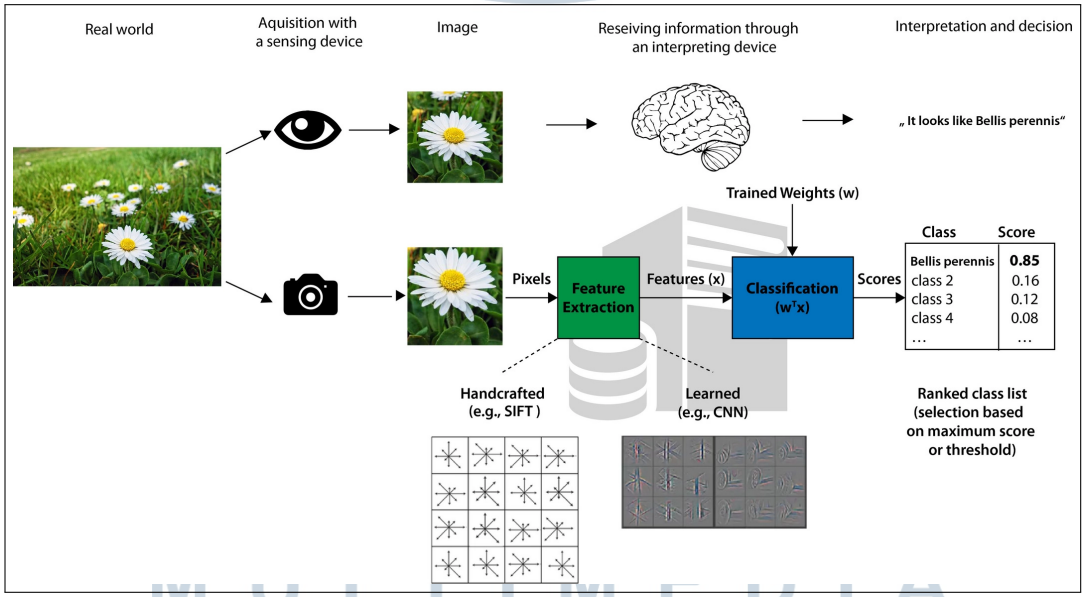
BAB 2
LANDASAN TEORI

2.1 Machine Learning

Machine learning adalah sebuah bentuk dari kecerdasan buatan yang dapat melakukan pekerjaan tanpa harus secara spesifik diprogram untuk menyelesaikan pekerjaan tersebut [1]. Sebaliknya, mesin belajar dari contoh penyelesaian pekerjaan sebelumnya melaui proses yang disebut *training*. Setelah setelah dilatih, maka hasilnya akan digunakan untuk mengerjakan pekerjaan yang serupa dimasa depan. *Machine learning* sangat berguna untuk melakukan ekstraksi informasi dari data yang sangat banyak dan terus bertambah dan sangat berguna untuk menganalisa data yang sulit dibentuk model analitikalnya seperti mengalanalisa gambar dan *video* [1].

Ranah *computer science* yang mempelajari pemahaman dan *insight* dari gambar dan vidio digital disebut dengan *computer vision* [1]. *Computer vision* terbagi menjadi 2 fase, yaitu *feature extraction* dan *classification*.



Gambar 2.1. Tipikal manusia dan *computer vision* dalam melakukan identifikasi [1]

MULTIMEDIA NUSANTARA

2.1.1 Feature Extraction

Feature Extraction adalah fase dimana data mentah akan diubah menjadi info representatif untuk fase klasifikasi. Gambar merupakan susunan ribuan piksel dimana masing-masing piksel tersebut memiliki informasi warna tersendiri [1]. Pada fase ini, piksel-piksel tersebut akan dikurangi dan direpresentasikan menjadi sebuah informasi relevan untuk digunakan dalam fase klasifikasi dalam bentuk informasi tekstur, warna, dan bentuk.

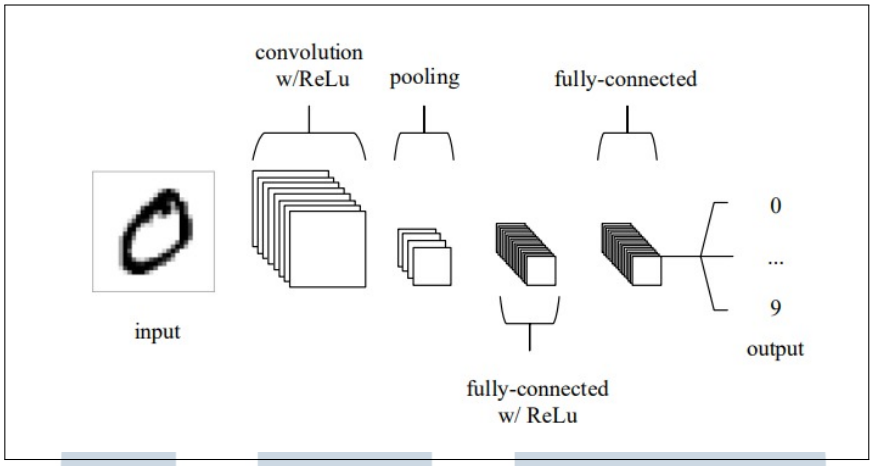
2.1.2 Classification

Classification adalah fase dimana hasil fitur yang telah diekstrak akan diklasifikasikan untuk mengidentifikasi objek yang ada pada gambar. Hasil dari *feature extraction* biasanya berbentuk *vector* dimana akan *vector* tersebut akan dipetakan menjadi *score of confidence* dengan menggunakan *classifier* [1]. Hasil *score* dari fase klasifikasi ini berbeda-beda tergantung pada sistem yang dibangun. Hasilnya dapat berupa pengidentifikasian apakah terdapat objek pada gambar atau tidak hingga membedakan objek yang terdapat pada gambar. Metode pengklasifikasian ini menggunakan algoritma dari *machine learning* seperti *Random Forest* atau *Artificial Neural Network* seperti CNN.

2.1.3 Convolutional Neural Network

CNN adalah sistem prosesing komputasi yang dapat bekerja selayaknya sistem saraf pada makhluk hidup [2]. CNN sendiri merupakan pengembangan langsung dari *Artificial Neural Network* dimana CNN terdiri dari neuron yang dapat belajar mengoptimisasi dirinya sendiri. CNN terdiri dari 3 *layer* utama, yaitu *convolutional layers*, *pooling layers*, dan *fully-connected layers*. Pada Gambar 2.2, cara kerja dari CNN dapat terbagi menjadi 4 bagian, yaitu: *input layer*, *convolutional layers*, *pooling layers*, dan *fully-connected layers*.

Input layer berguna untuk menampung nilai piksel dari gambar. Lalu *convolutional layer* akan menentukan output dari neuron yang tersambung dengan perhitungan skalar. *Pooling layer* akan mengurangi jumlah parameter yang teraktivasi untuk melakukan *downsampling* dari input yang diberikan. Terakhir, *fully-connected layers* akan memproduksi *class scores* untuk memberikan hasil klasifikasi [2].

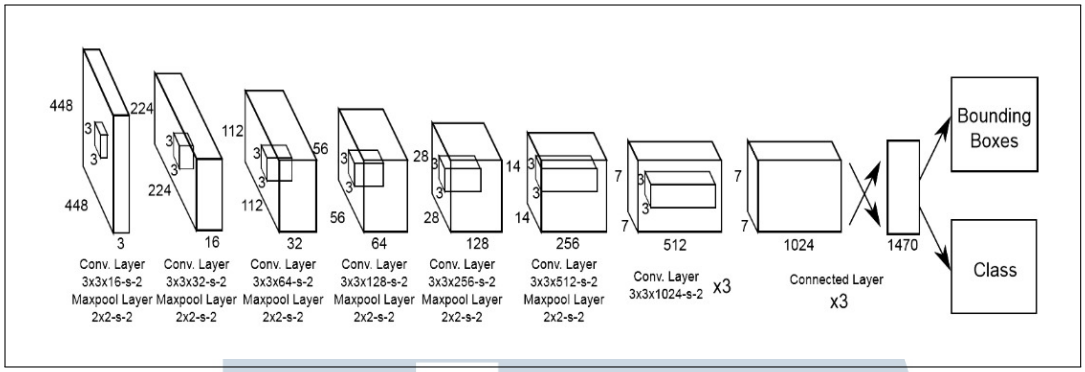


Gambar 2.2. Arsitektur CNN [2]

Dibalik keunggulan dan keuntungan CNN dalam melakukan pendeteksian objek, CNN juga memiliki kekurangan karena membutuhkan *resource* yang cukup besar. Jika gambar memiliki ukuran besar seperti contohnya 227x227, maka *memory resource* yang dibutuhkan adalah sekitar 70 megabytes untuk 1 gambar [2]. Dengan pemakaian *resource* yang cukup besar ini, jika terdapat gambar dengan ukuran lebih besar dari 128x128, maka gambar mentahnya harus dilakukan *resizing* agar *resource* yang digunakan bisa berkurang.

2.2 You Only Look Once (YOLO)

YOLO adalah sebuah *framework* CNN yang telah *layer*-nya telah di *predefined* terlebih dahulu. Arsitektur dari *framework* YOLO terdiri dari 27 *layer* CNN dimana terdapat 24 *convolutional layer* dan 3 *fully-connected layer* [3]. YOLO akan memberikan *bounding box* pada gambar yang akan dideteksi. *Bounding box* adalah sebuah kotak imajiner yang berguna untuk menjadi batas referensi dari *behaviour* yang perlu dipelajari oleh *layer-layer* CNN. Pada Gambar 2.3 terdapat 1 *layer* khusus pada *fully-connected layer* yang menjadi *final detection layer* dimana *layer* tersebutlah yang akan membagi *image input* ke dalam *Bounding Boxes* dan *Class score*.



Gambar 2.3. Arsitektur YOLO [3]

Dalam melakukan pendeteksian objek, YOLO memiliki keuntungan dalam aspek kecepatan dan efisiensi [10]. Karena kecepatan dan efisiensinya, YOLO cocok digunakan dalam *real-time object detection*. Namun YOLO juga memiliki beberapa limitasi sebagai berikut.

1. Sulit mendeteksi objek dengan ukuran kecil.
2. Sulit mendeteksi objek dengan ukuran yang berbeda dibandingkan objek sekitarnya.
3. Sensitif pada kondisi pencahayaan lingkungan.
4. Membutuhkan resource yang banyak sehingga akan berat dijalankan secara *real-time* pada perangkat seperti gawai.

2.3 Confusion Matrix

Confusion matrix adalah *layout* tabel yang memvisualisasikan performa dari hasil aktual dan hasil algoritma klasifikasi dalam model[11]. *Confusion matrix* memiliki 2 dimensi dimana 1 dimensi menunjukan hasil aktual dan dimensi lainnya menunjukan hasil prediksi algoritma klasifikasi model.

UNIVERSITAS MULTIMEDIA NUSANTARA

		Prediction outcome	
		p	n
Actual Value	p′	True Positive	False Negative
	n′	False Positive	True Negative

Tabel 2.1. Confusion Matrix

Pada Tabel 2.1 menunjukan bahwa nilai *True Positive* adalah nilai dimana hasil prediksi benar bahwa sebuah objek bernilai positif, sedangkan nilai *False Positive* adalah nilai dimana hasil prediksi salah bahwa sebuah objek bernilai negatif. Lalu nilai *False Negative* adalah nilai dimana hasil prediksi salah bahwa sebuah objek bernilai negatif dan nilai *True Negative* adalah nilai dimana hasil prediksi benar bahwa sebuah objek bernilai negatif.

2.3.1 Precision

Precision adalah tingkat keakuratan model dalam memprediksi nilai positif[12]. Nilai *precision* didapatkan dengan membandingkan nilai positif dengan keseluruhan prediksi positif sebuah objek. Nilai *precision* diperoleh melalui rumus 2.1.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

2.3.2 Recall

Recall adalah seberapa sering model berhasil memprediksi nilai positif jika nilai aktualnya positif[12]. Nilai *recall* diperoleh melalui rumus 2.2.

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

2.3.3 Accuracy

Accuracy adalah seberapa sering model berhasil memprediksi dengan benar dibandingkan dengan total hasil prediksi dari model klasifikasi[12]. Nilai *accuracy* diperoleh melalui rumus 4.3.

$$Recall = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.3}$$

2.3.4 Intersection over Union (IoU)

Intersection over Union (IoU) adalah nilai dari 2 area yang memiliki kesamaan di beberapa area[13]. 2 area yang memiliki kesamaan untuk mendapatkan nilai IoU adalah *bounding box* prediksi dan *bounding box* sebenarnya. IoU bisa didapatkan melalui rumus 2.4.

$$IoU = \frac{areaofoverlap}{areaofunion} \tag{2.4}$$

2.3.5 Mean Average Precision (mAP)

Mean average precision (mAP) adalah nilai rata-rata dari nilai AP dari setiap kelas[14]. Nilai AP didapatkan dari nilai *recall* dan *precision* melalui rumus 2.5 dengan R_n sebagai nilai *recall* dan P_n sebagai nilai *precision* pada iterasi n.

$$AP = \sum_{n}(R_n - R_{n-1})P_n \tag{2.5}$$

YOLOv7 menggunakan variasi mAP@0.5 dan mAP@0.5:0.95. mAP@n menghitung nilai mAP dengan membandingkan nilai *ground truth* dengan nilai IoU sama dengan n agar dapat terindikasi bahwa prediksi kelas benar[15]. Jadi mAP@0.5 menghitung nilai mAP dengan membandingkan nilai *ground truth* dengan nilai IoU 0.5 dan mAP@0.5:0.95 menghitung nilai mAP dengan membandingkan nilai *ground truth* dengan nilai IoU 0.5 hingga 0.95 agar dapat terindikasi bahwa prediksi kelas benar.