BAB II
TINJAUAN PUSTAKA

2.1. Biji Kopi Robusta

Biji kopi robusta merupakan salah satu jenis tanaman kopi dengan nama ilmiah *coffea canephora*. Nama kopi tersebut diambil dari kata robust, istilah dalam bahasa Inggris yang diartikan kuat. Minuman yang diekstrak dari biji kopi robusta biasanya memiliki citra rasa yang kuat dan cenderung lebih pahit dibandingkan arabika, berikut di bawah ini merupakan contoh dari buah kopi yang ditunjukkan pada Gambar 2.1.

Gambar 2.1 Kopi Robusta menjelang matang [2]

Gambar 2.1 merupakan buah kopi yang sudah matang dan akan siap diolah menjadi bubuk kopi. Biji Kopi robusta banyak sekali digunakan sebagai bahan baku kopi siap saji atau biasanya disebut *instant* dan juga sebagai pencampur kopi racikan untuk menambah kekuatan citra rasa kopi [2]. Selain itu kopi robusta juga bisa digunakan untuk membuat minuman kopi dengan campuran susu seperti *capucino, café latte,* dan *macchiato*. Biji kopi robusta dihargai lebih rendah dibandingkan dengan biji kopi arabika, dan Indonesia merupakan salah satu negara penghasil Kopi Robusta terbesar di dunia.

2.2. Klasifikasi Citra

Klasifikasi citra merupakan sebuah pekerjaan untuk memasukkan sebuah citra dan menetapkannya ke sebuah kategori. Salah satu permasalahan dalam

computer vision yang dapat disederhanakan dan memiliki berbagai macam aplikasinya. Salah satu aplikasi dalam klasifikasi citra adalah pengklasifikasian nama tempat pada suatu citra, pengklasifikasian jenis biji kopi, dan lain sebagainya berikut ini merupakan contoh klasifikasi citra biji kopi pada Gambar 2.2 di bawah ini.

Gambar 2.2 Klasifikasi Citra Biji Kopi [9]

Gambar 2.2 merupakan hasil klasifikasi yang terawasi, setiap masukkan citra pada *training set* diberi label. Saat klasifikasi, label tersebut akan menjadi perbandingan dengan hasil hipotesis yang diberikan oleh model pembelajaran dan akan menghasilkan nilai *error* [9]. Klasifikasi yang terawasi bisa sangat efektif dan akurat dalam mengklasifikasikan citra tempat maupun objek lainnya. Banyak metode dan algoritma yang dapat mendukung proses klasifikasi yang terawasi terutama dengan teknik *deep learning.*

2.3. *Deep Neural Network*

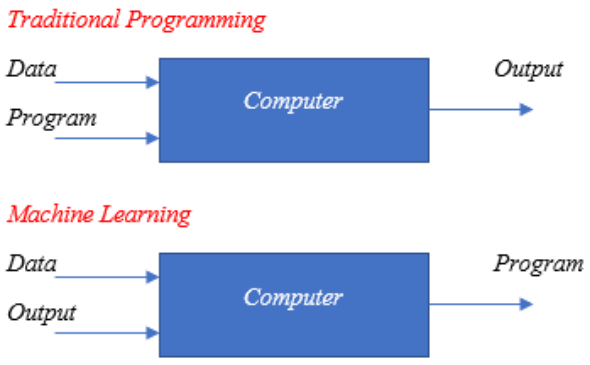
Artificial Neural Network (ANN) merupakan metode yang biasanya digunakan dalam peramalan maupun pengenalan pola. pada peramalan jaringan syaraf tiruan biasa digunakan sebagai peramalan nilai tukar mata uang asing, peramalan harga saham, peramalan cuaca, dan lain sebagainya, sedangkan untuk pengenalan pola biasanya jaringan syaraf tiruan digunakan untuk pengenalan pola huruf, pola tanda tangan hingga pola suara, serta wajah [16].

Deep learning merupakan bagian dari *machine learning* yang terdiri dari banyak lapisan yang dikenal dengan *hidden layer* dan membentuk jaringan saling terkoneksi. Lapisan tersebut adalah algoritma yang melakukan klasifikasi perintah yang di masukkan hingga menghasilkan keluaran.

Metode *deep learning* atau *deep neural network* yang sedang berkembang salah satunya adalah *convolutional neural network*. Jaringan ini menggunakan masukan berupa gambar, kemudian akan melalui lapisan konvolusi dan diolah berdasarkan *filter* yang ditentukan, setiap lapisan ini menghasilkan pola dari beberapa bagian citra yang memudahkan proses klasifikasi [17][18].

2.4. *Machine learning*

Machine learning adalah serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara merepresentasikan data-data tersebut dengan algoritma pembelajaran. *Machine learning* dapat membuat komputer memprogram diri mereka sendiri [19]. Jika pemrograman adalah pekerjaan untuk membuat otomasi, maka *machine learning* mengotomatisasi proses otomasi. pada dasarnya *machine learning* membiarkan data melakukan pekerjaan [20]. Berikut Gambar 2.3 merupakan perbandingan *machine learning* dengan pemrograman secara tradisional.



Gambar 2.3 *Traditional Programming and Machine learning*

Gambar 2.3 di atas dapat dilihat bahwa pemrograman secara tradisional data dan program dijalankan di komputer untuk menghasilkan keluaran. Sedangkan

pada *machine learning* data dan keluraran yang dijalankan di komputer untuk membuat sebuah program.

2.5. *Convoluition Neural Network*

Convolutional neural network merupakan salah satu pengembangan dari jaringan syaraf tiruan yang terinspirasi dari jaringan syaraf manusia dan biasa digunakan pada data gambar untuk mendeteksi dan mengenali suatu objek pada sebuah gambar. Teknik ini terinspirasi dari cara Mamalia atau Manusia menghasilkan persepsi visual seperti yang dilakukan Mamalia dan Manusia. Secara garis besar *Convolutional Neural Network* (CNN) tidak jauh beda dengan neural network biasanya. CNN terdiri dari *neuron* yang memiliki *weight*, bias dan *activation function*. *Convolutional layer* juga terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi *(pixels)* [21].

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang di desain untuk mengolah data dua dimensi. Pada CNN, setiap *neuron* direpresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap *neuron* hanya berukuran satu dimensi. CNN termasuk dalam *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [22]. CNN hampir sama dengan *neural network* pada umumnya yang memiliki neuron yang memiliki bobot dan bias. CNN memiliki 1 tahap *training* (*Supervised Backpropagation*). CNN terdiri dari *neuron* yang memiliki bobot, bias, dan fungsi aktivasi.

2.6. *Feature Extraction Layer*

Proses yang terjadi pada bagian ini adalah melakukan *encoding* dari sebuah *image* menjadi *features* yang berupa angka-angka yang merepresentasikan *image* tersebut. *Feature extraction layer* terdiri dari dua bagian yaitu c*onvolutional layer* dan *pooling layer* [23].

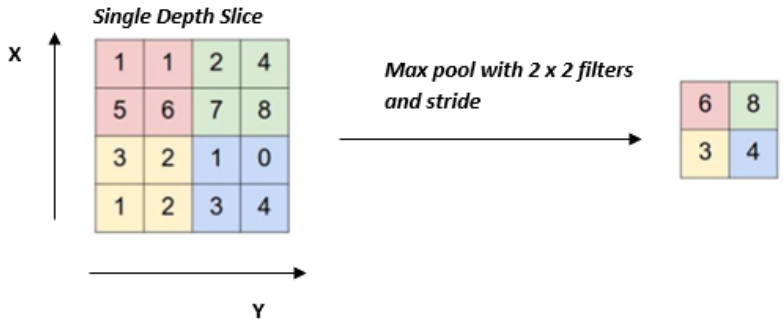
Convolutional layer terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah *filter* dengan panjang dan tinggi (*pixel*). Pada Gambar 2.6 terlihat layer pertama pada *feature extraction layer* adalah *convolutional layer* dengan ukuran $5 \times 5 \times 3$. Panjang 5 *pixel*, tinggi 5 *pixel*, dan tebal/jumlah 3 buah

sesuai dengan *channel* dari gambar tersebut. Ketiga *filter* ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi *dot* antara masukkan dan nilai dari *filter* tersebut sehinga menghasilkan sebuah *output* atau biasa disebut sebagai *actvation map* atau *feature map* [22].

Stride adalah parameter yang menentukan berapa jumlah pergeseran *filter*. Jika nilai *stride* adalah 1, maka conv. *filter* akan bergeser sebanyak 1 *pixel* secara *horizontal* lalu *vertical*. Semakin kecil *stride* maka akan semakin detail informasi yang kita dapatkan dari sebuah *input*, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Namun perlu diperhatikan bahwa dengan menggunakan *stride* yang kecil kita tidak selalu akan mendapatkan performa yang bagus [24].

Fungsi aktivasi berada pada tahap sebelum melakukan *pooling layer* dan setelah melakukan proses konvolusi. Pada tahap ini, nilai hasil konvolusi dikenakan fungsi aktivasi atau *activation function*. Terdapat beberapa fungsi aktivasi yang sering digunakan pada *convolutional network*, di antaranya reLU. Aktivasi reLU menjadi pilihan bagi beberapa peneliti karena sifatnya yang lebih berfungsi dengan baik.

Fungsi yang digunakan untuk aktivasi pada reLU, fungsi reLU adalah nilai *output* dari neuron bisa dinyatakan sebagai 0 jika *input*nya adalah negatif. Jika nilai *input* dari fungsi aktivasi adalah positif, maka *output* dari *neuron* adalah nilai *input* aktivasi itu sendiri. Berikut merupakan contoh dari *max pooling layer* pada Gambar 2.4 di bawah ini.



Gambar 2.4 Contoh *Max Pooling* [10]

Gambar 2.4 di atas, lapisan *pooling* menggunakan salah satu operasi maksimal yang merupakan operasi yang paling umum. Gambar 2.4 menunjukkan operasi dengan langkah 2 dan ukuran *filter* 2×2. dari ukuran *input* 4×4, pada masing-masing 4 angka pada *input* operasi mengambil nilai maksimalnya dan membuat ukuran *output* baru menjadi 2×2 [25].

2.7. Keras

Keras merupakan *neural network library* Bahasa Python yang mudah digunakan. Keras mampu berjalan di atas Tensorflow, Microsoft Cognitive Toolkit, R, Theano, atau PlaidML [26]. Keras dirancang untuk menjalankan eksperimen dengan *deep neural network* dengan cepat.

Keras berfokus untuk mudah digunakan, modular, dan dapat diperluas. Keras di kembangkan sebagai salah satu dari hasil research *pada project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).* Penulis utama dan pengembang adalah *Francois Chollet*, seorang *Google engineer. Chollet* juga adalah seorang pembuat deep dari *XCeption neural network* model [27][28] .

2.8. Bahasa Pemrograman

Python merupakan salah satu contoh bahasa tingkat tinggi. Contoh lain bahasa tingkat tinggi adalah Pascal, C++, Pert, Java, dan sebagainya. Sedangkan bahasa tingkat rendah merupakan bahasa mesin atau bahasa *assembly*. Secara sederhana, sebuah komputer hanya dapat mengeksekusi program yang ditulis dalam bentuk bahasa mesin. Oleh karena itu, jika suatu program ditulis dalam bentuk bahasa tingkat tinggi, maka program tersebut harus diproses dulu sebelum bisa dijalankan dalam komputer. Hal ini merupakan salah satu kekurangan bahasa tingkat tinggi yang memerlukan waktu untuk memproses suatu program sebelum program tersebut dijalankan.

Bahasa tingkat tinggi mempunyai banyak sekali keuntungan. Bahasa tingkat tinggi mudah dipelajari, mudah ditulis, mudah dibaca, dan tentu saja mudah dicari kesalahannya. Bahasa tingkat tinggi juga mudah diubah porTabel untuk disesuaikan dengan mesin yang menjalankannya. Hal ini 27 berbeda dengan bahasa mesin yang hanya dapat digunakan untuk mesin tersebut, dengan berbagai kelebihan ini, maka

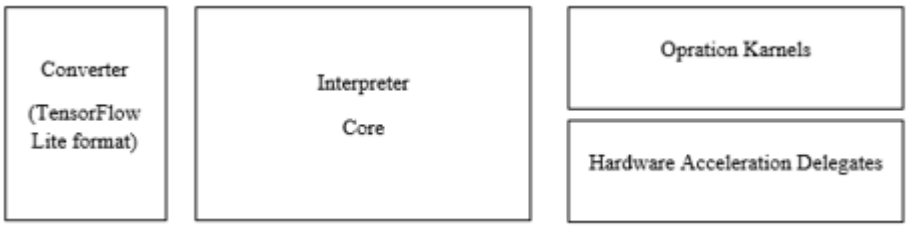
banyak aplikasi ditulis menggunakan bahasa tingkat tinggi. Proses mengubah dari bentuk bahasa tingkat tinggi ke tingkat rendah datam bahasa pemrograman ada dua tipe, yakni *interpreter* dan *compiler* [29].

Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan [33]. Java merupakan bahasa pemrograman yang membangun struktur Android Studio.

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada platform Android. Android Studio ini berbasis pada IntelliJ IDEA, sebuah IDE untuk Bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau layout, digunakan bahasa XML. Android studio juga terintegrasi dengan Android *Software Development Kit* (SDK) untuk mendeploy ke perangkat Android [34].

2.9. Tensorflow Lite

Tensorflow Lite adalah seperangkat alat yang dikembangkan oleh Tim *Google Brain* untuk membantu *developers* menjalankan model Tensorflow pada perangkat seluler, sistem tertanam, dan IoT [30]. Tensorflow *Lite* sendiri mempunyai keunggulan yaitu mempunyai sistem optimisasi yang membuat proses klasifikasi menjadi lebih ringan pada perangkat *mobile*. Berikut ini adalah beberapa optimisasi yang berada pada komponen Tensorflow *Lite* pada gambar 2.5 di bawah ini.



Gambar 2.5 Arsitektur Tensorflow *Lite*

Gambar 2.5 diatas merupakan tampilan umum pada arsitektur Tensorflow *Lite* yang terdiri dari *converter*, *interpreter*, *opration karnels*, dan *hardware*

acceleration. Arsitektur ini dapat diimplementasikan kedalam perangkat Android, sehingga pengguna dapat dengan mudah menggunakannya.

Tensorflow Lite *Converter* menggunakan model Tensorflow yang sudah dilatih untuk membuat TFLite Model. TFLite model merupakan Tensorflow model yang ukurannya sudah dikecilkan dan mampu menampilkan isi dari model tersebut. Flatbuffer merupakan format yang digunakan Tensorflow dan berperan penting untuk mengefisienkan model, membuat akses kedalam data lebih cepat selagi membuat ukurannya lebih kecil. Ada beberapa cara untuk membuat Tensorflow *Lite* model salah satunya adalah dengan menggunakan Python API [31]. Python API berfungsi untuk mempermudah mengkonversi model untuk digunakan sebagai bagian dari pengembangan model *pipeline* dan membantu mengurangi masalah lebih awal.

Interpreter Core atau TFLite *interpreter* berfungsi sebagai pengeksekusi agar TFLite model bisa berjalan di perangkat *mobile* menggunakan operator Tensorflow yang sudah dibatasi, dengan membatasi *default operators, library*, dan *tools* yang digunakan untuk menjalankan TFLite model, ukuran *Interpreter Core* sudah dikecilkan menjadi sekitar ~100KB[14].

Optimisasi pada Tensorflow *Lite* sudah sampai ketahap *hardware*. Untuk bekerja pada perangkat dengan banyak batasan berarti prosesor pada perangkat harus digunakan dengan sangat efisien. Tensorflow Lite NNAPI sudah ada pada semua Android dengan bersi 8.1 (API *Level* 27) ke atas, NNAPI berfungsi memberikan akselerasi terhadap Tensorflow Lite model pada Android dengan menggunakan akselerasi *hardware* seperti:

a. *Graphics Processing Unit* (GPU).
b. *Digital Signal Processor* (DSP).
c. *Neural Processing Unit* (NPU).
d. GPU Deligate

Tensorflow Lite juga mendukung akselerasi menggunakan GPU secara langsung. GPU dapat memproses data dengan ukuran 16 bit hingga 32 bit floating *point*, sehingga tidak dibutuhkan *quantization* untuk mendapatkan performa yang optimal [32].

2.10. Teachable Machine

Teachable Machine adalah suatu aplikasi berbasis *website* atau *web tool* yang berfungsi untuk mempermudah proses pembuatan *machine learning* sehingga dapat dilakukan semua orang. Teachable machine dapat dilakukan hanya dengan 3 tahap yaitu mengumpulkan data, melatih data, dan mengeskpor atau menyimpan model. Pada Gambar 2.6 merupakan proses dalam penggunaan Teachable Machine dalam melakukan *export* dan *training* model.

Gambar 2.6 Proses Menggunakan Teachable Machine, (a) *Gather* mengumpulkan data, (b) *Train* memproses data, (c) Export mengekstrak data

Gambar 2.6 di atas merupakan proses dari Teachable Machine yang bekerja dengan menggunakan Ternsorflow.js. Tensorflow.js merupakan sebuah *library* untuk *machine learning* yang berjalan di javascript yang berfungsi untuk melatih dan menjalankan model yang dibuat melalui *web browser* [35].

2.11. Kajian Pustaka

Pada penelitian skripsi ini mengacu pada jurnal-jurnal terkait dalam pembahasan yang sama. Kajian akan meliputi kekurangan serta kelebihan yang kemudian menjadi bahan pertimbangan dalam menambahkan metode yang akan digunakan, serta analisis yang harus ditambahkan pada penelitian ini. Penelitian yang berkaiatan tentang kualitas biji kopi sudah pernah dilakukan yakni, dalam proses pengemasan saat ini, penyortiran ini dilakukan secara manual. *Convolution Neural Network* diterapkan untuk secara otomatis untuk mengetahui informasi kecacatan Biji Kopi Arabika [11]. *Input* yang digunakan dalam penelitian ini adalah gambar Biji Kopi Arabika dengan proses penguraian yang telah dikeringkan. Sekenario yang terlibat dalam penelitian ini adalah pengumpulan data, preprocessing, klasifikasi dan pengujian. *Preprocessing* dilakukan dengan

memotong beberapa cakupan objek biji kopi yang hanya berisi gambar biji kopi. Klasifikasi dilakukan oleh CNN, untuk mendapatkan akurasi model yang terbaik, parameter yang ada harus diuji dan dievaluasi. Pengujian dilakukan untuk dua jenis model, model 2 kelas dan model 4 kelas. Hasil percobaan menunjukkan bahwa akurasi terbaik yang diperoleh untuk model 2-kelas adalah 82,46% dengan menggunakan tingkat pembelajaran 0,0001, konvolusi lapisan tunggal dengan lima belas *filter* dan 100 *neuron* pada lapisan tersembunyi. Ukuran *filter* adalah 3x3x3. Sedangkan model 4-kelas memperoleh akurasi terbaik 70,73% dengan dua lapisan konvolusional. Jumlah *filter* di setiap lapisan adalah 6 *filter* dengan ukuran 3x5x5 di lapisan pertama dan 18 *filter* dengan ukuran 6x3x3 di lapisan kedua.

Penelitian selanjutnya yang berkaitan dengan kopi yang sudah dilakukan, dalam penelitian ini kualitas biji kopi secara konvensional ditentukan oleh visual inspeksi, yang subjektif, membutuhkan upaya yang cukup besar dan waktu sehingga rawan dari kesalahan. Karakteristik biji kopi dari berbagai kota di *Cavite* [5]. Teknik pencitraan yang secara otomatis mengklasifikasi sampel biji kopi menurut jenis kopi. Yang menjadi titik uji pada penelitian ini berkaitan dengan spesifikasi biji kopi itu sendiri seperti luas, keliling serta diameter dari biji kopi tersebut dan presnetasi kebulatan yang diekstarsi dari 195 gambar dan 60 pengujian gambar. Pada penelitian ini teknik pengolahan citra yang digunaka yakni *Artificial neural network* (ANN) and *K nearest neighbor* (KNN), dengan skor klasifikasi 84,12%(k=1), 84,10%(k=2), 81,53%(k=3), 82,56%(k=4), 75,38%(k=5),80,35% (k=6), 38,79%(k=7), 77,44%(k=8), 72,82%(k=9) dan 78,45% (k=10). dari hasil penelitian tersebut dapat disimpulkan bahwa teknik tersebut dapat digunakan sebagai metode efektif untuk klasifikasi jenis biji kopi.

Penelitian selanjutnya yang berkaitan dengan kopi yang sudah dilakukan, dalam penelitiannya yakni mengklasifikasi tiga jenis kopi arabika menggunakan *computer vision* dengan metode klasifikasi menggunakan alexnet yang merupakan arsitektur CNN dengan anaisis beberapa variasi seperti SGDm, dan RMSProp dan *learning rate* 0,00005 dan 0,0001 [37]. Setiap jenis kopi menggunakan 500 data untuk pelatihan dan validasi dengan distribusi 70% pelatihan dan 30% validasi. Hasilnya menunjukkan bahwa semua Model AlexNet mencapai akurasi validasi yang sempurna nilai 100% dalam 1.040 iterasi. Penelitian ini juga menggunakan

100 data *testing-set* pada setiap jenis kopi kacang. Pada pengujian confusion matrix, akurasi mencapai 99,6%. Penelitian selanjutnya yang berkaitan dengan kopi yang sudah dilakukan, menjelaskan proses klasifikasi menggunakan metode CNN dengan akaurasi 93,33% dalam klasifikasi. Pengembangan aplikasi Android yang digunakan untuk klasifikasi menggunakan Tensorflow dan MobileNetV2 dengan datasheet 500 citra yang didistribusikan dalam lima kelas dengan 80% gambar yang digunakan untuk *training* dan 20% untuk validasi nya, dan hasilnya 80% akurasi pengenalan empat kelas, kinerja dari aplikasinya dengan metriks yang lebih tinggi dari 75% untuk tiga kelas yang di-identifikasi.

Penelitian selanjutnya yang berkaitan dengan *Image Clasification* yang sudah dilakukan, dalam penelitian ini klasifikasi menggunakan metode deep learning yang masukkan (input data) berupa citra yang menghasilkan sebuah pola dari beberapa bagian citra yang nantinya akan mudah diklasifikasikan. Pengklasifikasian ini diimplementasikan pada kebun dan sawah, dengan tujuan untuk membedakan karakteristiknya. Berdasarkan hasil klasifikiasi diperoleh hasil akurasi testing 75%. Dapat disimpulkan bahwa metode CNN dapat mengklasifikasikan citra kebun dan sawah dengan baik [23].

Penelitian selanjutnya yang berkaitan dengan *Image Clasification* yang sudah dilakukan, dalam penelitian ini klasifikasi menggunakan metode Deep learning (DL) yang dapat digunakan untuk mendeteksi dan mengenali suatu objek pada citra digital. Hasil yang didapatkan pada penelitian ini bahwa aplikasi android dapat berjalan dengan baik dengan akurasi sebesar 92. 33% dapat dilihat dari hasil pengujian dengan menggunakan metode 10-fold cross validation, semua menu yang tersedia dapat dijalankan dan penyebutan label objek sudah sesuai untuk pengenalan dan klasifikasi citra. Perhitungan presisi dan recall memiliki nilai yang baik, masing-masing sebesar 97,51% dan 94,33%. Pada proses klasifikasi, objek yang tidak ada pada dataset yang telah dimodelkan oleh sistem akan menjadi null atau tidak dikenali, terutama pada citra objek yang ditangkap oleh kamera Android yang memiliki banyak objek dan berdekatan [16].