

BAB III

LANDASAN TEORI

Pada bab ini akan dibahas landasan teori mengenai sistem yang akan dibangun.

3.1.Mobile

Mobile dapat diartikan menjadi dua yaitu kata sifat dan kata benda. *Mobile* sebagai kata sifat berarti aktif, giat, gesit, dan ringan tangan, sedangkan *mobile* sebagai kata benda merujuk pada *handphone*, *cellphone*, *cellular*, atau *gadget* lain yang dapat dibawa bergereak dengan mudah [17]. Perangkat ini dirancang untuk mudah dibawa dan mudah digenggam oleh penggunanya, contoh dari perangkat *mobile* sendiri antara lain: *tablet*, *e-reader*, dan *smartphone* [18]. Berangkat dari definisi *mobile* itu sendiri, hasil dari pengembangan sistem ini akan diimplementasikan ke sebuah *smartphone* yang mana merupakan contoh dari perangkat *mobile*.

3.2.iOS

iOS adalah sistem operasi perangkat *mobile* milik Apple yang rilis pada tahun 2007, sistem operasi ini khusus dikembangkan untuk iPhone (*smartphone* buatan Apple) dan bisa juga berjalan di *device* lain milik Apple seperti iPod, Apple TV, dan iPad [19]. Apple sendiri tidak memberikan lisensi iOS untuk produk yang tidak dibuat olehnya, hanya *device* dengan *hardware* buatan Apple yang dapat menjalankan sistem operasi ini [20]. Sifat eksklusif iOS tersebut membuat aplikasi yang dikembangkan akan lebih mudah *maintain*, berbeda dengan Android yang bersifat *open source* dimana konfigurasinya tergantung dari perusahaan manufakturnya [21]. Untuk kemudahan *maintenance* fitur di masa mendatang, penulis akan mengembangkan fitur *machine learning* ini pada sistem operasi iOS.

3.3.Machine Learning

Machine learning adalah kerangka algoritma terpadu yang didesain untuk mengidentifikasi dan mendeskripsikan data empiris serta fenomena yang mendasarinya secara akurat dengan atau tanpa campur tangan manusia sama

sekali [22]. *Machine learning* bergantung pada algoritma untuk menganalisa data dalam jumlah besar, data dalam jumlah besar itu dianalisis untuk mendapatkan prediksi mengenai apa yang mungkin terjadi di masa depan [23]. Kemampuan belajar dari data ini memungkinkan algoritma untuk membuat model yang mampu memecahkan masalah dengan mencari pola dari data masa lalu yang kompleks sekaligus meningkatkan kemampuannya setiap kali data baru dimasukkan ke dalam model [24]. Untuk mendefinisikan data yang rumit seperti gambar, penggunaan percabangan *if else* sangat tidak memungkinkan karena terlalu banyak. Oleh karena itu, peran *machine learning* sangat dibutuhkan dalam penelitian ini agar tiap-tiap data dapat didefinisikan dengan baik.

3.4. Deep Learning

Deep learning adalah bagian dari *machine learning* yang berbentuk seperti jaringan saraf (*neural network*), jaringan saraf ini meniru perilaku atau cara kerja otak manusia dalam memproses data dengan jumlah yang besar [25]. *Deep learning* merupakan sebuah model non-linear yang digunakan untuk mengetahui atau mengukur hubungan antara variabel-variabel yang ada, hubungan antar variabel ini dikontrol dengan sebuah parameter dengan algoritma optimasi *gradient descent* [26]. *Deep learning* memiliki berbagai macam arsitektur yang masing-masing memiliki kelebihan dan kekurangannya sendiri, salah satu arsitektur *deep learning* adalah *Convolutional Neural Network* atau CNN [27]. Objek dari penelitian ini adalah sebuah gambar yang bisa saja terdiri dari puluhan hingga jutaan piksel di dalamnya, dengan memanfaatkan *deep learning* hubungan antar piksel-piksel tersebut dapat diubah menjadi sebuah model. Model inilah yang nantinya digunakan sebagai acuan dalam melakukan pengklasifikasian.

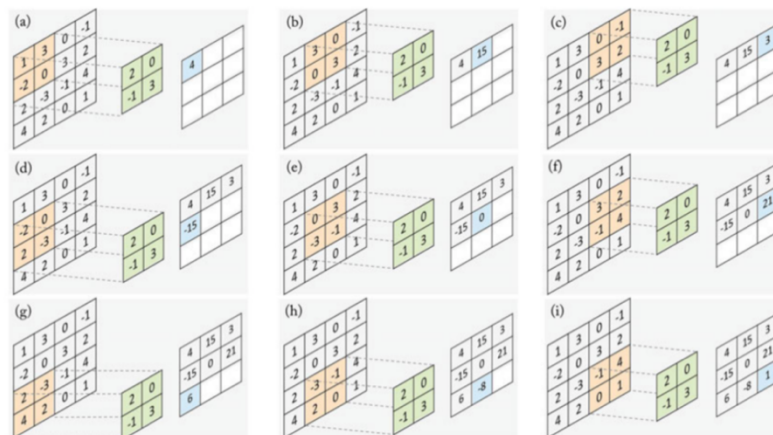
3.5. Convolutional Neural Network

Convolutional neural network atau CNN merupakan salah satu arsitektur *deep learning* yang dikhususkan untuk memproses data dengan banyak properti seperti gambar dan suara [28]. CNN merupakan salah satu arsitektur yang paling sering digunakan untuk melakukan pengklasifikasian data [4]. CNN

sendiri terdiri dari beberapa *layer*, yaitu *input layer*, *convolutional layer*, *pooling layer*, *fully-connected layer*, dan *output layer* [29].

1. Convolutional Layer

Bagian ini merupakan bagian utama dari CNN, bagian ini akan melakukan pemindaian di semua bagian pada gambar untuk mencari ciri tertentu. *Scanning* akan dilakukan dengan menggunakan *filter* yang ukurannya lebih kecil dari gambar *input* [30]. Pada Gambar 3.1. bagian tertentu dari suatu gambar dilakukan *scanning* dengan menggunakan *filter*, lalu hasil dari pemindaian ini adalah *Dot product*. *Dot product* adalah jumlah dari perkalian nilai yang ada pada gambar dengan bobot *filter*. Proses pada *convolutional layer* ini akan berlangsung sampai seluruh bagian dari gambar selesai dipindai dengan cara menggeser bagian yang akan dipindai [31].

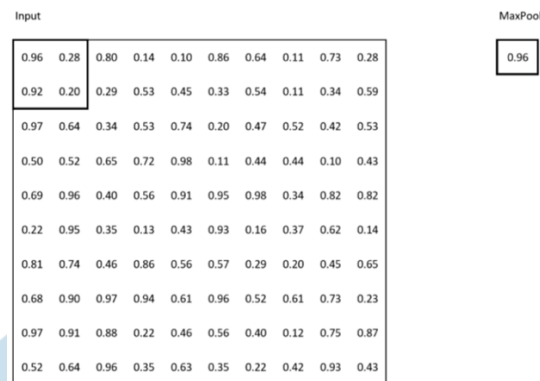


Gambar 3.1. Ilustrasi *Convolutional Layer*

2. Pooling Layer

Bagian ini biasanya diletakkan setelah *convolution layer* untuk mengurangi dimensi dari gambar yang dijadikan sebagai *input* [32]. *Pooling layer* akan melakukan *downsample* terhadap *feature maps* yang dikeluarkan oleh *convolutional layer*, hal ini menyebabkan jumlah parameter dan kompleksitas komputasi yang dibutuhkan akan mengalami penurunan [33]. Gambar 3.2 merupakan ilustrasi cara kerja

dari *pooling layer* jenis *MaxPool*, pada *pooling layer* ini hanya *value* terbesar yang akan diambil dari wilayah yang sedang dipindai.



Gambar 3.2. Ilustrasi *MaxPooling Layer*

3. *Fully Connected Layer*

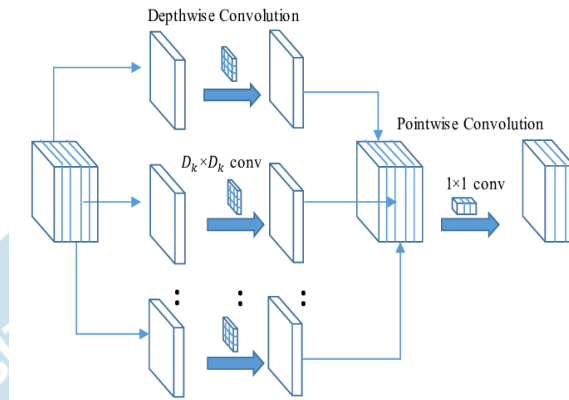
Bagian ini merupakan layer terakhir dari CNN berupa *array* satu dimensi. Namun sebelum masuk ke *fully connected layer*, harus menambahkan lapisan *Flatten* agar data atau *feature* yang sebelumnya merupakan multidimensi *array* menjadi sebuah *vector* dan bisa digunakan sebagai *input* oleh lapisan ini. Tujuan dari *layer* ini adalah untuk mengklasifikasikan hasil dari proses lapisan-lapisan sebelumnya [32].

Convolutional neural network memiliki banyak *pre-trained weight model*, yaitu sebuah model yang sudah dilatih dengan *dataset* tertentu dan bisa digunakan kembali dengan metode *transfer learning* [34]. Beberapa contoh dari *pre-trained model*, yaitu ResNet50, MobileNet, MobileNetV2, VGG, dan lain-lain [34]. CNN sangat cocok dengan kebutuhan dari penelitian ini yang ditujukan untuk mengklasifikasikan produk berdasarkan gambar. Penulis akan menggunakan salah satu dari *pre-trained model* yang dimiliki oleh CNN dalam membangun model untuk penelitian ini.

3.6. MobileNetV2

MobileNet merupakan contoh dari arsitektur CNN yang didesain khusus menggunakan *depthwise separable convolutions* agar ringan saat berjalan di *mobile devices* [6]. *Depthwise separable convolution* merupakan sebuah blok terdiri dari *1x1 pointwise convolution* dan *depthwise convolution*, untuk ilustrasi

dari *depthwise separable convolution* dapat dilihat pada Gambar 3.3. *Depthwise* digunakan untuk mengaplikasikan *single filter* tiap *input channel* sedangkan *pointwise convolution* digunakan untuk membuat kombinasi linear dari keluaran *depthwise layer* [6].



Gambar 3.3. Ilustrasi *Depthwise Separable Convolution*.

MobileNetV2 merupakan penerus dari MobileNet pendahulunya, hanya saja MobileNetV2 ini memiliki parameter yang lebih sedikit dan mampu menerima masukan berupa gambar dengan ukuran diatas 32x32 piksel sehingga performanya lebih unggul [35]. Arsitektur MobileNetV2 mendapat penambahan fitur berupa *linear bottlenecks* yaitu sebuah *inverted residual block* yang digunakan untuk mereduksi kompleksitas model dan menghapus aktivasi ReLU pada akhir proses *inverted residual block* agar informasi yang hilang pada saat aktivasi bisa dikembalikan [35]. Hasil akhir dari penelitian ini akan diimplementasikan ke dalam sebuah *mobile devices* yang mana memiliki kemampuan komputasi yang terbatas, oleh karena itu penulis menggunakan arsitektur MobileNetV2 agar fitur *machine learning* berjalan secara optimal meskipun *resource* komputasinya terbatas.