

## **BAB II**

### **DASAR TEORI**

#### **2.1. KAJIAN PUSTAKA**

Pada penelitian yang dilakukan oleh Restin Alfinda Zai dan Fajar Astuti Hermawati dengan judul “Sistem Deteksi Pemakaian Masker Menggunakan Metode *Convolutional Neural Network* (CNN). Penelitian ini membahas tentang bagaimana merancang sistem pendeteksi masker dengan menggunakan CNN sebagai metode dengan latar belakang penelitian ini adalah untuk membantu meredam penyebaran COVID-19 yang telah menimbulkan permasalahan terkait kesehatan, ekonomi, pendidikan dan lain sebagainya. Pada penelitian ini menggunakan dataset berupa citra wajah bermasker dan tidak bermasker dimana dataset akan dibagi menjadi dua bagian yakni *data train* dan *data test* dengan persentase 70% *data train* dan 30% untuk *data test*. Untuk input yang akan digunakan yaitu berupa citra yang diambil dari CCTV untuk 30 orang kategori per kelasnya. Dijelaskan bahwa sistem akan dibagi menjadi dua tahapan utama yakni tahap untuk deteksi lokasi wajah dan juga tahap klasifikasi wajah. Input wajah yang terdiri atas banyak wajah akan dideteksi menggunakan metode *Viola Jones*. Kemudian untuk setiap wajah yang dideteksi akan diklasifikasikan dengan CNN menjadi empat kategori yaitu wajah tanpa masker, wajah bermasker medis, wajah bermasker tidak menggunakan masker medis, dan juga pemakaian masker yang tidak benar. Hasil pengujian dari penelitian ini menunjukkan bahwa saat *proses training dataset* hasil yang didapatkan cukup baik dengan *MiniBatch Size* 40 dan *MaxEpoch* 90 mendapat akurasi 84.23% dengan estimasi waktu *training* 169 menit 26 detik. Untuk pengujian menggunakan input CCTV dengan jumlah video yang digunakan, 40 video dengan jumlah kategori kelas sebanyak 10 video didapatkan hasil data valid 36 dari 40 data wajah yang diuji. Ada beberapa data video yang tidak valid dideteksi, hal ini dipengaruhi oleh kualitas resolusi kamera CCTV dan pencahayaan saat pengujian[5].

Pada penelitian yang dilakukan pada tahun 2020 oleh Samuel Ady Sanjaya dan Suryo Adi Rakhwan dengan judul “Face Mask Detection Using *MobilenetV2* in The Era of COVID-19 Pandemic”. Pada penelitian ini membahas tentang bagaimana

mengembangkan sistem deteksi masker dengan algoritma *machine learning* melalui metode klasifikasi citra *MobileNetV2*. Pada perancangan sistemnya dibagi menjadi beberapa tahapan yang terdiri dari pengumpulan *dataset*, *Pre-processing*, *Split the Data*, *Building the Model*, *Testing the Model*, dan *Implementing the Model*. Untuk dataset yang digunakan terdiri atas data *image* orang bermasker dengan jumlah 1916 dan orang tidak bermasker berjumlah 1930. Pada tahapan *Pre-processing* yang dilakukan adalah mengubah ukuran gambar, mengubah gambar menjadi *array* melakukan proses *input* gambar menggunakan *MobileNetV2*, dan terakhir adalah pemberian label pada gambar. Pembuatan model yang telah dilakukan yang kemudian diuji mendapatkan hasil akurasi tertinggi sekitar 90% dengan menggunakan *epochs* 20. Untuk pengujian sistem secara keseluruhan mendapatkan hasil baik dengan *input* menggunakan kamera akan menampilkan hasil wajah bermasker dan wajah tidak bermasker[6].

Penelitian lain yang dilakukan oleh Randy Cahya Wihandika dengan judul “Deteksi Masker Wajah Menggunakan Metode *Adjacent Evaluation Local Binary Pattern*”. Pada penelitian ini membahas tentang perancangan sistem deteksi masker dengan menggunakan metode *Adjacent Evaluation Local Binary Pattern* yang mana metode ini merupakan pengembangan dari *local binary pattern* (LBP). Langkah yang dilakukan adalah dengan menggunakan citra *input* yang berupa citra berwarna dikonversi menjadi citra berwarna keabuan (*grayscale*), kemudian teksturnya akan diekstraksi ciri menggunakan beberapa metode yang berbeda. Hasil dari ekstraksi ciri tersebut yang berupa *histogram* akan masuk ke proses klasifikasi sehingga informasi yang didapatkan apakah wajah pada citra input menggunakan masker atau tidak menggunakan masker. Karena tujuan dari penelitian ini adalah untuk mengetahui apakah seseorang sedang mengenakan masker atau tidak, maka untuk kasus klasifikasinya adalah *biner* atau masuk dalam kategori dua kelas “bermasker” dan “tidak bermasker”. Metode lain yang digunakan juga sebagai klasifikasi adalah metode *support vector machine* (SVM) dengan berbagai *kernel* linier, polinomial dan sigmoid. Hasil dari penelitian ini menunjukkan dengan metode *adjust evaluation local binary pattern* memberikan akurasi yang baik dengan menambahkan metode *extra trees* sebagai metode klasifikasi menghasilkan akurasi 97,93%. Dilakukan juga pengujian terhadap pengaruh jumlah data latih

terhadap akurasi dan *F-measure* dengan jumlah data latih sebanyak 80% dan 10%, hasilnya adalah bahwa dengan data latih sebanyak 10% atau 217 citra akurasi yang didapatkan 96,32% tetapi dengan jumlah data latih sebanyak 80% maka akurasi yang dihasilkan sebesar 98,08% yang artinya semakin banyak data latih semakin besar akurasi yang dihasilkan[7].

Penelitian yang dilakukan pada tahun 2020 oleh Tri Septiana Puspita Putri, Mohamad Al Fikih, Novendra Setyawan dengan judul “Face Mask Detection Covid-19 Using *Convolutional Neural Network* (CNN)”. Penelitian ini memanfaatkan ekstraksi fitur dari citra yang dipelajari oleh beberapa *hidden layer* dimana cara kerja ini merupakan cara kerja dari CNN sehingga dapat mendeteksi sebuah objek. Dalam metode penelitiannya yang dilakukan berkaitan tentang *image processing* dengan memanfaatkan *machine learning* untuk mempelajari ciri khusus dari setiap objek terutama dalam membaca obyek wajah yang menggunakan masker dan tidak menggunakan masker. Dalam proses perancangan sistem ada dua proses yang dilakukan yaitu *Train Face Mask Detector* dan juga *Apply Face Mask Detector*. Untuk dataset yang digunakan proses yang dilakukan adalah pengambilan data training secara langsung menggunakan hasil gambar yang didapat dari mahasiswa di lingkungan Universitas Muhammadiyah Malang serta mengkombinasikan dengan gambar yang ada di internet. Ada sebanyak 2000 sampel dataset yang dikumpulkan dimana 95,2% digunakan untuk tahap *training* dan 4,8% digunakan untuk tahap pengujian. Tahapan *training* data dengan menggunakan CNN dataset yang digunakan terdiri dari 1904 data gambar *training* dan 96 data gambar uji. Arsitektur penelitian yang dikembangkan dilatih sebanyak 20 *epoch* (*iterations*) dan *batch size* 32. Untuk hasil dari penelitian ini dari training data menghasilkan akurasi sebesar 0,9933% dan *training loss* 0,0213%. Untuk pengujian dengan input citra wajah dihasilkan bahwa orang bermasker medis dan non medis dengan berkerudung dideteksi dengan baik dengan akurasi pembacaan diatas 87,70%. Untuk input citra wajah bermasker medis dan non medis dengan kondisi orang memakai topi juga menghasilkan akurasi pembacaan yang baik yakni dengan nilai akurasi diatas 84,99%. Untuk jarak pendeteksian yang digunakan adalah 40cm, 140cm, 200cm, dan 230 cm menghasilkan tingkat akurasi pembacaan yang baik yakni diatas 82,84% dengan akurasi tertinggi 96,32% [4].

## 2.2. DASAR TEORI

Pada penelitian ini untuk merancang sistem deteksi masker didasari oleh beberapa teori mendasar sebagai acuan dalam perancangan. Berikut ini adalah beberapa teori yang digunakan untuk menunjang kegiatan penelitian :

### 2.2.1. Artificial Intelligence (AI)

*Artificial intelligence* atau yang sering disebut sebagai kecerdasan buatan merupakan bidang keilmuan yang mampu melakukan simulasi kecerdasan dengan cara menirukan cara kerja otak manusia[8]. Umumnya, *artificial intelligence* bekerja seperti otak manusia yang mampu mengerjakan pekerjaan yang berkaitan tentang analisa serta klasifikasi sebuah data. Dalam beberapa pendapat yang dikemukakan oleh pakar keilmuan salah satunya *H. A. Simon (1987)* yang dikutip dari buku “*Dasar Komputasi Cerdas Artificial Neural Network(2016)*” mengemukakan bahwa *Artificial Intelligence* merupakan kumpulan penelitian, aplikasi, dan instruksi yang berkaitan dengan pemrograman komputer yang bertujuan untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

Pada dasarnya *Artificial Intelligence* cakupan bidang ilmunya sangat luas dan dapat dibagi menjadi dua bagian yakni :

- Konsep fundamental, yakni berkaitan tentang pencarian, representasi sebuah pengetahuan dan juga dasar sebuah penalaran
- *Soft Computing* (komputasi cerdas), yakni berkaitan dengan logika samar/*fuzzy logic*, *machine learning*, *artificial neural network*, *genetic algorithm*. [9]

### 2.2.2. Deep Learning

*Deep learning* merupakan bagian dari *machine learning* yang algoritmanya menyerupai struktur otak manusia. Sistem saraf otak manusia sangat kompleks dalam mengenali berbagai macam benda, warna, suara dan lainnya dengan konsep ini *deep learning* mengadopsi sistem saraf otak dalam membentuk struktur dan jaringan dalam proses pembelajarannya. Dalam proses kerjanya *deep learning* dapat beradaptasi dengan jumlah data yang besar serta mampu menyelesaikan berbagai permasalahan yang sulit dilakukan oleh beberapa jenis *machine learning* lainnya[10]. Jaringan saraf

*deep learning* mempelajari metadata sebagai input dan mengolahnya menggunakan lapisan tersembunyi transformasi non linier dari data *input* untuk menghasilkan *output*. *Deep learning* dapat bekerja secara mandiri sehingga akan dapat memecahkan masalah dari yang mudah hingga kepada yang kompleks.

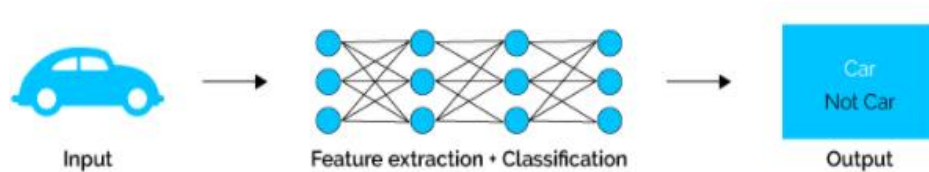
Didalam *deep learning* terdapat beberapa tipe yakni *deep learning* untuk *unsupervised learning* dan *hybrid deep learning*. Tipe *deep learning unsupervised learning* digunakan untuk variabel yang memiliki label sehingga harus menghitung nilai yang lebih tinggi dari unit yang diamati dalam analisis polanya. Sedangkan *hybrid deep network* yakni metode yang ditujukan untuk mencapai hasil data yang lebih baik dengan menggunakan pembelajaran yang diawasi atau tidak diawasi dalam proses analisis polanya[11]. Algoritma dari *deep learning* ada beberapa jenis yakni *Deep Neural Network* (DNN), *Artificial Neural Network* (ANN) dan juga *Convolutional Neural Network* (CNN). *Deep learning* menggunakan beberapa *hyperparameter training* di dalamnya yakni sebagai berikut [12]:

Tabel 2.1 *Hyperparameter Training*[13]

<b>Parameter</b>	<b>Fungsi</b>
<i>Hidden layer</i>	Untuk menentukan jumlah lapisan tersembunyi dan jumlah neuron di setiap lapisan dalam arsitektur <i>deep learning</i>
<i>Epochs</i>	Untuk mewakili jumlah iterasi yang harus dilakukan untuk memproses dataset. Satu epoch berarti menandakan sebuah algoritma <i>deep learning</i> telah belajar dari training dataset secara keseluruhan
<i>Learning rate</i>	Parameter training untuk menghitung nilai koreksi bobot pada waktu proses <i>training</i> yang mana nilai <i>learning rate</i> berada pada 0 sampai 1
<i>Batch Size</i>	Merupakan <i>Hyperparameter</i> yang digunakan dalam proses training dan berguna dalam penentuan jumlah contoh pelatihan dalam satu iterasi

<i>Evaluate Performance</i>	Kinerja model akan dievaluasi performansinya berdasarkan parameter <i>epoch</i> , <i>batch size</i> , dan <i>learning rate</i> yang telah disetting sampai dengan mendapatkan model dengan kinerja yang paling baik berdasarkan akurasinya
-----------------------------	--

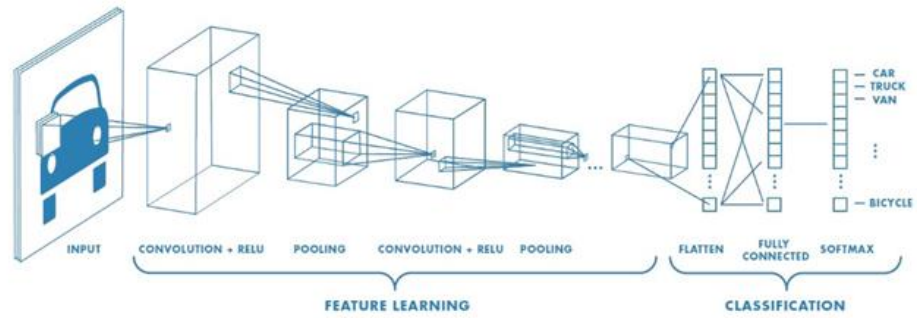
Secara sederhana struktur dari *deep learning* dapat dilihat seperti gambar berikut ini :



Gambar 2.1 Struktur *Deep Learning*[14]

### 2.2.3. Convolutional Neural Network (CNN)

Metode *convolutional neural network* (CNN) merupakan salah satu metode yang sering digunakan dalam klasifikasi gambar[15]. Metode CNN juga masih termasuk dalam jenis *deep learning* karena didalam CNN terdapat tingkat jaringan dan banyak diimplementasikan dalam data citra. Metode ini merupakan pengembangan yang dilakukan dari *Multilayer Perceptron* (MPL) yang telah dirancang untuk melakukan proses pengolahan data secara dua dimensi[5]. CNN memiliki dua metode yakni : klasifikasi menggunakan *feedforward* dan dalam tahap pembelajarannya menggunakan *backpropagation*. Penggunaan CNN dapat mengurangi sejumlah parameter bebas dan dapat menangani deformasi gambar input seperti halnya skala, rotasi, dan translasi. CNN terdiri atas tiga jenis lapisan saraf utama yakni : *convolution layers*, *pooling layers*, dan *fully connected layers*. Dari setiap jenis yang ada pada CNN memiliki peranan yang berbeda-beda[15]. Berikut ini merupakan penggambaran dari lapisan arsitektur CNN :



Gambar 2.2 Struktur CNN[16]

Secara umum metode CNN sama halnya dengan *neural network* lainnya seperti ANN atau DNN hanya saja pada CNN memiliki layer khusus yang dinamakan dengan *Convolutional Layer*.

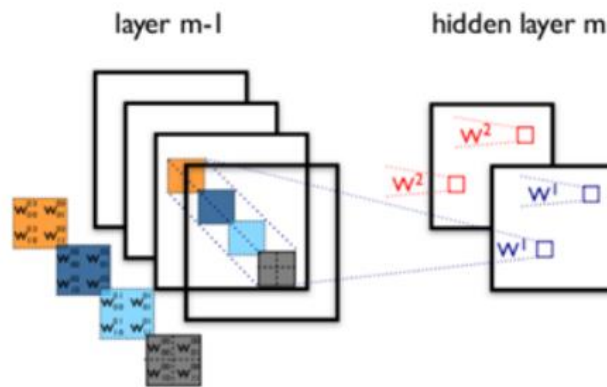
a. *Convolutional Layer*

Pada CNN terdapat *Convolutional layer* yang merupakan proses utama didalam sistem kerja CNN untuk beroperasi melakukan konvolusi pada *output* dari layer sebelumnya. Konvolusi merupakan istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Pada lapisan ini akan dilakukan operasi konvolusi antar matriks *filter*. Proses dari konvolusi dengan matriks *filter* nantinya akan menghasilkan keluaran matriks *feature map*. *Feature map* yang akan dihasilkan dapat dihitung menggunakan persamaan berikut ini[17]:

$$n_{out} = \left( \frac{n_{in} - k + 2p}{s} \right) + 1 \quad (2.1)$$

dimana :

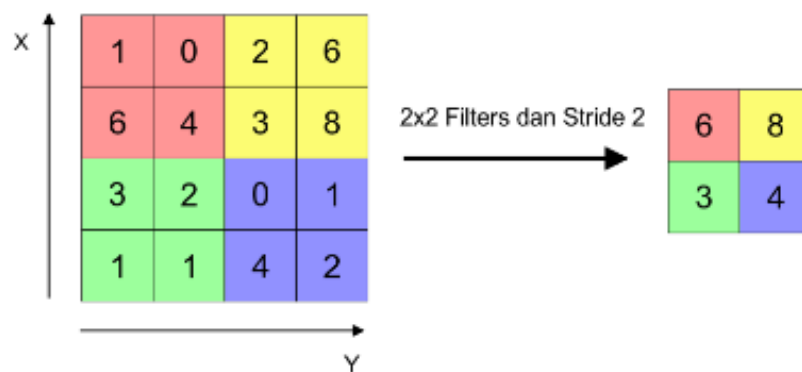
- $n_{out}$  = ukuran *feature map*
- $n_{in}$  = ukuran matriks masukan
- $k$  = ukuran matrik *filter*
- $p$  = ukuran padding
- $s$  = *stride*



Gambar 2.3 Convolutional Layer[18]

b. Pooling Layers

Fungsi dari *pooling layers* yang terdapat pada CNN bertugas untuk mengurangi dimensi spasial (lebar x tinggi) dari volume masukan untuk proses lapisan konvolusi berikutnya. Lapisan penggabungan yang dilakukan tidak akan mempengaruhi dimensi kedalaman volume. Proses *pooling layers* ini juga sering disebut *subsampling* atau *downsampling*, sebagai pengurangan ukuran yang beresiko menyebabkan hilangnya informasi secara simultan[17]. *Pooling layer* terdiri atas sebuah *filter* dengan ukuran dan *stride* tertentu. Seluruh area *feature map* atau *activations map* akan digeser sesuai dengan jumlah stride pada setiap pergeseran yang dilakukan. Yang sering digunakan dalam penerapan *pooling layer* adalah *Max Pooling 2x2* dan *Average Pooling*. Salah satu contohnya adalah ketika menggunakan *Max Pooling 2x2* dengan *stride 2* maka setiap saat pergeseran filter nilai yang akan diambil adalah nilai yang paling besar di area 2x2, sedangkan *Average Pooling* akan mengambil nilai rata-rata[19].

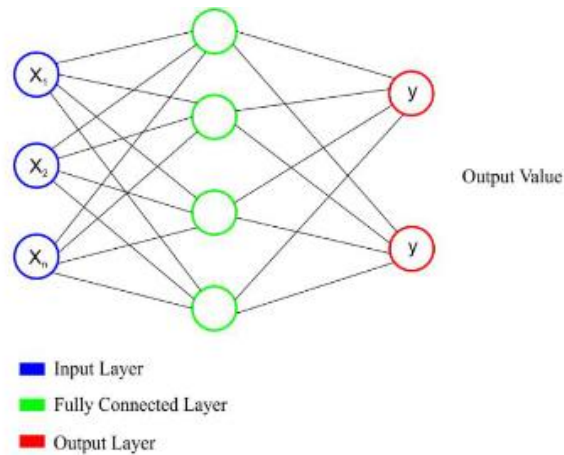


Gambar 2.4 Pooling Layer[19]



### c. Fully Connected Layers.

Lapisan ini akan mengikuti beberapa proses konvolusi dan lapisan penyatuan, penalaran tingkat tinggi di saraf jaringan yang dilakukan melalui lapisan *fully connected layers*[20]. Feature map yang dihasilkan pada tahap sebelumnya berbentuk *multidimensional array*, sehingga sebelum memasuki tahap *Fully Connected Layer*, *feature maps* tersebut akan melalui proses *flatten* atau *reshape*. Proses dari *flatten* akan menghasilkan sebuah vektor yang digunakan sebagai *input* dari *fully connected layer*. Didalam *fully connected layer* terdapat beberapa *hidden layer*, *action function*, *output layer* dan *loss function*[19].

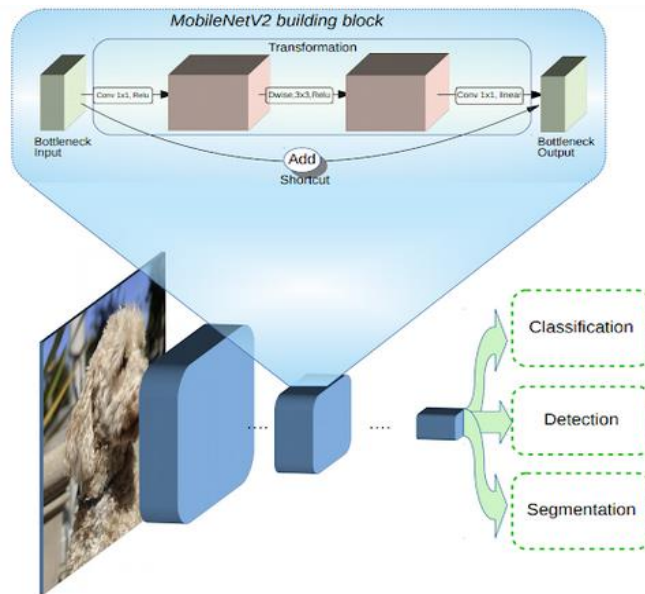


Gambar 2.5 Fully Connected Layer[19]

### 2.2.4. MobileNetV2

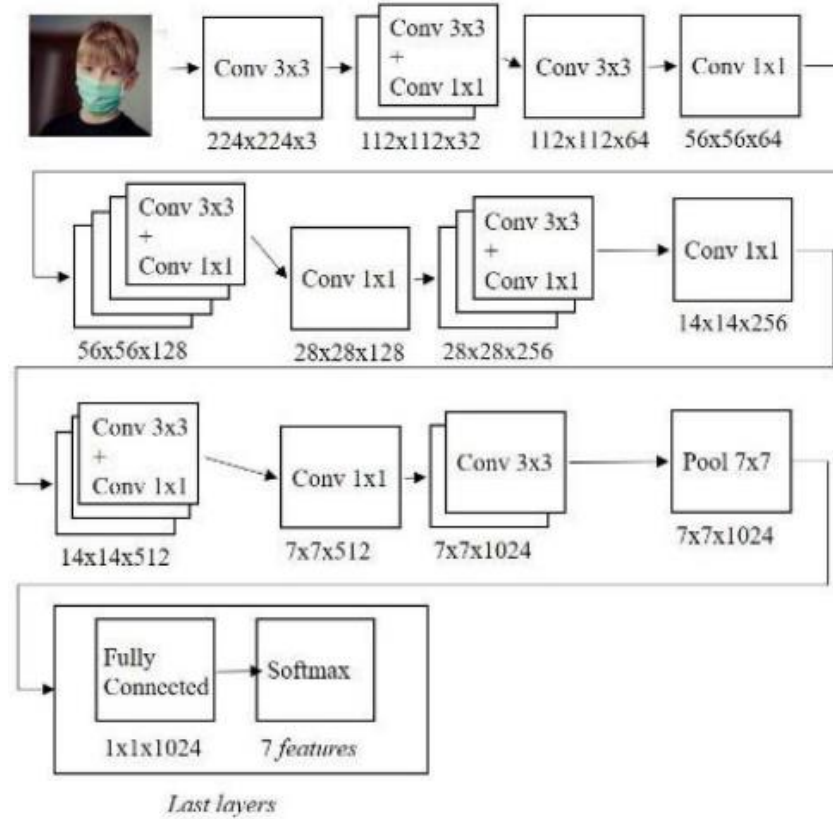
Ada beberapa jenis arsitektur yang terdapat pada CNN salah satunya adalah arsitektur menggunakan *MobileNetV2*. *MobileNetV2* merupakan pengembangan arsitektur yang dapat digunakan pada perangkat *mobile*. Dimana arsitektur ini memiliki beberapa kelebihan yakni ketebalan dari *filter* konvolusi yang sesuai dengan gambar. Dengan kelebihan ini tentu akan sangat menghemat ukuran dari model yang dibuatnya. Pada arsitektur ini juga terdapat *bottleneck* yang diletakkan pada bagian *input* dan *output* sehingga proses pelatihan akan menjadi lebih akurat dan cepat dibandingkan dengan arsitektur lainnya[21]. Maka kemungkinan untuk pembuatan aplikasi berbasis CNN sangat mungkin dilakukan pada perangkat *mobile* dengan arsitektur *MobileNetV2*. Pada dasarnya *MobileNet* akan membagi

konvolusi menjadi *depthwise convolution* dan *pointwise*. *MobileNetV2* merupakan versi kedua dari *MobileNet* yang dirilis dari pengembangan *MobileNetV1* dimana *MobileNetV2* ditambahkan fitur *linear bottleneck* dan *shortcut connections* antar *bottlenecks*. Pada bagian *bottleneck* terdapat *input* dan *output* dari model dan adanya penambahan *add shortcut* memungkinkan untuk pelatihan atau proses *training* akan lebih cepat dengan akurasi yang lebih baik juga[16].



Gambar 2.6 Arsitektur *MobileNetV2*[16]

Berikut ini pada gambar 2.7 akan menunjukkan arsitektur konvolusi pada *MobileNetV2*. Layer konvolusi akan dimasukkan *input* berupa citra dengan ukuran 224x224 pixel. Kemudian pada layer berikutnya akan dilakukan proses *pooling* untuk mengurangi dimensi citra. Pada *fully connected layer* dan *softmax* akan digunakan untuk proses klasifikasi gambar



Gambar 2.7 layer-layer pada MobileNetV2[16]

### 2.2.5. Tensorflow dan Keras

*Tensorflow* merupakan sebuah *library open source* yang berhasil dikembangkan oleh tim *google brain* untuk melakukan komputasi numerik dan dengan *machine learning* skala besar. Di dalam *tensorflow* banyak penggabungan model yang dilakukan dengan algoritma *machine learning* dan algoritma *deep learning*. Untuk membangun aplikasi dengan *framework* dalam *tensorflow* digunakan python sebagai penyedia *Application Programming Interface (API) front end* sekaligus akan menjalankan aplikasi tersebut dengan performa tinggi. Dalam proses kerjanya *tensorflow* dapat bekerja dengan berbagai macam penyelesaian masalah mulai dari melatih dan menjalankan jaringan syaraf tiruan untuk klasifikasi tulisan tangan, pengenalan gambar, penyematan sebuah kata, *recurrent neural network*, *sequence to sequence* model untuk terjemahan dan bahkan sampai simulasi berbasis PDE (*Partial Differential Equation*)[22]. Sedangkan, *Keras* merupakan *high level neural network API* yang dituliskan dalam

bahasa python dengan kemampuan yang dapat berjalan diatas *Tensorflow*[16]

### 2.2.6. Parameter Uji

Untuk mengetahui apakah sebuah sistem model yang dibangun sudah bekerja dengan baik atau tidak tentu diperlukan parameter uji sebagai pembandingan hasil yang telah didapatkan oleh sistem. Dalam penelitian ini parameter uji yang digunakan yaitu akurasi, *recall*, *F1-score*, dan *Confusion Matrix*

#### 1. Accuracy/Akurasi

Parameter akurasi akan menjelaskan keakuratan model yang dapat diklasifikasi dengan benar. Akurasi merupakan rasio prediksi benar (*positif dan negatif*) terhadap seluruh data. Jadi dapat dikatakan bahwa akurasi adalah seberapa dekat nilai prediksi yang dihasilkan dengan nilai aktual (nilai sebenarnya). Berikut ini adalah persamaan yang dapat digunakan untuk menghitung akurasi :

$$\text{Akurasi} = \frac{(\text{TP}+\text{TN})}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \quad (2.2)$$

Dimana TP (*True Positive*) adalah kasus dimana yang diprediksi (*Positive*) menghasilkan nilai yang benar. TN(*True Negative*) adalah kondisi dimana yang diprediksi (*Negative*) menghasilkan nilai yang benar. Untuk FP(*False Positive*) adalah kasus dimana yang diprediksi *positive* ternyata menghasilkan hasil yang salah. Hasil salah (*Negative*), FN (*False Negative*) adalah kasus yang diprediksi *positif* namun hasilnya tidak benar.

#### 2. Recall

*Recall* merupakan parameter uji yang menggambarkan keberhasilan model dalam memulihkan informasi. Jadi, *Recall* adalah rasio prediksi positif benar terhadap seluruh data positif benar. Berikut ini adalah persamaan yang digunakan dalam menghitung *Recall* :

$$\text{Recall} = \frac{(\text{TP})}{(\text{TP}+\text{FN})} \quad (2.3)$$

#### 3. Presisi

*Presisi* akan menggambarkan bagaimana tingkat akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Secara matematis *Presisi* dapat dituliskan dalam persamaan :

$$\text{Presisi} = \frac{(\text{TP})}{(\text{TP}+\text{FP})} \quad (2.4)$$

#### 4. *F1-Score*

Parameter *F1-Score* merupakan kalkulasi evaluasi di dalam pencarian informasi yang akan menggabungkan antara *Recall* dan Presisi. Secara matematis *F1-Score* dapat ditulis dalam persamaan :

$$\text{F1 Score} = 2 \times \frac{(\text{Recall} \times \text{Presisi})}{(\text{Recall} + \text{Presisi})} \times 100 \quad (2.5)$$

#### 5. *Confusion Matrix*

Dalam mengukur suatu *classifier* untuk melakukan prediksi pada kategori yang berbeda biasanya menggunakan confusion matrix sebagai metode yang dapat memecahkan masalah tersebut. Istilah yang ada pada *confusion matrix* untuk representasi proses klasifikasi digunakan : *True Positif*(TP), *True Negatif*(TN), *False Positif*(FP), dan *False Negatif*(FN)[23].

		<b>Actual Values</b>	
		<b>1 (Positive)</b>	<b>0 (Negative)</b>
<b>Predicted Values</b>	<b>1 (Positive)</b>	<b>TP</b> (True Positive)	<b>FP</b> (False Positive) <small>Type I Error</small>
	<b>0 (Negative)</b>	<b>FN</b> (False Negative) <small>Type II Error</small>	<b>TN</b> (True Negative)

Gambar 2.8 *Confusion Matrix*[24]