

BAB II STUDI LITERATUR

2.1 Privasi

Privasi menurut KBBI adalah kebebasan; keleluasaan pribadi. Secara makna privasi adalah kondisi di mana seseorang maupun sekelompok tidak ingin diganggu dari publik, atau untuk mengontrol arus informasi mengenai diri mereka. Privasi merupakan hal yang sangat penting bagi individu karena pada dasarnya seseorang pasti memiliki sisi diri yang tidak ingin diketahui orang lain dan akan ada keinginan dari individu tersebut untuk melindungi rahasia dirinya [1].

2.2 *Face Recognition*

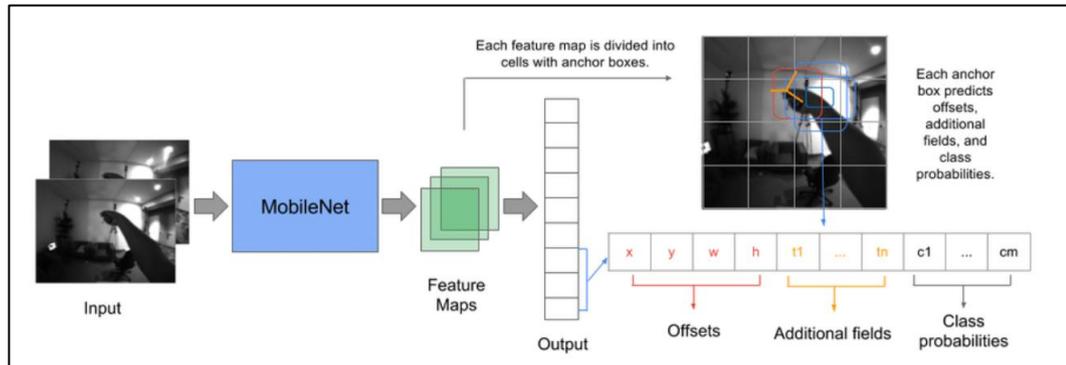
Pengenalan wajah merupakan sebuah sistem identifikasi pribadi yang menggunakan karakteristik wajah seseorang [6]. Pengenalan wajah menjadi salah satu bidang penelitian paling aktif dari *computer vision* [7], dan telah berhasil diterapkan di berbagai bidang dengan bantuan teknologi komputer dan teknologi jaringan [8]. *Face recognition* menjadi dasar pada penelitian ini karena program diharuskan untuk mengenali wajah pemilik laptop sehingga fitur *privacy guard* dapat diimplementasikan. Pada penelitian ini metode MobileNet-SSD digunakan untuk mengimplementasikan *face recognition*. Sedangkan untuk *learning* menggunakan metode *transfer learning*.

2.2.1 MobileNet

MobileNet merupakan arsitektur *convolutional neural network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan *computer vision* yang biasanya menggunakan *resource* yang tinggi. MobileNet digunakan pada penelitian ini mempertimbangkan *resource* yang hanya dibebankan kepada CPU sehingga diharapkan mampu memberikan performa yang baik dalam melakukan prediksi. Jaringan ini dapat mengurangi jumlah parameter tanpa mengorbankan akurasi [9].

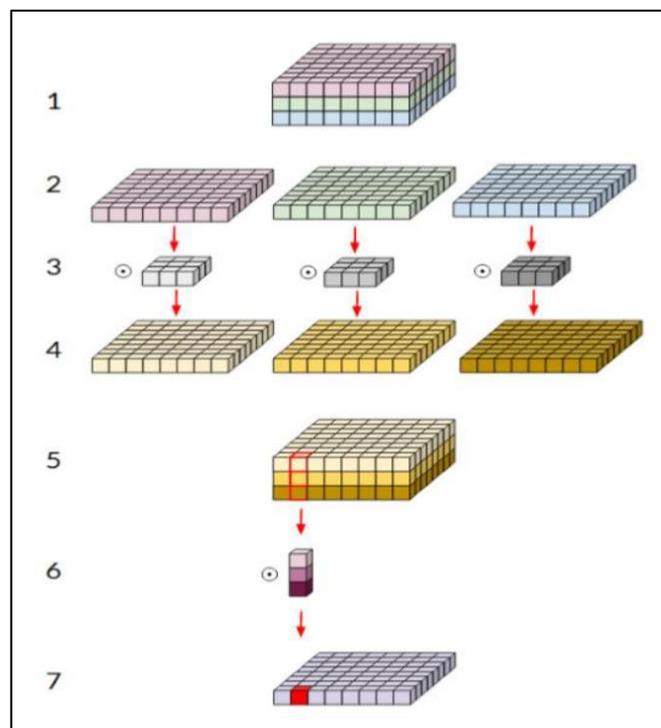
Arsitektur MobileNet membagi konvolusi menjadi dua yaitu *depthwise convolution* dan *pointwise convolution*. MobileNet memanfaatkan *batch normalization* (BN) dan *rectified-linear unit* (ReLU) untuk diterapkan pada *depthwise convolution* dan *pointwise convolution*. MobileNet dibangun di atas arsitektur jaringan yang efisien

dengan menggunakan konvolusi yang dapat dipisahkan secara mendalam untuk menghasilkan *deep neural network* yang ringan [10]. Alur kerja MobileNet dapat dilihat pada Gambar 2.1.



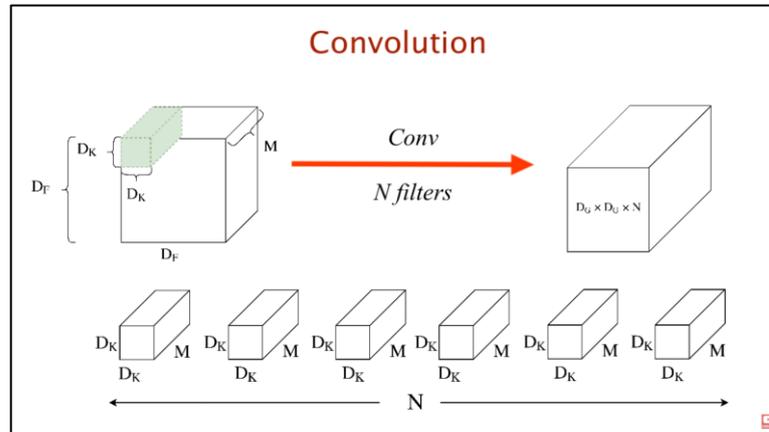
Gambar 2.1 Alur kerja MobileNet

Depthwise separable convolution (DSC) menggantikan konvolusi standar dengan 2 tahap operasi. Pertama *depthwise convolution* dimana setiap filter hanya melakukan proses terhadap sebuah *feature map input* secara mendalam. Kedua *pointwise convolution* yang merupakan 1×1 *convolution layer* yang digunakan untuk menggabungkan jalur informasi dari *depthwise layer*. Ilustrasi DSC dapat dilihat pada Gambar 2.2



Gambar 2.2 Ilustrasi *depthwise separable convolution*

DSC menyebabkan jalur konvolusi menjadi jauh lebih efisien dengan menggunakan parameter yang jauh lebih sedikit dibandingkan konvolusi tradisional/biasa. Pada gambar 2.3 merupakan ilustrasi dari konvolusi tradisional.

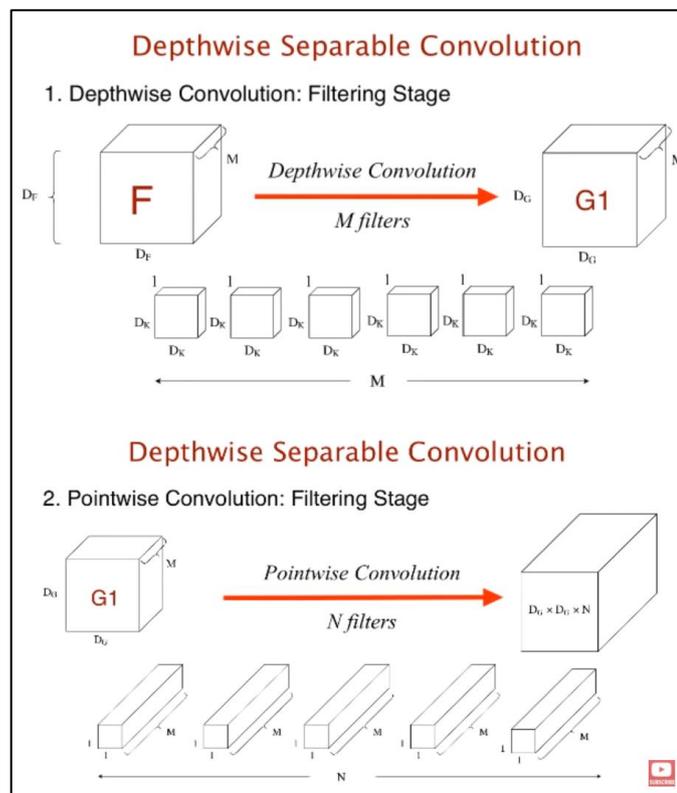


Gambar 2.3 Ilustrasi perhitungan *traditional convolution*

Berdasarkan Gambar 2.3 maka memiliki jumlah komputasi sebesar persamaan 2.1

$$D_K \times D_K \times M \times D_G \times D_G \times N \quad (2.1)$$

Sedangkan untuk DSC ilustrasi perhitungannya dapat dilihat pada Gambar 2.4.



Gambar 2.4 Ilustrasi perhitungan *depthwise separable convolution*

Berdasarkan Gambar 2.4 maka terbentuk persamaan 2.2 untuk menghitung jumlah komputasi DSC.

$$D_K \times D_K \times M \times D_G \times D_G + N \times D_G \times D_G \times M \quad (2.2)$$

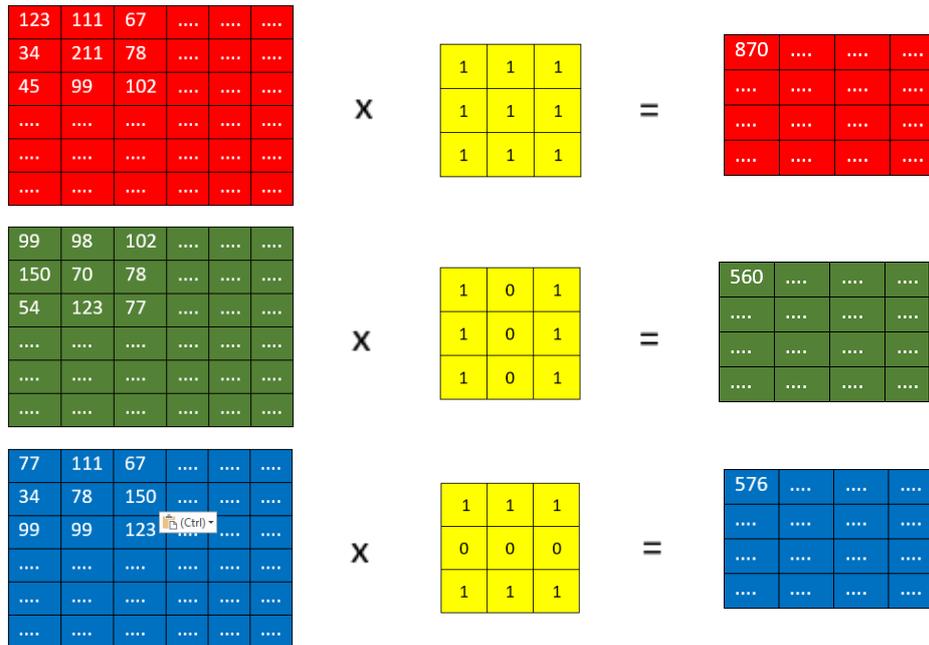
Jika persamaan 2.2 dibandingkan dengan persamaan 2.1 maka didapatkan persamaan 2.3.

$$\frac{1}{D_K^2} + \frac{1}{N} \quad (2.3)$$

Apabila jika $D_K^2 = 3$ dan $N = 1024$ maka DSC memiliki komputasi 9 kali lebih efisien dibandingkan dengan konvolusi tradisional. Tahapan dalam MobileNet tidak jauh berbeda dengan CNN, beberapa tahapan tersebut adalah sebagai berikut:

1. Depthwise Convolution

pada *depthwise convolution* setiap *channel* akan dikalikan dengan filter yang berbeda-beda.



Gambar 2.5 Ilustrasi *depthwise convolution*

Perhitungan *channel red*

$$123 + 111 + 67 + 34 + 211 + 78 + 45 + 99 + 102 = 870$$

Perhitungan *channel green*

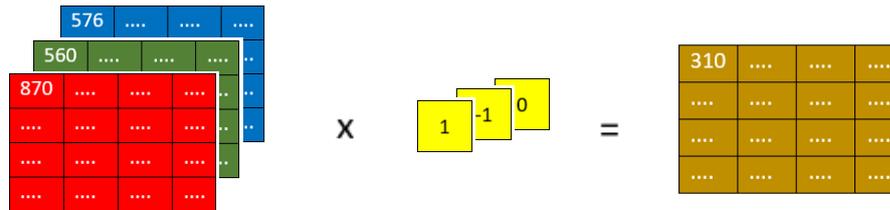
$$99 + 98 + 102 + 150 + 70 + 78 + 54 + 123 + 77 = 560$$

Perhitungan *channel blue*

$$77 + 111 + 67 + 34 + 78 + 150 + 99 + 99 + 123 = 576$$

2. *Pointwise Convolution*

Pointwise convolution akan mengkonvolusi lagi hasil dari *deptwise* dengan filter 1x1, jumlah filter sendiri dapat ditentukan sesuai kebutuhan sehingga menghasilkan jumlah *channel* sesuai dengan jumlah filter.



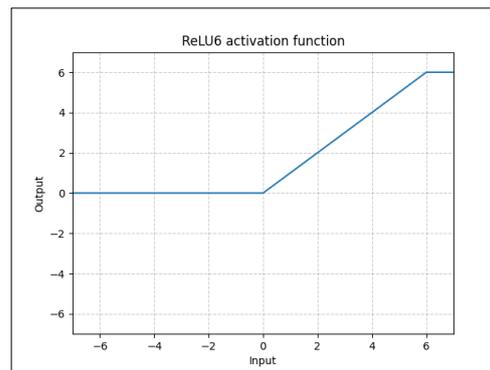
Gambar 2.6 Ilustrasi *pointwise convolution*

Hasil di atas dapat diilustrasikan dengan perhitungan berikut:

$$(870 \times 1) + (560 \times (-1)) + 576 \times 0 = 310$$

3. *Relu6*

Relu6 adalah salah satu fungsi aktivasi yang digunakan dalam MobileNet untuk mengubah nilai-nilai *feature map* pada *range* tertentu sehingga menghasilkan *feature map* yang baru. *Relu6* memiliki arti nilai *output* maksimum adalah 6 sehingga apabila *input* melebihi 6 maka akan diubah menjadi 6. Sedangkan jika *input* bernilai negatif maka akan diubah menjadi 0. Ilustrasi *relu6* dapat dilihat pada Gambar 2.7 sedangkan persamaan *relu6* dapat ditunjukkan pada persamaan 2.3.

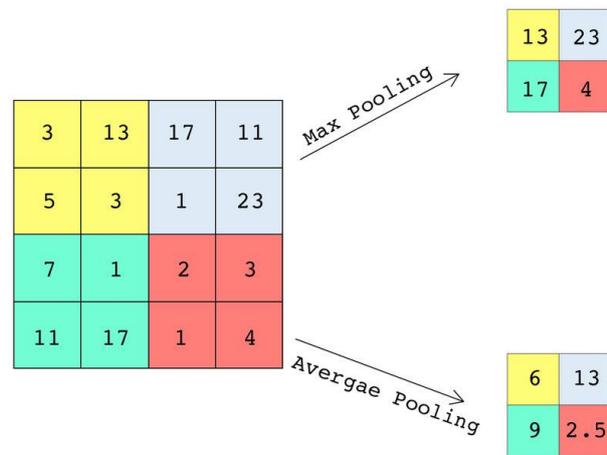


Gambar 2.7 *Relu6*

$$\text{Min}(\text{Max}(0, x), 6) \quad (2.3)$$

4. Pooling

Pooling adalah layer yang berfungsi untuk mengurangi dimensi dari *feature map* tanpa mengurangi informasi penting dari *feature map* tersebut. *Pooling* dibagi menjadi 2 yaitu *average pooling* dan *max pooling*. *Average pooling* adalah hasil dari penjumlahan nilai *feature* dibagi dengan banyaknya *feature* yang dijumlahkan sedangkan *max pooling* adalah nilai maksimum yang diambil dalam *feature map* tersebut. Ilustrasi *pooling* dapat dilihat pada Gambar



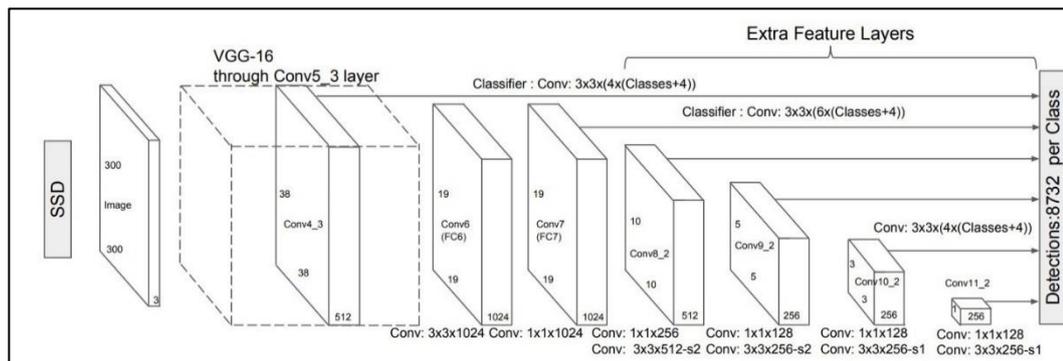
Gambar 2.8 Max pooling dan average pooling

2.2.2 Single Shot Detector

Single Shot Detector adalah sebuah metode untuk mengenali atau mendeteksi sebuah objek pada suatu gambar dengan menggunakan *single deep neural network* dan salah satu algoritma deteksi objek yang paling populer karena kemudahan implementasi, akurasi yang baik vs rasio yang dibutuhkan komputasi [5]. *Single Shot Detector* hanya perlu mengambil satu bidikan tunggal untuk mendeteksi beberapa objek didalam gambar. Metode *Single Shot Detector* ini termasuk kedalam deteksi objek secara *real time*, meskipun lebih intuitif daripada rekan-rekannya seperti R-CNN, *Fast R-CNN* *Faster R-CNN* dan *You Only Look Once (YOLO)*, *Single Shot Detector (SSD)* adalah algoritma yang sangat kuat. Selain itu *Single Shot Detector* memiliki akurasi yang lebih baik [11].

Perbedaan utama antara objek detektor klasik misalnya R-CNN dan SSD adalah bahwa objek detektor klasik menggunakan proposal wilayah sedangkan metode SSD menggunakan data *groundtruth* yang harus ditetapkan agar hasil yang

ditentukan dalam rangkaian hasil detektor yang jelas. Metode SSD melakukan pencocokan terhadap objek menggunakan *default bounding box* dengan melalui berbagai macam skala dan rasio untuk setiap *feature map*. Setiap elemen dari *feature map* mempunyai sejumlah kotak yang saling berhubungan antara lain. Setiap *bounding box* dengan *Intersection over Union (IoU)* lebih besar dari 0,5 dianggap cocok terhadap objek ada pada setiap kategori. Pada Gambar 2.5 merupakan arsitektur dari SSD.



Gambar 2.9 Arsitektur SSD

Mobilenet-SSD terdiri dari SSD yang berperan sebagai *base model*, sedangkan Mobilenet sebagai *network model*. SSD akan mengatur pendeteksian objek dengan membuat *bonding box*. MobileNet akan bekerja untuk mengekstrak fitur yang akan nantinya diklasifikasi. Penggabungan SSD dan MobileNet akan membantu dalam proses pembuatan aplikasi deteksi objek.

2.2.3 Transfer Learning

Transfer learning merupakan suatu metode yang memanfaatkan model yang sudah dilatih terhadap suatu *dataset* untuk menyelesaikan permasalahan lain yang serupa dengan cara menggunakannya sebagai *starting point*, memodifikasi dan meng-*update* parameternya sehingga sesuai dengan *dataset* yang baru.

Neural network dalam biasanya membutuhkan banyak data pelatihan untuk mempelajari representasi data. Jika data pelatihan tidak mencukupi, ada beberapa teknik untuk membantu model jaringan saraf tiruan mempelajari representasi data menggunakan data pelatihan kecil. Salah satu tekniknya adalah mentransfer pengetahuan model jaringan saraf terlatih lainnya ke model kami. Teknik ini dikenal sebagai pembelajaran transfer atau pembelajaran otodidak

2.3 Python

Python adalah bahasa pemrograman yang populer dan tingkat tinggi yang artinya tingkatannya mudah untuk dipelajari oleh manusia [12]. Bahasa pemrograman ini diciptakan oleh Guido van Rossum seseorang warga negara Belanda dan diperkenalkan sejak tahun 1991. Sampai saat ini bahasa pemrograman Python hampir dipakai di segala bidang seperti *artificial intelligence*, *machine learning*, *deep learning*, *game*, sistem berbasis web, dan bahkan dapat membuat mesin pencari sendiri. Jadi secara umum, bahasa pemrograman ini dapat dipakai dalam pengembangan di segala bidang teknologi.

2.4 OpenCV

OpenCV adalah sebuah *library* yang bersifat *open source* (bebas digunakan untuk hal yang bersifat akademis maupun komersial) yang berisi banyak fungsi pemrograman untuk teknologi *computer vision* secara *real time*. *Library* ini ditulis dalam C dan C++ dan berjalan di bawah Linux, Windows dan Mac OS. OpenCV ditulis dalam C yang dioptimalkan dan dapat memanfaatkan prosesor *multicore* [5]. *Library* ini berfokus terutama pada pemrosesan gambar secara *real time*. *Library* ini tersedia untuk bahasa seperti C#, Python, Ruby dan Java (menggunakan JavaCV) telah dikembangkan untuk mendorong adopsi oleh *audiens* yang lebih luas [13].

2.5 Tensorflow

TensorFlow merupakan *machine learning library* yang diluncurkan oleh Google yang dapat digunakan secara internal sebagai program *open source*. Tujuannya adalah untuk membantu orang memperoleh *tool* yang tepat dalam riset, *coding*, maupun pengembangan *machine learning*. TensorFlow adalah *library software open source* untuk komputasi numerik menggunakan grafik aliran data. Ini adalah sistem yang mentransfer struktur data kompleks ke jaringan saraf tiruan untuk analisis dan pemrosesan. Ini dapat digunakan di banyak bidang pembelajaran mendalam seperti pengenalan ucapan dan pengenalan gambar [14].

2.6 Keras

Keras adalah *application programming interface* (API) yang dibangun di atas Tensorflow, yang memberikan kemampuan kepada pengguna untuk mengimplementasikan, melatih, dan menguji jaringan dengan cepat [15]. Saat ini, Keras dianggap sebagai salah satu *library machine learning* terbaik di Python. Keras dianggap terbaik karena menyediakan beberapa utilitas dalam hal menyusun model, memproses *dataset*, maupun memvisualisasikan grafik.

2.7 Pynput

Pynput adalah sebuah *library* dari Python yang berfungsi untuk mengontrol dan *monitoring* sebuah *device input*. Pynput memiliki dua perangkat *input* yang didukung. Pertama adalah `pynput.mouse` yang berfungsi untuk mengontrol dan *monitoring mouse* atau *trackpad*, sedangkan yang kedua adalah `pynput.keyboard` yang berfungsi untuk mengontrol dan *monitoring keyboard*.

2.8 Google Colaboratory

Google Colaboratory (Google Colab) yang secara luas dikenal dengan Colab adalah layanan gratis yang disediakan oleh Google untuk setiap orang yang memiliki akun Gmail [16]. Layanan ini ditujukan untuk mempermudah pendidikan dan penelitian yang berhubungan dengan pembelajaran mesin [17]. Colab menyediakan *runtime* yang sepenuhnya dikonfigurasi untuk *deep learning* dan memberikan akses secara gratis untuk menggunakan GPU yang tangguh.

Dikarenakan layanan ini gratis tentu saja memiliki beberapa batasan sumber daya seperti RAM yang disediakan yaitu sebesar 12GB (sebenarnya 12,7GB namun 0,8GB sudah digunakan). RAM tersebut cukup besar mengingat ini merupakan layanan gratis namun ada beberapa kondisi di mana membutuhkan lebih banyak RAM. Selain itu Colab memiliki batasan ruang *disk* sebesar 108GB di mana hanya 77GB yang tersedia untuk pengguna. Google Colab dapat digunakan selama maksimal 12 jam, yang mana setelah 12 jam *runtime* akan diulang kembali dan pengguna harus memulai pekerjaannya kembali agar dapat melanjutkannya.

2.9 Confusion Matrix

Confusion matrix adalah suatu metode yang biasa digunakan sebagai tolak ukur terhadap performa suatu model pembelajaran. Pada *confusion matrix* terdapat 4 komponen, yaitu:

1. *True Positive (TP)*

True positive merupakan jumlah prediksi positif yang sesuai dengan data aktual yang juga positif.

2. *True Negative (TN)*

true negative merupakan jumlah prediksi negatif yang sesuai dengan data aktual yang juga negatif.

3. *False Positive (FP)*

False Positif merupakan jumlah prediksi positif yang memiliki data aktual negatif.

4. *False Negative (FN)*

False Negative adalah jumlah prediksi negatif yang memiliki data aktual positif.

Untuk lebih jelas mengenai *confusion matrix* dapat dilihat pada Tabel II.1. *Confusion matrix* dapat digunakan untuk mencari nilai *accuracy*, *precision*, dan *recall*. *Accuracy* adalah rasio prediksi benar dengan keseluruhan data. *Precision* adalah prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Recall* adalah rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

Tabel 2.1 Tabel *confusion matrix*

		Nilai Aktual	
		+	-
Nilai Prediksi	+	TP	FP
	-	FN	TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.4)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2.5)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.6)$$

2.10 Studi Pustaka

Penelitian yang berjudul “*A deep learning based solution for construction equipment detection: from development to deployment*” menjelaskan tentang pemanfaatan metode *single shot detector* dengan arsitektur MobileNet untuk mendeteksi peralatan konstruksi secara *realtime*. Tahapan pengembangan pada penelitian tersebut meliputi persiapan data, pemilihan model, pelatihan model, dan evaluasi model. Penelitian ini juga membandingkan hasil pengujian antara GTX 1080, Jetson TX2 dan Raspberry Pi3 di mana Jetson memiliki terbaik. Peninjauan hasil menjelaskan bahwa kinerja MobileNet-SSD cukup baik yaitu dengan tingkat akurasi di atas 90% dan konsisten.

Pada penelitian yang berjudul “*Application of improved mobilenet-ssd on underwater sea cucumber detection robot*” menjelaskan tentang implementasi metode MobileNet-SSD yang sudah di tingkatkan sehingga menjadi MD-SSD untuk mendeteksi teripang dilaut. MD-SSD diciptakan dengan menggabungkan struktur *residual* dan konvolusi dilatasi. Gabungan tersebut dapat mengatasi kurangnya masalah akurasi pada MobileNet-SSD dalam mendeteksi teripang di dalam air dengan hasilnya yaitu dapat mendeteksi dengan akurat secara *realtime* dengan GPU [18].

Pada penelitian yang berjudul “*Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON*” menjelaskan tentang bagaimana mengenali ekspresi kantuk dan menggunakan citranya sebagai pemicu diaktifkannya alarm yang terpasang di dalam kendaraan sehingga dapat mencegah ingin terjadinya kecelakaan akibat mengantuk. Pada penelitian ini menggunakan opencv dan dlib di mana OpenCV sendiri adalah *library* yang dapat dimanfaatkan untuk mengolah citra secara *real time* sedangkan dlib adalah sebuah *detector landmark* yang sudah dilatih sebelumnya di dalam *library* dlib [19].

Pada penelitian yang berjudul “*A Convolutional Neural Network based on TensorFlow for Face Recognition*” menjelaskan tentang bagaimana metode CNN untuk mengenali wajah seseorang dengan menggunakan tensorflow. Pada penelitian ini struktur jaringan dari CNN yang dibuat yaitu dimulai dari *input image, random cropping, convolution feature, convolution feature after sampling, image feature*, dan yang terakhir *identification*. CNN yang dikembangkan pada penelitian ini mendapatkan akurasi yang cukup baik yaitu sebesar 87% [14].

Pada penelitian yang berjudul “*Face Detection and Tracking Using OpenCV*” menjelaskan tentang opencv untuk mendeteksi wajah dengan berbagai algoritma yaitu *haar cascades, camshif algorithm, finding faces via color* dan *finding faces via motion*. Selain itu juga pada penelitian ini membahas tentang perbedaan antara opencv dengan matlab di mana opencv lebih baik secara *speed, probability*, dan *cost*.

Pada penelitian yang berjudul “*Real-Time Face Detection and Recognition in Complex Background*” menjelaskan tentang algoritma yang diimplementasikan menggunakan serangkaian metode pemrosesan sinyal yang meliputi Ada Boost, *cascade classifier, Local Binary Pattern (LBP)*, fitur *Haar-like, facial image pre-processing* dan *Principal Component Analysis (PCA)*. Algoritma Ada Boost diimplementasikan dalam *cascade classifier* untuk melatih detektor wajah dan mata dengan akurasi deteksi yang kuat. Deskriptor LBP digunakan untuk mengekstrak fitur wajah untuk deteksi wajah cepat. Algoritma *Eyes Detection* mengurangi tingkat deteksi wajah palsu. Citra wajah yang terdeteksi kemudian diproses untuk memperbaiki orientasi dan meningkatkan kontras, oleh karena itu akurasi pengenalan wajah tetap tinggi. Terakhir, algoritma PCA digunakan untuk mengenali wajah secara efisien [20].

Beberapa penelitian yang telah dijelaskan secara singkat dan hasilnya dapat dilihat pada Tabel 2.2. Beberapa penelitian tersebut juga yang dijadikan sebagai referensi untuk menentukan metode hingga teknologi yang akan digunakan pada penelitian ini.

Tabel 2.2 Penelitian terkait

No	Peneliti	Masalah	Metode	Hasil
1.	Saeed Arabi (2019) [21]	Mendeteksi dan mengidentifikasi peralatan konstruksi	MobileNet-SSD	Metode Single Shot Detector (SSD) memiliki tingkat akurasi di atas 90% untuk mendeteksi secara <i>realtime</i> .
2.	Yufeng Yao (2019) [18]	Meningkatkan MobileNet-SSD (MD-SSD) pada robot pendeteksi teripang bawah air	MobileNet-SSD	Tingkat akurasi dan <i>recall</i> model MD-SSD mencapai 93,55% dan 92,68%, yaitu 2,76% dan 6,29% lebih tinggi dibandingkan MobileNet-SSD. Kecepatan deteksi lebih tinggi 43,65 <i>frame/s</i> daripada SSD, akurasi deteksi lebih tinggi dari SSD 0,46%.
3.	Farhan Sindy (2019) [10]	Mendeteksi objek manusia secara <i>realtime</i> dengan metode MobileNet-SSD	MobileNet-SSD	Akurasi tertinggi mencapai 91,67% didapatkan pada percobaan ke-4 dengan lingkungan memiliki intensitas cahaya 550 lux dan jarak terhadap

				objek 1-5 meter. Sedangkan untuk akurasi terendah diperoleh oleh percobaan ke-1 dengan nilai 50,89% yang dilakukan pada lingkungan yang memiliki intensitas cahaya 50 lux
4.	Afrizal Zein (2018) [19]	Mengenal ekspresi kantuk dan menggunakan citranya sebagai pemicu diaktifkannya alarm yang terpasang di dalam kendaraan	OpenCV dan DLIB python	Pengujian menunjukkan perolehan tingkat akurasi pendeteksian kantuk mencapai sebesar diatas 90%.
5.	Liping Yuan (2017) [14]	Mengimplementasikan CNN (<i>Convolutional Neural Network</i>) yang berbasis pada tensorflow untuk <i>face recognition</i>	Tensorflow dan CNN	Metode CNN memiliki akurasi sebesar 87% dengan jumlah <i>true positif</i> adalah 87 dan jumlah <i>false positif</i> adalah 22
6.	Kruti Goyal (2017) [13]	Menggunakan OpenCV untuk mendeteksi wajah dengan berbagai algoritma	<i>Opencv, Haar cascades, camshif algorithm, finding faces via color</i> dan	Algoritma <i>haar cascades</i> menjadi yang paling efisien untuk mendeteksi wajah dibandingkan lainnya dan

			<i>finding faces via motion</i>	memberikan akurasi yang baik.
7.	Xin Zhang (2017) [20]	Mengimplementasikan algoritma yang efisien dan kuat untuk deteksi dan pengenalan wajah secara <i>real time</i>	Ada <i>Boost Algorithm, Cascade Classifier, Local Binary Pattern, Haar-Like Features, Principal Component Analysis</i>	Tingkat positif-benar secara keseluruhan sebesar 98,8% untuk deteksi wajah dan 99,2% untuk pengenalan wajah yang benar.