNet.



Gambar 78. Proses deteksi 1 tahap

Pada Tabel 13 berikut dapat dilihat bagaimana perkembangan *detector* 1 tahap berkembang dari waktu ke waktu (Lohia, Kadam, Joshi, & Bongale, 2021).

No	Model	Tahun
1.	YOLO (<i>You Only Look Once</i>)	2016
2.	SSD (<i>Single Shot Detector</i>)	2016
3.	YOLOv2 (versi ke-2 YOLO)	2017
4.	DSSD (<i>Deconvolutional Single Shot Detector</i>)	2017
5.	RetinaNet	2017
6.	DCN	2017
7.	DCNv2	2018
8.	M2Det	2018
9.	YOLOv3 (versi ke-3 YOLO)	2018
10.	CornerNet512	2018
11.	NAS-FPN	2019
12.	EfficientDet	2020
13.	Fast-D	2020
14.	Yolov4	2020

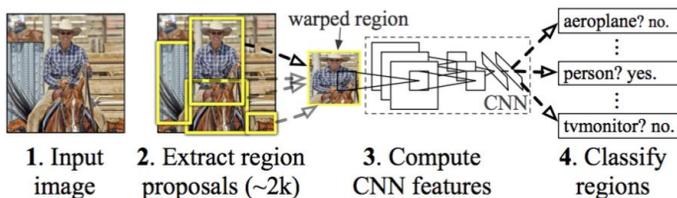
Tabel 13. Perkembangan *One-stage Object Detector*.

B. R-CNN (*Region-based CNN*)

R-CNN adalah teknik pendeteksian objek yang menggunakan pendekatan CNN berbasis wilayah untuk menerapkan pencarian selektif dengan jaringan saraf. Algoritma pendeteksian objek ini adalah penemuan yang menghasilkan hasil yang baik dalam kumpulan data besar untuk deteksi dan pelokalan objek. Arsitektur R-CNN dapat dilihat pada Gambar X.

R-CNN menggunakan algoritma *computer vision* tingkat rendah seperti *Selective Search* dan *Edge Boxes* untuk menghasilkan proposal, kemudian mengadopsi CNN untuk mengekstrak fitur untuk melatih SVM *classifier* dan *bounding box regressor* (Lu, Li, Li, & Yan, 2020). Adapun komponen utama R-CNN adalah sebagai berikut (Poonkuntran, Dhanraj, & Balusamy, 2022):

1. *Input* citra: Model mengambil gambar sebagai masukan.
2. Ekstrak proposal wilayah: Model kemudian mengekstrak proposal wilayah menggunakan metode pencarian selektif. Metode pencarian selektif secara iteratif mengelompokkan wilayah *berdasarkan* fitur umum yang mungkin berisi objek dan kemudian memilih proposal wilayah untuk diproses lebih lanjut.
3. Ekstraksi fitur dari proposal wilayah yang diekstraksi: Model CNN yang telah dilatih sebelumnya (misalnya, AlexNet, ResNet) kemudian digunakan untuk mengekstrak fitur dari proposal wilayah ini. Untuk menyesuaikan setiap proposal wilayah agar sesuai *sebagai* input ke R-CNN, dimensi gambar diubah di wilayah itu agar sesuai dengan CNN menggunakan dilasi dan pembungkus.
4. Modul klasifikasi: Berdasarkan fitur yang diperoleh pada tahap sebelumnya, mesin vektor pendukung (SVM) digunakan untuk mengklasifikasikan objek *yang* terlihat. SVM *classifier* memiliki tingkat *recall* yang tinggi tetapi presisi rendah sehingga diimplementasikan sebagai tahap pertama dalam pipa deteksi.
5. Modul lokalisasi: Memprediksi koordinat *bounding box* yang akan digambar di atas objek yang dilokalkan. Proses ini juga dikenal sebagai masalah regresi untuk menggambar *bounding box*. Langkah ini *menghasilkan* empat parameter (x, y, w, h). Dimana (x, y) adalah koordinat asal dan (w, h) adalah lebar dan tinggi kotak.



Gambar 79. Cara kerja R-CNN.

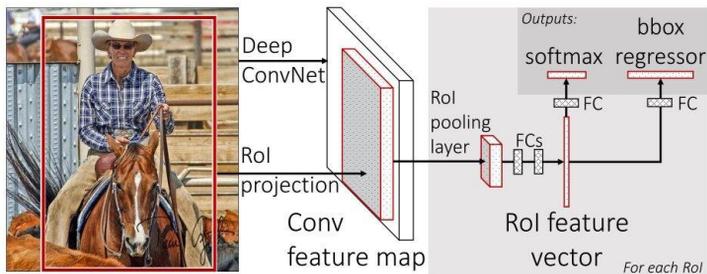
Ada beberapa kelebihan dan kekurangan menggunakan R-CNN yaitu sebagai berikut:

Kelebihan	Kekurangan
Dapat menerapkan CNN apa pun untuk ekstraksi fitur	Tidak terlalu cepat dan melakukan perhitungan yang berlebihan
Kinerja CNN dapat ditingkatkan dengan menerapkan arsitektur konvolusi yang lebih baik	Proposal objek harus diskalakan ulang ke resolusi dan rasio aspek yang ditetapkan
Mudah untuk menentukan jumlah cluster hanya dengan melihat Dendrogram	proses pelatihan yang sangat sulit dengan kapasitas sistem file yang besar

Tabel 13. Kelebihan dan Kekurangan R-CNN.

C. Fast R-CNN

Fast R-CNN adalah pendeteksi objek yang dikembangkan sendiri oleh Ross Girshick, peneliti AI *Facebook* dan mantan Peneliti *Microsoft*. Fast R-CNN mengatasi beberapa masalah di R-CNN. Seperti namanya, salah satu keunggulan Fast R-CNN dibandingkan R-CNN adalah perbaikan kecepatannya (Gad, 2020). Fast R-CNN mengusulkan untuk mengekstrak fitur untuk setiap proposal pada peta fitur bersama dengan penggabungan piramida spasial (Lu, Li, Li, & Yan, 2020).



Gambar 80. Cara kerja Fast R-CNN.

Arsitektur Fast R-CNN digambarkan pada Gambar 80. Serangkaian proposal wilayah dikirim ke jaringan bersama

dengan gambar *input*. Kotak merah pada gambar masukan adalah contoh visual usulan wilayah untuk gambar ini. Untuk menghasilkan banyak proposal wilayah untuk setiap gambar, Fast R-CNN tidak membuatnya melainkan menerimanya sebagai *input*. Komponen utama Fast R-CNN yaitu sebagai berikut:

1. ConvNet: Untuk ekstraksi fitur, untuk memproses seluruh gambar jaringan menggunakan kombinasi lapisan konvolusi dan *max-pooling*. Fitur yang diekstraksi adalah peta fitur.
2. Lapisan penyatuan RoI (*Region of Interest*): Merupakan ilustrasi sederhana dari lapisan *Spatial Pyramid Pooling* (SPP) dengan hanya satu lapisan piramida. Dengan cara mengambil vektor fitur dalam jendela ukuran tetap dari peta fitur konvolusi dan memasukkannya ke lapisan yang terhubung penuh atau biasa disebut *Fully Connected* (FC) *layers*. Lapisan *max-pooling* digunakan untuk menghasilkan peta fitur yang lebih kecil dengan tingkat spasial yang tetap ($\text{tinggi} \times \text{lebar}$)
3. Lapisan *output*: Memiliki dua lapisan keluaran paralel yaitu keluaran pertama menggunakan lapisan yang terhubung penuh untuk mengklasifikasikan gambar, diikuti oleh pengklasifikasi *softmax*. Dan kaluaran kedua menggunakan pelatihan multi-tugas untuk pelatihan simultan dari pengklasifikasi dan regresi tidak beraturan. Ukuran *bounding box* dan koordinat lokasi untuk objek yang diklasifikasikan adalah outputnya.

Ada beberapa kelebihan dan kekurangan menggunakan Fast R-CNN yaitu sebagai berikut:

Kelebihan	Kekurangan
Dibandingkan dengan R-CNN yang memiliki beberapa tahapan, Fast R-CNN membangun jaringan yang hanya memiliki satu tahap.	<i>Region Proposal Network</i> (RPN) dilatih di mana semua jangkar dilakukan <i>mini-batch</i> untuk ukuran 256, lalu diekstraksi dari

Fast R-CNN tidak meng- <i>cache</i> fitur yang diekstraksi sehingga tidak membutuhkan banyak penyimpanan <i>disk</i> .	satu gambar. Karena semua sampel dari satu gambar mungkin berkorelasi (yaitu fiturnya serupa), jaringan mungkin memerlukan banyak waktu untuk bisa melakukan konvergensi.
Lebih akurasi dibanding R-CNN.	

Tabel 14. Kelebihan dan Kekurangan Fast R-CNN.

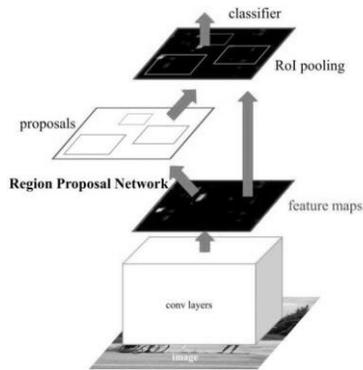
D. Faster R-CNN

Faster R-CNN diperkenalkan pada tahun 2015 oleh Kaiming He setelah diketahui hambatan arsitektur Fast R-CNN adalah pencarian selektif. Dimana teknik tersebut perlu menghasilkan 2.000 proposal per-gambar inovasi Face R-CNN terletak pada waktu pelatihan seluruh arsitektur. Di Faster R-CNN, proses tersebut digantikan oleh jaringan proposal wilayah (pawangfg, 2020).

Faster R-CNN mengintegrasikan proses usulan wilayah ke dalam jaringan yang dalam dan membuat seluruh detektor menjadi model yang dapat dilatih dari ujung ke ujung (Lu, Li, Li, & Yan, 2020). Komponen Faster R-CNN adalah sebagai berikut:

1. *Region Proposal Network* (RPN): Dibangun untuk dilatih dari awal hingga akhir hingga menghasilkan resolusi lapangan berkualitas tinggi, yang kemudian digunakan R-CNN untuk deteksi. *Output* dari RPN adalah objek prediksi yang terikat dan skor objek untuk setiap piksel.
2. Detektor Fast R-CNN: Model Fast R-CNN yang telah dilatih sebelumnya *digunakan* sebagai pendeteksi objek.
- 3.

Modul RPN bertanggung jawab untuk menghasilkan proposal wilayah yang menerapkan konsep perhatian pada jaringan saraf, sehingga memandu modul deteksi Fast R-CNN saat mencari objek dalam gambar. Cara kerja Faster R-CNN dapat dilihat pada Gambar 81.



Gambar 81. Cara kerja Faster R-CNN.

Dari ketiga teknik yang sudah dijabarkan yaitu R-CNN, Fast R-CNN, dan Faster R-CNN, berikut pada Tabel 15 dirangkum perbedaan ketiganya (Verma, 2021).

Kategori	R-CNN	Fast R-CNN	Faster R-CNN
Metode proposal wilayah	Pencarian selektif	Pencarian selektif	<i>Region Proposal Network (RPN)</i>
Waktu prediksi	40-50 detik	2 detik	0,2 detik
Komputasi	Waktu komputasi tinggi	Waktu komputasi tinggi	Waktu komputasi rendah
mAP pada set data uji Pascal VOC 2007 (%)	58,5	<ul style="list-style-type: none"> • 66.9 (dilatih dengan VOC 2007) • 70.0 (saat dilatih dengan VOC 2007 dan 2012) 	<ul style="list-style-type: none"> • 69.9 (dilatih dengan VOC 2007)
mAP pada dataset uji	53,3	<ul style="list-style-type: none"> • 65.7 (dilatih 	<ul style="list-style-type: none"> • 67.0 (dilatih

Pascal VOC 2012 (%)		dengan VOC 2012) <ul style="list-style-type: none"> • 68.4 (saat dilatih dengan VOC 2007 dan 2012) 	dengan VOC 2012) <ul style="list-style-type: none"> • 70.4 (dilatih dengan VOC 2007 dan 2012) • 75.9 (dilatih dengan VOC 2007, 2012 dan COCO)
---------------------	--	--	--

Tabel 15. Perbandingan R-CNN, Fast R-CNN dan Faster R-CNN.

Mean Average Precision (mAP) adalah metrik yang digunakan untuk mengevaluasi model deteksi objek seperti Fast R-CNN, YOLO, Mask R-CNN, dll. Nilai *mean* dari *average precision* (AP) dihitung dari nilai recall dari 0 hingga 1. Rumus mAP didasarkan pada sub metrik berikut (Shah, 2022):

1. *Confusion Matrix*

Adalah matriks yang berisi atribut *True Positif* (TP): Model memprediksi label dan cocok dengan benar sesuai kebenaran dasar. *True Negatif* (TN): Model tidak memprediksi label dan bukan merupakan bagian dari kebenaran dasar. *False Palsu* (FP): Model memprediksi label, tetapi itu bukan bagian dari kebenaran dasar (Kesalahan Tipe I). dan *False Negatif* (FN): Model tidak memprediksi label, tetapi merupakan bagian dari kebenaran dasar. (Kesalahan Tipe II).

2. *Intersection over Union* (IoU)

Persimpangan di atas Union menunjukkan tumpang tindih dari koordinat kotak pembatas yang diprediksi ke kotak kebenaran dasar. IoU yang lebih tinggi menunjukkan koordinat kotak pembatas yang diprediksi sangat mirip dengan koordinat kotak kebenaran dasar.

3. Recall

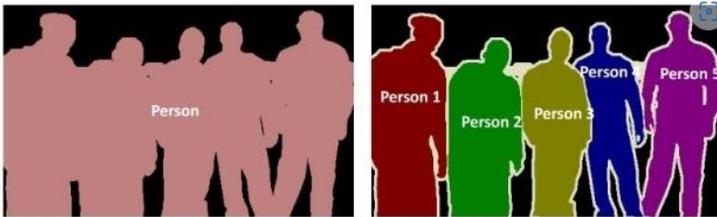
Recall mengukur seberapa baik model dapat menemukan true positif (TP) dari semua prediksi (TP+FN).

4. Precision

Presisi mengukur seberapa baik model dapat menemukan true positif (TP) dari semua prediksi positif. (TP+FP).

E. Mask R-CNN

Mask R-CNN adalah *Convolutional Neural Network* (CNN) yang canggih dalam hal segmentasi gambar dan segmentasi instan (Setyaningsih & Edy, 2022). Mask R-CNN dikembangkan di atas Faster R-CNN, Jaringan Saraf Konvolusi Berbasis Wilayah. Langkah pertama untuk memahami cara kerja Mask R-CNN membutuhkan pemahaman tentang konsep Segmentasi Citra. Tugas visi komputer dalam segmentasi citra adalah proses mempartisi gambar digital menjadi beberapa segmen (kumpulan piksel, juga dikenal sebagai objek gambar). Segmentasi ini digunakan untuk menemukan objek dan batas (garis, kurva, dll.). Ada 2 jenis utama dari segmentasi gambar yang termasuk dalam Mask R-CNN yaitu segmentasi semantic dan segmentasi instan (Odemakinde, 2022).



Gambar 82. Segmentasi semantik (kiri) dan segmentasi instan (kanan).

1. Segmentasi Instan

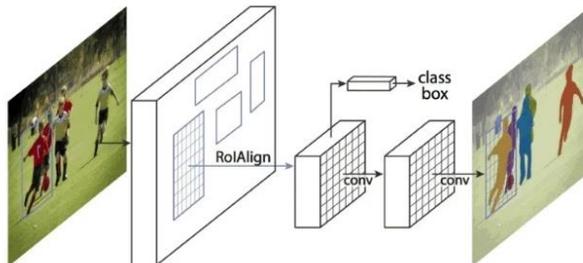
Segmentasi ini berkaitan dengan deteksi yang sesuai dari semua objek yang ada di dalam satu gambar. Oleh karena itu, kombinasi dari deteksi objek, lokalisasi objek, dan klasifikasi objek merupakan faktor penting dalam segmentasi instan. Dengan kata lain, jenis segmentasi ini memiliki fokus

untuk memberikan perbedaan yang jelas antara setiap objek yang diklasifikasikan sebagai *instance* serupa. Seperti yang ditunjukkan pada contoh Gambar 82 (kanan) di atas, untuk segmentasi instan semua objek adalah orang, tetapi proses segmentasi ini memisahkan setiap orang sebagai satu kesatuan.

2. Segmentasi Semantik

Dikenal sebagai segmentasi latar depan karena menonjolkan subjek gambar, bukan latar belakang. Dalam segmentasi semantik, setiap piksel diklasifikasikan menurut kelas objeknya (Géron, 2019). Misalnya, jalan, mobil, pejalan kaki, bangunan dan lain sebagainya, seperti yang ditunjukkan pada Gambar 82 (kiri). Dapat dilihat bahwa objek yang berbeda dari kelas yang sama tidak dibedakan. Kesulitan utama dalam tugas segmentasi ini adalah bahwa ketika gambar melalui CNN biasa, citra secara bertahap kehilangan resolusi spasial karena lapisan dengan langkah lebih besar dari 1. Sehingga mengurangi akurasi CNN biasa dalam mengenali objek manusia.

Adapun kemiripan antara Faster R-CNN dan Mask R-CNN yaitu keduanya memiliki cabang untuk klasifikasi dan regresi *bounding box*. Kedua teknik ini juga menggunakan arsitektur ResNet 101 untuk mengekstrak fitur dari gambar dan menggunakan Jaringan Proposal Wilayah (RPN) untuk menghasilkan Wilayah Kepentingan (RoI).



Gambar 83. Cara kerja Mask R-CNN.

Cara kerja Mask R-CNN adalah sebagai berikut (Khandelwal, 2019):

- a. CNN dijalankan pada citra untuk menghasilkan peta fitur.
- b. *Region Proposal Network* (RPN) menggunakan CNN untuk menghasilkan beberapa *Region of Interest* (RoI) menggunakan pengklasifikasi biner ringan. Proses ini dilakukan dengan menggunakan 9 kotak jangkar di atas gambar. Pengklasifikasi mengembalikan skor objek ataupun tanpa objek. Penekanan *Non Max* diterapkan pada Jangkar dengan skor objektivitas tinggi.
- c. Jaringan RoI *Align* mengeluarkan beberapa *bounding box* dari satu kotak yang sesuai dan melengkungkannya ke dalam dimensi yang tetap.
- d. Fitur melengkung kemudian dimasukkan ke dalam lapisan yang terhubung sepenuhnya untuk membuat klasifikasi menggunakan *softmax* dan prediksi *bounding box* lebih disempurnakan menggunakan model regresi.
- e. Fitur melengkung juga dimasukkan ke dalam pengklasifikasi *Mask*, yang terdiri dari dua CNN untuk mengeluarkan *mask* biner untuk setiap RoI. *Mask Classifier* memungkinkan jaringan untuk menghasilkan *mask* untuk setiap kelas tanpa persaingan antar kelas.

Kunci utama Mask R-CNN adalah penyelarasan piksel-ke-piksel, yang merupakan bagian utama dari Fast R-CNN dan Faster R-CNN yang hilang. Mask R-CNN mengadopsi prosedur dua tahap yang sama dengan tahap pertama yang identik (yaitu RPN). Mask R-CNN mudah diterapkan dan dilatih karena kerangka kerja Faster R-CNN yang memfasilitasinya berbagai desain arsitektur yang fleksibel. Selain itu, cabang *mask* hanya menambahkan *overhead* komputasi kecil, memungkinkan sistem yang cepat dan eksperimen yang cepat.

Teknik Mask R-CNN mudah dilatih. Performa kinerjanya mengungguli semua entri model tunggal yang ada pada setiap tugas. Metode ini juga sangat efisien dan

hanya menambahkan sedikit overhead ke Faster R-CNN. Mask R-CNN mudah digeneralisasi ke tugas lain yang menambah nilai fleksibilitasnya. Contohnya, Mask R-CNN bisa digunakan untuk estimasi pose manusia dalam *framework* yang sama.

Framework ini dapat dengan mudah diperluas ke estimasi pose manusia. Dengan memodelkan lokasi *keypoint* sebagai *one-hot mask*, dan mengadopsi Mask R-CNN untuk memprediksi K *mask*, satu untuk setiap jenis *keypoint* K . Misalnya bahu kiri, siku kanan, dengkul kaki kiri, dan lain sebagainya. Tugas ini membantu menunjukkan fleksibilitas Mask R-CNN. Eksperimen ini diutamakan untuk menunjukkan keumuman kerangka kerja Mask R-CNN. Penelitian yang dilakukan oleh (He, Gkioxari, Dollar, & Girshick, 2018) menunjukkan bagaimana mesin dapat mengenali berbagai pose manusia dalam posisi yang berbeda-beda seperti yang terlihat pada Gambar 84.



Gambar 84. Cara kerja Mask R-CNN.

Contoh lainnya adalah R-CNN digunakan untuk meningkatkan *OpenStreetMap* dengan menambahkan lapangan bisbol, sepak bola, tenis, sepak bola, dan bola basket. Proyek ini menerapkan algoritma Mask R-CNN untuk mendeteksi fitur dengan tujuan mengidentifikasi lapangan olahraga dari sumber citra satelit. Lapangan olahraga cocok untuk algoritma Mask R-CNN karena terlihat jelas untuk dideteksi dalam citra satelit terlepas dari tutupan pohon (tidak seperti bangunan). Selain itu, metode ini efisien karena lapangan olahraga berbentuk “gumpalan” dan bukan bentuk garis seperti jalan raya.



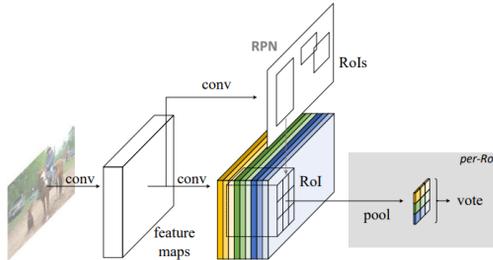
Gambar 85. Contoh pemetaan *OpenStreetMap* dengan Mask R-CNN.

F. R-FCN (*Region-based Fully Convolutional Network*)

R-FCN mengusulkan jaringan konvolusi penuh berbasis wilayah untuk menghasilkan fitur sensitif wilayah untuk deteksi (Lu, Li, Li, & Yan, 2020). Jaringan Konvolusi Penuh Berbasis Wilayah atau R-FCN adalah jenis detektor objek berbasis wilayah yang berbeda dengan detektor objek berbasis wilayah sebelumnya seperti Fast/Faster R-CNN yang menerapkan subjaringan per-wilayah sebanyak ratusan kali. R-FCN sepenuhnya *convolutional* pada hampir semua komputasi yang dibagikan di seluruh gambar. Untuk melakukan hal proses tersebut, R-FCN menggunakan peta skor posisi-sensitif untuk mengatasi dilema antara terjemahan-invarians dalam klasifikasi gambar dan terjemahan-varians dalam deteksi objek (Dai, Li, He, & Sun, 2016).

Seperti yang dijelaskan pada teknik-teknik sebelumnya, untuk pendekatan jaringan proposal wilayah tradisional (RPN) seperti R-CNN, Fast R-CNN dan Faster R-CNN, proposal wilayah dihasilkan oleh RPN terlebih dahulu. Kemudian *RoI pooling* dilakukan, dan melalui layer *fully connected* (FC) dilakukan klasifikasi dan regresi *bounding box*. Pada R-FCN, RPN masih digunakan untuk memperoleh proposal wilayah, tetapi tidak seperti seri R-CNN, lapisan FC setelah pengumpulan *RoI* dihapus. Sebaliknya, semua kompleksitas utama dipindahkan sebelum pengumpulan *RoI* untuk menghasilkan peta skor. Semua proposal wilayah, setelah pengumpulan *RoI*, akan menggunakan kumpulan peta skor yang sama untuk melakukan

pemungutan suara rata-rata. Dengan demikian, R-FCN menjadi lebih cepat daripada Faster R-CNN dengan mAP yang kompetitif (Tsang, Review: R-FCN – Positive-Sensitive Score Maps (Object Detection), 2018).



Gambar 86. Arsitektur R-FCN.

G. YOLO (*You Only Look Once*)

YOLO adalah arsitektur deteksi objek yang sangat cepat dan akurat. Keluarga YOLO juga merupakan kumpulan dari berbagai kerangka kerja deteksi keberatan yang berkembang dan meningkat selama bertahun-tahun . Pertama kali diusulkan oleh Joseph Redmon dalam makalah tahun 2015 yaitu YOLOv1 yang merupakan jaringan deteksi tunggal yang mengintegrasikan dua komponen utama, detektor dan pengklasifikasi. Kemudian ditingkatkan pada tahun 2016 dengan munculnya YOLOv2 atau YOLO9000 yang mampu mendeteksi 9.000 objek. Pada tahun 2018 YOLOv3 sebagai versi ketiga dirilis, model ini memiliki hasil yang lebih baik dari model sebelumnya. Hingga akhirnya disusul versi ke-4 pada tahun 2020. Model YOLO bekerja sangat cepat sehingga dapat berjalan secara *realtime* di dalam video (Géron, 2019).

Algoritma YOLO banyak digunakan untuk pendeteksian objek dan diimplementasikan menggunakan DCNN tunggal. Pada metode YOLO dengan Google Net, ciri-ciri suatu objek diperoleh dari citra (Jabbar, Prasad, Peng, Reaz, & Madureira, 2021). Selain dalam video, YOLO juga juga dirancang khusus untuk mendeteksi objek dalam gambar dengan mempertimbangkan kecepatan dan penggunaan waktu nyata (Poonkuntran, Dhanraj, & Balusamy, 2022).

- b. Model membagi gambar input ke sel $S \times S$ dan setiap sel menyediakan prediksi *bounding box*. Jadi, karena pengambilan sampel yang turun model menggunakan fitur yang agak kasar untuk memprediksi kotak pembatas.
- c. Sulit untuk melokalisasi objek atau kelompok objek minor. Oleh karena itu, sumber utama kesalahan adalah lokalisasi.

YOLOv1 memiliki *kelebihan* dan *kekurangan*, diantaranya yaitu sebagai berikut (Thatte, 2020) (Chablani, 2017):

Kelebihan	Kekurangan
Kecepatan 45 <i>frame</i> per-detik.	YOLOv1 mengalami kesulitan dalam mendeteksi benda-benda kecil yang muncul secara berkelompok.
Jaringan memahami representasi objek umum yang memungkinkan model ini untuk melatih jaringan pada gambar dunia nyata dan prediksi yang dihasilkan masih cukup akurat.	Mengalami kesulitan dalam mendeteksi objek yang memiliki rasio aspek yang tidak biasa.
Versi dinilai lebih cepat dengan arsitektur yang <i>relative</i> kecil.	Membuat lebih banyak kesalahan pelokalan dibandingkan dengan Fast R-CNN.

Tabel 16. Kelebihan dan Kekurangan YOLOv1.

2. YOLOv2

YOLO versi kedua ini juga dinamai YOLO9000 karena model ini mampu mendeteksi 9.000 objek. Ini adalah model dengan arsitektur tiga puluh lapisan (DarkNet-19) yang

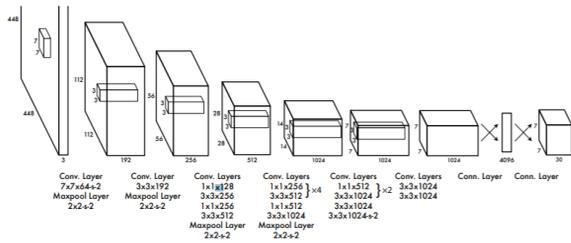
terbagi menjadi sembilan belas lapisan konvolusi dan sebelas lapisan tambahan untuk deteksi objek (Poonkuntran, Dhanraj, & Balusamy, 2022).

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Gambar 88. Arsitektur DarkNet-19.

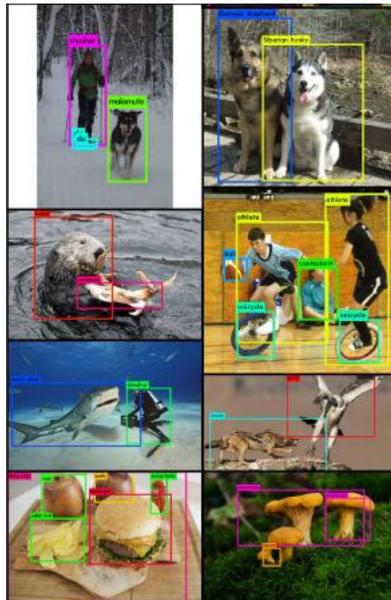
YOLOv2 mengatasi keterbatasan versi sebelumnya yaitu YOLOv1 untuk deteksi objek. YOLO memberikan tingkat *recall* yang rendah jika dibandingkan dengan teknik berbasis wilayah. YOLOv2 terutama berkonsentrasi pada peningkatan tingkat penarikan sambil mempertahankan akurasi klasifikasi. Dengan menambahkan normalisasi *batch* dalam nilai mAP YOLOv1, meningkatkan performa secara signifikan. Dengan modifikasi ini, lapisan *dropout* dihapus dari jaringan tanpa *overfitting*. YOLOv1 mengalami masalah resolusi pada output karena menerima ukuran gambar input 224×224 dan dikonversi ke 448 untuk deteksi objek. Oleh karena itu, YOLOv2 menyetel jaringan dengan ukuran gambar 448×448 untuk mengatasi masalah resolusi. Hal ini meningkatkan kompleksitas model deteksi objek, sehingga untuk mengurangi kerumitan YOLOv2 menghapus lapisan yang terhubung penuh dari YOLOv1 dan memperkenalkan kotak jangkar untuk mendeteksi kotak pembatas alih-alih koordinat.

Pada Gambar 89 dapat dilihat arsitektur YOLOv2 yang menggunakan algoritma pengelompokan *K-means* untuk mengenali kotak pembatas *K* tertinggi dari gambar. Dimana versi sebelumnya, YOLOv1 tidak memiliki kontrol pada prediksi lokasi, sehingga membuat tidak seimbang pada iterasi awal. (Jabbar, Prasad, Peng, Reaz, & Madureira, 2021).



Gambar 89. Arsitektur YOLOv2.

YOLO9000 dapat mendeteksi berbagai macam kelas objek secara *realtime* seperti yang terlihat pada Gambar 90 (Redmon & Farhadi, 2016).



Gambar 90. Mendeteksi berbagai macam kelas objek dengan YOLO9000/YOLOv2.

YOLOv2 memiliki kelebihan dan kekurangan, seperti yang dirangkum dalam Tabel 17 berikut (Hui, Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3, 2018):

Kelebihan	Kekurangan
Peningkatan akurasi dengan kenaikan mAP hingga 2%.	YOLO9000 menggunakan tiga prior (kotak jangkar) membatasi ukuran <i>output</i> .
YOLOv2 dimulai dengan gambar beresolusi 224×224 untuk pelatihan pengklasifikasi tetapi kemudian menyetel ulang pengklasifikasi lagi dengan resolusi 448×448 menggunakan waktu yang jauh lebih sedikit.	Model ini belajar untuk menemukan objek dalam gambar menggunakan data deteksi dalam <i>dataset</i> COCO dan belajar untuk mengklasifikasikan berbagai objek menggunakan data dari <i>dataset</i> ImageNet. Karena COCO tidak memiliki label kotak pembatas untuk banyak kategori, YOLO9000 kesulitan memodelkan beberapa kategori seperti "kacamata hitam" atau "celana renang".
Peningkatan kecepatan dengan GoogleNet dan DarkNet.	mendeteksi objek tunggal yang tidak cocok untuk beberapa kumpulan data seperti OID

Tabel 17. Kelebihan dan Kekurangan YOLOv2.

3. YOLOv3

YOLOv2 menggunakan jaringan standar DarkNet-19 untuk mengekstrak fitur, tetapi YOLOv3 menggunakan DarkNet-53 untuk ekstraksi fitur gambar. DarkNet-53 terdiri dari 53 lapisan *convolutional* dan lebih efisien daripada Darknet-19. Arsitektur DarkNet-53 dapat dilihat pada

Gambar 91. DarkNet-53 adalah arsitektur 53-layer yang dilatih melalui dataset ImageNet dan merupakan jaringan saraf yang sangat dalam. Model ini memiliki 53 lapisan tambahan untuk deteksi objek dengan total 106 lapisan yang terhubung penuh (Poonkuntran, Dhanraj, & Balusamy, 2022).

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	128 × 128
	Convolutional	64	3 × 3	
	Residual			
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	64 × 64
	Convolutional	128	3 × 3	
	Residual			
	Convolutional	256	3 × 3 / 2	32 × 32
3x	Convolutional	128	1 × 1	32 × 32
	Convolutional	256	3 × 3	
	Residual			
	Convolutional	512	3 × 3 / 2	16 × 16
4x	Convolutional	256	1 × 1	16 × 16
	Convolutional	512	3 × 3	
	Residual			
	Convolutional	1024	3 × 3 / 2	8 × 8
5x	Convolutional	512	1 × 1	8 × 8
	Convolutional	1024	3 × 3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 91. Arsitektur DarkNet-53.

Dalam metode YOLOv3, gambar input dibagi menjadi grid. Kotak jangkar dihitung dari setiap kotak objek terdekat. Setiap kotak jangkar memberikan nilai skor, memberikan informasi tentang seberapa akurat memprediksi objek. Satu kotak jangkar mendeteksi hanya satu objek iklan yang memberikan satu nilai skor. YOLOv3 memprediksi *offset* relatif terhadap koordinat sel grid, di mana (0,0) berarti bagian kiri atas suatu sel, dan (1,1) berarti bagian bawah kanan. Untuk setiap sel kisi, YOLOv3 dilatih untuk memprediksi hanya kotak pembatas yang pusatnya terletak di sel itu. YOLOv3 menerapkan fungsi aktivasi logistik ke koordinat kotak pembatas untuk memastikan mereka tetap dalam kisaran 0 hingga 1. YOLOv3 dinilai cepat dan akurat sehubungan dengan nilai mAP dan IoU. YOLOv3

mendeteksi beberapa objek dengan menghasilkan multilabel untuk beberapa kotak pembatas (Jabbar, Prasad, Peng, Reaz, & Madureira, 2021).

Keunggulan YOLOv3 dibanding model-model sebelumnya adalah ketepatan rata-rata untuk objek kecil meningkat, model ini lebih baik daripada Faster R-CNN namun masih berada di bawah RetinaNet. Karena ada peningkatan mAP secara signifikan, maka kesalahan pelokalan berkurang. Prediksi pada skala atau rasio aspek yang berbeda untuk objek yang sama ditingkatkan karena penambahan metode seperti piramida fitur (Sonawane, 2018).

4. YOLOv4

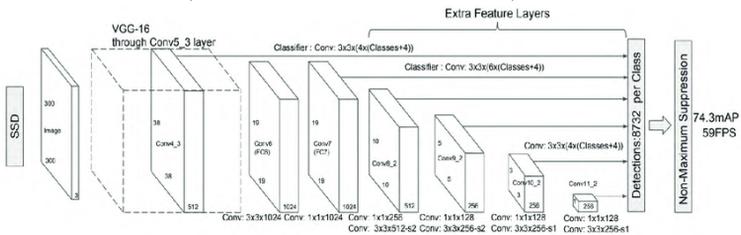
Perkembangan YOLO dan kesuksesan model-modelnya dengan cepat mendapat popularitas di kalangan komunitas *computer vision* karena kecepatannya yang luar biasa dengan akurasi yang baik. Namun, pada Februari 2020 Joseph Redmon, pencipta YOLO mengumumkan bahwa beliau telah menghentikan penelitiannya di bidang *computer vision* dikarenakan beberapa kekhawatirannya mengenai potensi dampak negatif dari pekerjaannya. Penelitian YOLO terus dilanjutkan di komunitas visi komputer, generasi ke-4 YOLO telah dirilis pada April 2020 oleh Alexey Bochkovskiy (Bochkovskiy, Wang, & Liao, 2020). YOLO v4 telah dianggap sebagai model *real-time* tercepat dan paling akurat untuk deteksi objek (Supeshala, 2002).

YOLO v4 juga berbasis DarkNet dan telah memperoleh nilai AP sebesar 43,5 persen pada *dataset* COCO bersama dengan kecepatan *real-time* 65 FPS pada *Tesla V100*, mengalahkan detektor tercepat dan paling akurat dalam hal kecepatan dan akurasi. Jika dibandingkan dengan YOLOv3, peningkatan AP sebesar 10% dan FPS sebesar 12%. Pada tahun 2020, YOLO terus berkembang dalam waktu yang sangat singkat. YOLOv5 rilis 2 bulan setelah versi ke-4, versi ini dikembangkan oleh Glenn Jocher, yang sudah dikenal di kalangan komunitas karena menciptakan implementasi

PyTorch yang populer dari YOLO v3. Versi milik Jocher dirilis ke publik pada Juni 2020. Namun versi tersebut keluar tanpa didampingi penulisan makalah. Versi YOLO selanjutnya yaitu YOLOv5 muncul pada Juli 2020 dan disusul oleh PP-YOLO. Yang merupakan peningkatan YOLO baru lainnya berdasarkan kerangka *deep learning* yang disebut *PaddlePaddle*, meningkatkan model YOLO v3 untuk mendapatkan keseimbangan yang lebih baik antara efektivitas dan efisiensi (Long, Deng, Wang, Zhang, & dkk, 2020).

H. SSD (Single Shot Detector)

Dalam aplikasi *real-time*, algoritma pendeteksian objek yang paling banyak digunakan adalah *Single Shot multibox Detector* (SSD) karena implementasinya yang mudah dan akurasi yang baik yang diimplementasikan menggunakan *Deep Neural Network* sederhana. Arsitektur SSD dapat dilihat pada Gambar 92 di bawah ini (Rohan, Rabah, & Kim, 2019).



Gambar 92. Proses deteksi 1 tahap.

SSD meningkatkan kecepatan deteksi dengan cara melakukan penghapusan data yang berlebihan. Algoritma deteksi objek SSD terdiri dari dua tahap, yaitu dengan membuat peta ekstraksi fitur dan yang menerapkan filter konvolusi untuk mendeteksi objek. Dalam algoritma SSD, fitur gambar diekstraksi menggunakan jaringan yang telah ditentukan Vgg16.

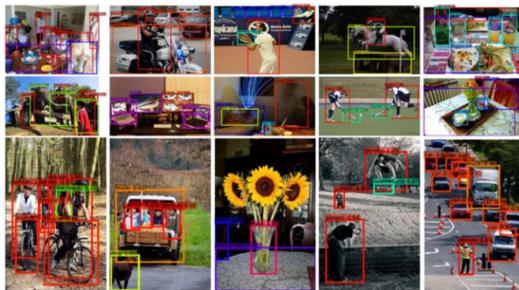
Algoritma ini menciptakan empat prediksi dari setiap sel. Setiap prediksi memberikan 21 skor untuk setiap kelas. *Bounding box* skor tertinggi terkait dengan citra kelas yang dibatasi. Jika banyak prediksi tidak memiliki skor yang valid, artinya tidak

ada objek yang terdeteksi. Pada tahap selanjutnya, filter konvolusi diterapkan antara lokasi dan skor kelas untuk membuat prediksi. Algoritma SSD juga dapat digunakan untuk mendeteksi beberapa objek (Jabbar, Prasad, Peng, Reaz, & Madureira, 2021).

SSD dirancang untuk deteksi objek secara *real-time*. Untuk memulihkan penurunan akurasi, SSD menerapkan beberapa peningkatan termasuk fitur multi-skala dan kotak *default*. Peningkatan ini memungkinkan SSD untuk mencocokkan akurasi Faster R-CNN menggunakan gambar beresolusi lebih rendah, yang selanjutnya mendorong kecepatan menjadi lebih tinggi.

Untuk objek yang lebih kecil, SSD berkinerja tidak lebih baik daripada Faster R-CNN. Dengan SSD, objek kecil hanya dapat dideteksi di lapisan resolusi lebih tinggi (lapisan paling kiri). Tetapi lapisan tersebut berisi fitur tingkat rendah, seperti tepi atau tambalan warna, yang kurang informatif untuk klasifikasi. SSD memiliki kesalahan pelokalan yang lebih rendah dibandingkan dengan R-CNN tetapi lebih banyak kesalahan klasifikasi yang berhubungan dengan kategori serupa (Hui, 2018).

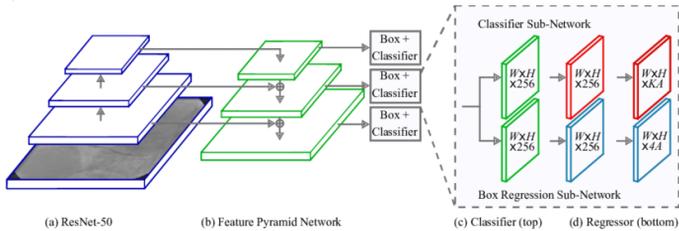
Dengan menggunakan SSD, kita hanya perlu mengambil satu bidikan untuk mendeteksi beberapa objek di dalam citra. SSD jauh lebih cepat dibandingkan dengan pendekatan berbasis RPN dua tembakan (Tsang, 2018). Pada Gambar 93 dapat dilihat bagaimana SSD dapat mendeteksi dengan benar objek-objek yang “bertindihan”.



Gambar 93. Hasil deteksi pada *dataset* MS COCO dengan SSD.

I. RetinaNet

Yi Lin pada 2018 merancang detektor objek satu tahap sederhana yang disebut RetinaNet, dinamai untuk pengambilan sampel padat lokasi objek dalam gambar input. Desainnya menampilkan piramida fitur dalam jaringan yang efisien dan penggunaan kotak jangkar. RetinaNet efisien dan akurat (Lin, Goyal, Girshick, He, & Dollár, 2018).



Gambar 94. Arsitektur RetinaNet dengan (a) ResNet-50, (b) Feature Pyramid Network sebagai ekstraktor fitur untuk (c) mengklasifikasikan probabilitas keberadaan lesi dan (d) meregresi koordinat kotak pembatas.

RetinaNet adalah model deteksi objek satu tahap yang menggunakan fungsi menghilangkan fokus untuk mengatasi ketidakseimbangan kelas selama pelatihan. Kehilangan fokus menerapkan istilah modulasi pada kehilangan entropi silang untuk memfokuskan pembelajaran pada contoh negatif yang sulit. RetinaNet adalah jaringan tunggal dan terpadu yang terdiri dari jaringan tulang punggung dan dua sub-jaringan khusus tugas. Tulang punggung bertanggung jawab untuk menghitung peta fitur konvolusi pada seluruh gambar input dan merupakan jaringan konvolusi mandiri. *Subnet* pertama melakukan klasifikasi objek convolutional pada *output backbone* sedangkan *subnet* kedua melakukan regresi *bounding box* konvolusi. Kedua *subnetwork* memiliki desain sederhana untuk deteksi padat satu tahap (Tan, Huangfu, Wu, & dkk, 2021).

DAFTAR PUSTAKA

- Acharya, T., & Ray, A. K. (2005). *Image Processing : Principles and Applications*. Canada: John Wiley & Sons, Inc.
- Akkaya, B., & Çolakoğlu, N. (2019). Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases. *y-BIS 2019 Conference Book: Recent Advances in Data Scence and Business Analytics* (pp. 162-171). Istanbul: Mimar Sinan Fine Arts University Publications.
- Amatullah, L., Ein, I., & Santoni, M. M. (2021). Identifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur dan Warna Dengan Menggunakan Metode K-Nearest Neighbor. *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, (hal. 783-791). Jakarta.
- AO Kaspersky Lab. (2022). *Traffic monitoring and congestion detection*. Retrieved November 1, 2022, from kaspersky: <https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition>
- Arif, N., & Wahyuni, F. S. (2016). Penggunaan Metode Machine Learning untuk Pengenalan Pola Tutupan Lahan pada Citra Satelit. *Seminar Nasional Teknologi Informasi dan Multimedia* (hal. 1-6). Yogyakarta: STMIK AMIKOM .
- Arsy, R. F. (2013). Metode Survei Deskriptif Untuk Mengkaji Kemampuan Interpretasi Citra pada Mahasiwa Pendidikan Geografi FKIP Universitas Tadulako. *Kreatif*, 16(3), 62-72.

- Babic, M., Farahani, M. A., & Wuest, T. (2021). Image Based Quality Inspection in Smart Manufacturing Systems: A Literature Review,. *Procedia CIRP*, 103, 262-267.
- Bandyopadhyay, H. (2022, Oktober 7). *YOLO: Real-Time Object Detection Explained*. Retrieved November 4, 2022, from V7: <https://www.v7labs.com/blog/yolo-object-detection#h3>
- Bernstein, C. (2020, Februari). *face detection*. Retrieved from SearchEnterpriseAI: <https://www.techtarget.com/searchenterpriseai/definition/face-detection#:~:text=Face%20detection%20%2D%2D%20also%20called,human%20faces%20in%20digital%20images.>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *Computer Vision and Pattern Recognition (cs.CV)*, 1-17.
- Boesch, G. (2022). *A Complete Guide to Image Classification in 2022*. Retrieved November 1, 2022, from viso.ai: <https://viso.ai/computer-vision/image-classification/>
- Brownlee, J. (2019, Maret 13). *9 Applications of Deep Learning for Computer Vision*. Retrieved November 1, 2022, from Machine Learning Mastery: <https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/>
- Brownlee, J. (2019). *Deep Learning for Computer Vision: Image Classification, Object Detection and Face Recognition in Python*. San Fransisco: Machine Learning Mastery.

Chablani, M. (2017, Agustus 21). *YOLO – You only look once, real time object detection explained*. Retrieved November 5, 2022, from Towards Data Science: <https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006>

Computation, I. o. (2021, Januari 30). *Preprocessing with Computer Vision Part VI: Contour Detection*. Retrieved Oktober 25, 2022, from Medium: <https://medium.com/perspectivesondatascience/preprocessing-with-computer-vision-part-vi-contour-detection-68a6fb2942e0>

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. *Computer Vision and Pattern Recognition (cs.CV)*, 1-11.

Damayanti, E. (2013). *Pengolahan Citra Digital 2 : Untuk SMK/MAK Kelas XI*. Malang: Kementerian Pendidikan & Kebudayaan RI.

Datta, A. K., Datta, M., & Banerjee, P. K. (2016). *Face Detection and Recognition Theory and Practice*. Boca Raton: CRC Press.

Davies, E. R. (2012). *Computer and Machine Vision : Theory, Algorithms, Practicalities 4th Edition*. Waltham: Elsevier Inc.

Dinata, R. K., & Hasdyna, N. (2020). *Machine Learning : Panduan Memahami Data Science, Supervised Learning, Unsupervised Learning dan Reinforcement Learning*. Lhokseumawe: Unimal Press.

- esri. (2020). *Australian State Automated Large-Area Land Classification with Machine Learning*. Retrieved November 1, 2022, from ArcNews:
<https://www.esri.com/about/newsroom/arcnews/australian-state-automated-large-area-land-classification-with-machine-learning/>
- Gad, A. F. (2020). *Ahmed Fawzy Gad*. Retrieved November 5, 2022, from PaperspaceBLog:
<https://blog.paperspace.com/faster-r-cnn-explained-object-detection/>
- Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (Second Edition ed.). Sebastopol: O'Reilly Media Inc.
- Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing Third Edition : Instructor's Manual*. London: Pearson.
- Google Developers. (2022, Juli 18). *k-Means Advantages and Disadvantages*. Retrieved 10 27, 2022, from Machine Learning Google Developers:
<https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages>
- Gunawan, K. D. (2013). Klasifikasi Ditra Buah Jeruk Kintamani Berdasarkan Fitur Warna dan Ukuran Menggunakan Pendekatan Euclidean Distance. *Kumpulan Artikel Mahasiswa Pendidikan Teknik Informatika (KARMAPATI)*, 2(1), 21-274.

- Gurucharan, M. (2022, Juli 28). *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network*. Retrieved 10 28, 2022, from upGrad: <https://www.upgrad.com/blog/basic-cnn-architecture/>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction 2nd edition*. New York: Springer.
- He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2018). Mask R-CNN. *Computer Vision and Pattern Recognition (cs.CV)*, 1-12.
- Hui, J. (2018, Maret 18). *Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3*. Retrieved November 5, 2022, from Medium: <https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>
- Hui, J. (2018, Maret 14). *SSD object detection: Single Shot MultiBox Detector for real-time processing*. Retrieved November 5, 2022, from Medium: <https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06>
- Id, I. D. (2021). *Machine Learning : Teori, Studi Kasus dan Implementasi Menggunakan Python*. Pekanbaru: UR Press.
- Ikeuchi, K. (Ed.). (2021). *Computer Vision: A Reference Guide (Second Edition ed.)*. Redmond: Springer Nature Switzerland AG.
- Iriyanto, S. Y., & Zaini, T. M. (2014). *Pengolahan Citra Digital*. Bandar Lampung: Anugrah Utama Raharja (AURA).

Jabbar, M. A., Prasad, K. M., Peng, S.-L., Reaz, M. B., & Madureira, A. (Eds.). (2021). *Machine Learning Methods for Signal, Image and Speech Processing*. Denmark: River Publishers.

Jähne, B. (2005). *Digital Image Processing 6th revised and extended edition*. Heidelberg: Springer.

Jiang, X., Hadid, A., Pang, Y., Granger, E., & Feng, X. (Eds.). (2019). *Deep Learning in Object Detection and Recognition*. Beach Road.

K, D. (2019, Juni 14). *Top 4 advantages and disadvantages of Support Vector Machine or SVM*. Retrieved 10 27, 2022, from Medium: <https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107>

K, D. (2019, Mei 27). *Top 5 advantages and disadvantages of Decision Tree Algorithm*. Retrieved 10 26, 2022, from Medium: <https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a>

Kadir, A., & Susanto, A. (2012). *Pengolahan Citra : Teori dan Aplikasinya*. Yogyakarta: Yogyakarta Andi.

Kapur, S. (2017). *Computer Vision with Python 3 : Image classification, object detection, video processing, amd more*. Birmingham: Packt Publishing Ltd.

- Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*, 5455–5516.
- Khan, S., Rahmani, H., Shah, S. A., & Bennamoun, M. (2018). *A Guide to Convolutional Neural for Computer Vision*. California: Morgan & Claypool Publisher.
- Khandelwal, R. (2019, Juli 31). *Computer Vision: Instance Segmentation with Mask R-CNN*. Retrieved November 5, 2022, from Towards Data Science: <https://towardsdatascience.com/computer-vision-instance-segmentation-with-mask-r-cnn-7983502fcad1>
- Kou, A., Ganju, S., & Kasam, M. (2019). *Practical Deep Learning for Cloud, Mobile, and Edge : Real-World AI and Computer-Vision Projects Using Python, Keras, and TensorFlow*. California: O'Reilly Media, Inc.
- Kristiani, R., & Puspitorini, A. (2017). Tata Rias Korektif untuk Warna Kulit Gelap pada Pengantin Bridal. *Jurnal Tata Rias*, 06(2), 80-85.
- Lakshmanan, V., Görner, M., & Gillard, R. (2021). *Practical Machine Learning for Computer Vision : End-to-End Machine Learning for Images*. California: O'Reilly Media, Inc.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal Loss for Dense Object Detection. *Computer Vision and Pattern Recognition (cs.CV)*, 1-10.

- Liu, M., Bao, X., & Pang, L. (2019). Image Recognition and Classification based on Elastic Model and BOF Algorithm. *International Journal of Performability Engineering*, 2794-2804.
- Lohia, A., Kadam, K., Joshi, R., & Bongale, D. A. (2021). Bibliometric Analysis of One-stage and Two-stage Object Detection. *Library Philosophy and Practice*, 4910.
- Long, X., Deng, K., Wang, G., Zhang, Y., & dkk. (2020). PP-YOLO: An Effective and Efficient Implementation of Object Detector. *Computer Vision and Pattern Recognition (cs.CV)*, 1-8.
- López, C. A., Bhonsale, S., Peeters, K., & Impe, J. F. (2020, November 27). Manifold Learning and Clustering for Automated Phase Identification and Alignment in Data Driven Modeling of Batch Processes. *frontiers in Chemical Engineering*, 1-18.
- Lu, L., Zheng, Y., & Gustavo Carneiro, L. Y. (2017). *Deep Learning and Convolutional Neural Networks for Medical Image Computing : Precision Medicine, High Performance and Large-Scale Datasets*. New York: Springer Cham.
- Lusiana, Veronica. 2019. "Deteksi Tepi Pada Citra Digital Menggunakan Metode Kirsch Dan Robinson." *Jurnal Teknologi Informasi DINAMIK* 18(2): 182-89.
- Lu, X., Li, Q., Li, B., & Yan, J. (2020). MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection. *Computer Vision and Pattern Recognition (cs.CV)*, 1-16.

Manchanda, S., & Sharma, S. (2016). Analysis of Computer Vision based Techniques for Motion Detection. *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)* (pp. 445-450). Uttar Pradesh: IEEE.

Mandal, K., Sen, A., Chakraborty, A., Roy, S., Batabyal, S., & Bandyopadhyay, S. (2011). Road Traffic Congestion Monitoring and Measurement using Active RFID and GSM Technology. *2011 14th International IEEE Conference on Intelligent Transportation Systems*, (pp. 1375-1379). Washington DC.

Meißner, P. (2018). *Indoor Scene Recognition by 3-D Object Search: For Robot Programming by Demonstration*. Cham: Springer.

MLNerds. (2019, Fbruary Kamis). *How does KNN algorithm work ? What are the advantages and disadvantages of KNN?* Retrieved Oktober Sabtu, 2022, from Machine Learning Interviews : Tools to crack your data science Interviews: <https://machinelearninginterview.com/topics/machine-learning/how-does-knn-algorithm-work-what-are-the-advantages-and-disadvantages-of-knn/>

MLNerds. (2019, Februari 14). *How does KNN algorithm work? What are the advantages and disadvantages of KNN?* Retrieved Oktober 27, 2022, from Machine Learning Interviews : Tools to crack your data science Interviews: <https://machinelearninginterview.com/topics/machine-learning/how-does-knn-algorithm-work-what-are-the-advantages-and-disadvantages-of-knn/>

- Mohanadasan, S. (2020, April 2). *Visual Search Guide: Who Uses It, Benefits, and Optimization Tips*. Retrieved November 1, 2022, from Semrush Blog: <https://www.semrush.com/blog/visual-search-guide-benefits-optimization/#header3>
- Odemakinde, E. (2022). *Everything about Mask R-CNN: A Beginner's Guide*. Retrieved November 5, 2022, from viso.ai: <https://viso.ai/deep-learning/mask-r-cnn/>
- Oktafanda, E. (2022). Klasifikasi Citra Kualitas Bibit dalam Meningkatkan Produksi Kelapa Sawit Menggunakan Metode Convolutional Neural Network (CNN). *Jurnal Informatika Ekonomi Bisnis*, 4(3), 72-77.
- Patel, R., & Patel, S. (2020). A Comprehensive Study of Applying Convolutional Neural Network for Computer Vision. *International Journal of Advanced Science and Technology*, 2161-2174.
- pawangfg. (2020, Maret 1). *R-CNN vs Fast R-CNN vs Faster R-CNN | ML*. Retrieved November 5, 2022, from GeekSforGeeks: <https://www.geeksforgeeks.org/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-ml/>
- Peryanto, A., Yudhana, A., & Umar, R. (2019). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. *Jurnal Format*, 8(2), 138-147.

Poonkuntran, S., Dhanraj, R. K., & Balusamy, B. (Eds.). (2022). *Object Detection with Deep Learning Models: Principles and Applications*. Boca Raton: CRC Press.

Prabowo, D. A., Abdullah, D., & Manik, A. (2018). Deteksi Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking. *Jurnal Pseudocode*, *V*(2), 85-91.

Prakash, K. B. (2021). *Programming with TensorFlow: Solution for Edge Computing Applications*. (K. B. Prakash, & G. R. Kanagachidambaresan, Eds.) Cham: Springer Nature Switzerland AG.

Putri, S. R., & Sulistyaningrum, D. R. (2020). Penerapan Metode Color Invariant untuk Penghapusan Bayangan pada Citra Digital. *Jurnal Sains dan Seni ITS*, *9*(2), A6-A11.

Rahmadwati. (2014). Sistem Diagnosis Kanker Servik Berdasarkan Karakteristik Morfologia. *Jurnal EECCIS*, *7*(2), 191-196.

Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. *Computer Vision and Pattern Recognition (cs.CV)*, 1-9.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Computer Vision and Pattern Recognition (cs.CV)*, 1-10.

Rohan, A., Rabah, M., & Kim, S. (2019). Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2. *IEEE Access*, 1-1.

- Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Trans Pattern Analysis and Machine Intelligence*, 1(20), 23-28.
- Russell, R. (2018). *Machine Learning : Step-by-Step Guide To Implement Machine Learning Algorithms with Python*. California: CreateSpace Independent Publishing Platform.
- Santos, M., & Oliveira, L. (2018). Iterative over-Segmentation via Edge Clustering. *ISEC*, 1-30.
- Sawhney, P. (2021, September 13). *Introduction to Object Detection for Self Driving Cars*. Retrieved November 1, 2022, from Medium: <https://sawhney-prateek97.medium.com/introduction-to-object-detection-for-self-driving-cars-8c4c78b853f9>
- Scikit Learn. (2022). *Manifold Learning*. Retrieved 10 28, 2022, from Scikit-learn Developers: <https://scikit-learn.org/stable/modules/manifold.html#:~:text=Manifold%20learning%20is%20an%20approach,sets%20is%20only%20artificially%20high>.
- Setyaningsih, E. R., & Edy, M. S. (2022, Maret). YOLOv4 dan MaskR-CNN Untuk Deteksi Kerusakan Pada Karung Komoditi. *TEKNIKA*, 11(1), 45-52.
- Shah, D. (2022, Oktober 7). *Mean Average Precision (mAP) Explained: Everything You Need to Know*. Retrieved November 4, 2022, from V7: <https://www.v7labs.com/blog/mean-average-precision#h1>

Shanmugamani, R. (2018). *Deep Learning for Computer Vision: Expert techniques to train advanced networks using TensorFlow and Keras*. Birmingham: Packt Publishing Ltd.

Smeaton, A. F. (2021). Computer Vision for Supporting Image Search (A Compelling Application for Vision Processing). *Advances in Visual Informatics*, 1-10.

Solem, J. E. (2012). *Programming Computer Vision with Python*. California: O'Reilly Media, Inc.

Sonawane, A. (2018, Maret 29). *YOLOv3: A Huge Improvement*. Retrieved November 5, 2022, from Medium: <https://sonawaneanand.medium.com/yolo3-a-huge-improvement-2bc4e6fc44c5>

Sultana, S. I. (2020, Desember 21). *HOW THE HIERARCHICAL CLUSTERING ALGORITHM WORKS*. Retrieved 10 28, 2020, from Dataaspirant: <https://dataaspirant.com/hierarchical-clustering-algorithm/#:~:text=The%20advantage%20of%20Hierarchical%20Clustering,not%20suitable%20for%20large%20datasets>.

Supeshala, C. (2002, Agustus 23). *YOLO v4 or YOLO v5 or PP-YOLO?* Retrieved November 5, 2022, from Medium: <https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109>

Supiyanto, & Suparwati, T. (2021, Maret). Perbaikan Citra Menggunakan Metode Contrast Stretching. *Jurnal Siger Matematika*, 02(01), 13-18.

Szeliski, R. (2022). *Computer Vision : Algorithms and Applications 2nd Edition*. Washington: Springer.

Tan, L., Huangfu, T., Wu, L., & dkk. (2021). Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. *BMC Med Inform Decis Mak*, 21(324), 1-11.

Thatte, A. V. (2020, Mei 21). *Evolution of YOLO – YOLO version 1*. Retrieved November 5, 2022, from Towards Data Science: <https://towardsdatascience.com/evolution-of-yolo-yolo-version-1-afb8af302bd2>

Tsang, S.-H. (2018, Oktober 21). *Review: R-FCN – Positive-Sensitive Score Maps (Object Detection)*. Retrieved November 5, 2022, from Towards Data Science: <https://towardsdatascience.com/review-r-fcn-positive-sensitive-score-maps-object-detection-91cd2389345c>

Tsang, S.-H. (2018, November 3). *Review: SSD – Single Shot Detector (Object Detection)*. Retrieved November 5, 2022, from Towards Data Science: <https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11>

Verma, Y. (2021, September 10). *R-CNN vs Fast R-CNN vs Faster R-CNN – A Comparative Guide*. Retrieved November 5, 2022, from AIM: <https://analyticsindiamag.com/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-a-comparative-guide/>

Viras, S. (2019, Februari 17). *How Computer Vision Detects Motion: A simple explanation of how OpenCV detects motion in videos*. Retrieved November 1, 2022, from Medium:

<https://sehulviras.medium.com/how-computer-vision-detects-motion-67a789349fc8>

Wang, Y., & Ye, J. (2020, Oktober 29). An Overview Of 3D Object Detection. *Computer Vision and Pattern Recognition*.

Wen, S. (2019, November 2). *Object Detection with YOLO | Bringing Vision to Self-Driving Cars*. Retrieved November 1, 2022, from Towards Data Science: <https://towardsdatascience.com/object-detection-with-yolo-bringing-vision-to-self-driving-cars-980295226830>

Williams, G. (2018, Juni 16). *ML In Visual Search*. Retrieved November 1, 2022, from Medium: <https://medium.com/gsi-technology/ml-in-visual-search-part-i-d54cf4f2b509>

Yang, F., Sun, Q., Jin, H., & Zhou, Z. (2020). Superpixel Segmentation With Fully Convolutional Networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13964-13973). CVPR Virtual.

GLOSARIUM

A

- Akurasi : Sejauh mana data sesuai dengan entitas aktual yang diukur atau digambarkan.
- Algoritma : Sebuah proses atau seperangkat aturan yang harus diikuti untuk tugas tertentu.

B

- Bias : Preferensi sadar atau bawah sadar yang mendukung atau berlawanan dengan seseorang, sekelompok orang, atau hal tertentu.
- Big data : *Dataset* yang besar dan kompleks di mana biasanya mencakup jangka waktu panjang, yang memungkinkan analisis data untuk mengatasi masalah bisnis dengan jangkauan yang luas.
- Binning* : Teknik pra-pemrosesan data yang digunakan untuk mengurangi efek kesalahan pengamatan kecil. Nilai data asli yang masuk ke dalam interval kecil tertentu.
- Boolean* : Suatu tipe data yang hanya memiliki dua kemungkinan nilai, biasanya benar atau salah

C

- Centroid* : Pusat geometris atau pusat gambar, dari bangun datar atau bangun padat adalah posisi rata-rata aritmatika dari semua titik di permukaan gambar.
- Cloud* : Tempat untuk menyimpan data secara *online*, bukannya di *hard drive* komputer.

D

- Data : Sekumpulan fakta.
- Database* (basis data) : Sekumpulan data yang disimpan di dalam sistem komputer.

D	
Dataset	: Kumpulan data yang dapat dimanipulasi atau dianalisis sebagai satuan unit.
Data model	: Alat untuk mengatur elemen data dan bagaimana mereka berhubungan satu sama lain.
E	:
F	:
G	:
H	:
I	:
<i>Input</i>	: Data yang masuk ke dalam sebuah sistem
J	:
K	:
L	:
M	:
N	:
O	:
<i>Open data</i>	: Data yang tersedia untuk umum.
Operator	: Simbol yang menamai operasi atau penghitungan yang akan dilakukan.
P	:
Parameter	: Ukuran suatu keadaan secara relatif, saat mengevaluasi kinerja, status, kondisi, dll.
<i>Pixel</i>	: Dalam pencitraan digital, area iluminasi kecil pada layar tampilan yang, bila

dikombinasikan dengan area lain yang berdekatan, membentuk gambar digital.

Q

R

Record : Kumpulan data terkait dalam suatu tabel data, biasanya disebut juga dengan baris.

S

Sample : Segmen dari populasi yang mewakili seluruh populasi.

Semantik : Konsep dan proses *understanding* dan *recognizing* citra pada level skala piksel.

Small data : Data dalam jumlah kecil dan spesifik, yang biasanya dikumpulkan dalam waktu singkat, yang berguna untuk membuat keputusan sehari-hari.

Split : Fungsi yang membagi teks di sekitar karakter tertentu dan menempatkan setiap fragmen ke dalam sel baru yang terpisah.

T

Tipe Data : Atribut yang menggambarkan sekeping data berdasarkan nilai-nilai, bahasa pemrograman, atau operasi yang dapat dilakukannya.

U

Unik : Nilai yang tidak dapat memiliki duplikat.

V

W

X

Y

Z

PROFIL PENULIS



Dr. Arnita, M.Si. lahir di Pangkalan Berandan, 21 Juni 1976. Memperoleh gelar sarjana dari Jurusan Matematika Universitas Sumatera Utara pada Tahun 1999. Memperoleh Gelar Master dari Program Studi Statistika Institut Pertanian Bogor pada tahun 2005 dan Memperoleh gelar Doktor dari program Studi Matematika Universitas Sumatera Utara. Saat ini penulis tercatat sebagai dosen di prodi Ilmu Komputer Universitas Negeri Medan sejak Tahun 2008 hingga sekarang. Beberapa karya buku yang sudah pernah dihasilkan diantaranya adalah Pengantar Statistika tahun 2013, Teknik Peraman Data Time Series tahun 2020, Pengantar Statistika Terapan tahun 2020, Pengenalan Data Mining tahun 2021 dan Teori Graph dan Aplikasinya tahun 2021. Beberapa artikel hasil penelitian terkait *data mining* dan statistika sudah dihasilkan dan dipublikasi baik di jurnal serta prosiding nasional juga internasional bereputasi.



Fitrihuda Aulia lahir di Medan, 29 Desember 2000, saat ini sedang melanjutkan pendidikan S-1 di Universitas Negeri Medan (UNIMED) Fakultas MIPA. Adapun program studi yang diambil adalah ilmu komputer.



Rinjani Cyra Nabila lahir di Binjai, 19 Mei 2001, saat ini sedang menjalani kuliah untuk memperoleh gelar sarjana dari program studi Ilmu Komputer jurusan Matematika pada tahun 2018 hingga saat ini. Penulis pernah mengikuti program pelatihan analisis data pada maret tahun 2022.



Nita Suryani lahir di Ujung-Tanjung, 17 Juni 2000. Pernah mengikuti program pelatihan analisis data pada maret tahun 2022. Saat ini sedang menjalani kuliah untuk mendapat gelar sarjana dari Jurusan Matematika Universitas Negeri Medan dari mulai Tahun 2018 hingga saat ini.