

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 STUDI LITERATUR

Berdasarkan dari kajian pustaka yang terdiri dari beberapa penelitian mengenai proses klasifikasi citra dengan berbagai macam ekstraksi ciri dan metode yang digunakan, maka referensi dari penelitian terdahulu sangat penting untuk dilakukan agar terhindar dari penjiplakan atau duplikasi. Hal ini bertujuan sebagai bahan untuk kontribusi penelitian bagi penulis agar penelitian tentang tema ini terus berkembang. Berikut beberapa ulasan tentang penelitian terdahulu yang pernah dilakukan.

Penelitian mengenai “*Deep Learning* untuk Deteksi Tanda Nomor Kendaraan Bermotor Menggunakan Algoritma *Convolutional Neural Network* Dengan Python dan Tensorflow” yang dilakukan oleh Imam Taufiq (2018). Dalam penelitian ini menggunakan algoritma *Convolutional Neural Network* untuk klasifikasi dan mendeteksi plat kendaraan bermotor pada sebuah gambar[20].

Dalam penelitian ini 502 dataset gambar digunakan dan rasionya adalah 80% untuk pelatihan dan 20% untuk pengujian. Pelatihan membutuhkan lebih dari 25.000 langkah dengan *batch* 8 dan ketika *batch* yang digunakan adalah 100.000 langkah diperlukan hingga model yang dilatih dapat mendeteksi keberadaan TNKB dan menghasilkan akurasi sekitar 99% pada citra cakram kendaraan bermotor.

Penelitian Feri Sulianta, Ari Purno Wahyu pada tahun 2019 yang berjudul “Pemanfaatan Teknologi *Smart Farming* dan *Intelephant Imaging* untuk Meningkatkan Sikap Dan Kesadaran Konservasi Area Perkebunan Kopi” meneliti tentang implementasi dan pengawasan konservasi yang menggunakan sistem *smart farming* dan *intelephant imaging*. Dengan adanya penerapan *intelephant imaging* dapat diakumulasikan dan diterapkan untuk mendukung kegiatan konservasi. Penggunaan sistem ini dibantu dengan menggunakan perangkat lunak (*software*) dan sensor yang dapat mendukung sistem dalam melakukan analisis pada konservasi berupa pengenalan jenis hama, jenis dan gejala penyakit yang dapat menyerang perkebunan kopi[18].

Selain itu, sistem yang berbentuk sensor ini yang terkoneksi dengan jaringan *microcontroller* dapat dimanfaatkan sebagai pengamanan area perkebunan kopi, mencegah terjadinya kontak langsung antara hewan dan manusia. Sistem konservasi ini tidak menggantikan fungsi dan petugas melainkan dimanfaatkan sebagai alat bantu meringankan tugas [9].

Penelitian tentang “*Object recognition with deep learning applied to fashion items detection in images*” oleh Helder Filipe de Souza Russa [22]. Dalam penelitian ini, item fashion khusus yang diklasifikasikan dan dideteksi oleh Fast RCNN. Menggunakan data sejumlah 3677 gambar data latih untuk setiap kategori dan 696 gambar uji untuk setiap kategori yang dihasilkan. Hasil penelitian menunjukkan bahwa penggunaan metode CNN untuk mendeteksi fashion item yang dikenakan seseorang memiliki rata-rata akurasi sekitar 78% untuk celana, 65% celana serta untuk aksesoris kacamata rata-rata 57%. Metode Fast RCNN digunakan untuk lebih mengurangi waktu pelatihan objek[22].

Penelitian Fitri Muwardi, Abdul Fadlil pada tahun 2017 yang berjudul “Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak” yang meneliti tentang cahaya ruangan dapat mempengaruhi kualitas hasil citra itu, Dalam penelitian ini ekstraksi ciri yang lebih akurat dengan menggunakan ekstraksi ciri histogram manhattan yaitu ekstraksi ciri dari segi warna citra, Dalam sistem pengenalan bunga, metode klasifikasi jarak manhattan dan *euclidean* yang lebih akurat dalam mengklasifikasi citra jenis bunga adalah metode *manhattan*, hasil pengujian dapat diketahui bahwa sistem identifikasi citra jenis bunga menunjukkan tingkat akurasi yang tertinggi adalah 85% dengan menggunakan metode jarak manhattan dengan ekstraksi ciri histogram. Sedangkan tingkat akurasi yang paling rendah adalah 77% dengan menggunakan metode jarak *euclidean* dengan ekstraksi ciri statistik orde 1[7].

Penelitian Xiaochun Mai et al. pada tahun 2018, yang berjudul “*Classifier Fusion for Small Fruit Detection*“, tujuan yang ingin dicapai pada penelitian ini adalah mendeteksi buah kecil yang ada pada pohonnya pada penelitian ini buah yang akan dideteksi adalah buah almond yang masih berada pada pohonnya[19].

## 2.2 LANDASAN TEORI

### 2.2.1 Perkembangan Teknologi Pertanian di Indonesia

Di tengah pesatnya perkembangan teknologi, PT Bank Negara Indonesia Tbk (BBNI) mendorong pertanian menerapkan *smart farming*, melalui Gerakan Menyongsong Pertanian 4.0. Hal ini diharapkan bisa mempersiapkan petani untuk masa tanam, dengan bantuan pemanfaatan teknologi digital. Melalui *smart farming*, proses budidaya petani makin efektif dan akan menghasilkan produksi yang makin meningkat, dan pada akhirnya semakin meningkatkan kesejahteraan masyarakat. *Smart farming* 4.0 ini menerapkan teknologi sensor tanah dan cuaca, sehingga dapat memantau kondisi lahan secara *realtime*, dan diharapkan mampu memberikan data pertanian yang lebih terukur serta presisi[6].

Penerapan *smart farming* 4.0 bukan hanya sekedar tentang penerapan teknologi semata, tetapi kunci utamanya adalah data yang terukur. Data ini yang nantinya akan dibaca dan digunakan sebagai parameter bagi pelaku pertanian untuk mendapatkan acuan dan rekomendasi dalam praktek pertaniannya, sehingga diharapkan lebih efisien dan efektif. Jenis Teknologi Pertanian Cerdas (*Smart farming technology / SFT*) dibagi menjadi tiga kategori utama yang, seperti disebutkan di atas, mencakup sistem siklus pertanian presisi:

1. Teknologi akuisisi data: kategori ini berisi semua teknologi survei, pemetaan, navigasi, dan penginderaan.
2. Teknologi analisis dan evaluasi data: teknologi ini berkisar dari model keputusan sederhana berbasis komputer hingga manajemen pertanian yang kompleks dan sistem informasi termasuk banyak variabel berbeda.
3. Teknologi aplikasi presisi: kategori ini berisi semua teknologi aplikasi, dengan fokus pada aplikasi tingkat variabel dan teknologi panduan

Terdapat serangkaian teknologi yang dapat diklasifikasikan di setiap kategori *Smart farming technology*. Teknologi Akuisisi Data *Smart farming technology* untuk merekam dan memetakan bidang dan karakteristik tanaman dibagi ke dalam kategori di bawah ini:

1. Teknologi sistem satelit navigasi global (sebenarnya teknologi ini merekam posisi sebenarnya yang dapat digunakan untuk tujuan yang berbeda seperti panduan, pemetaan, dll.)

2. Teknologi pemetaan
3. Akuisisi data properti lingkungan
4. Mesin dan propertinya Teknologi sistem satelit navigasi global (GNSS)



Gambar 2.1 Tampilan *smart farming* 4.0

### 2.2.2 Tanaman Pakcoy

Pakcoy (*Brassica rapa L.*) merupakan jenis tanaman sayuran daun yang sangat penting di Indonesia, karena memiliki nilai ekonomis yang tinggi. Pakcoy banyak diminati sebagai sayuran karena kandungan gizi yang tinggi dan rasanya yang enak. Kandungan gizi yang terkandung dalam 100 g. Pakcoy memiliki protein 2,39 mg, lemak 0,39 mg, karbohidrat 4,09 mg, kalsium 220 mg, fosfor, 38 mg besi dan vitamin C 102 mg. Dengan kandungan tersebut Pakcoy berkhasiat untuk mencegah kanker, katarak, stroke, cacat bawaan, hipertensi dan penyakit jantung [12].

Budidaya tanaman Pakcoy bisa dilakukan secara organik maupun anorganik, namun untuk mendapatkan hasil tanaman Pakcoy yang bermutu tinggi tentunya budidaya secara organik harus segera dilakukan, mengingat semakin banyak residu yang dihasilkan dari penggunaan pupuk anorganik dan pestisida anorganik. Hal tersebut dapat mempengaruhi hasil dari produksi tanaman dan mempercepat kerusakan tanah secara fisik dan biologi. Budidaya Pakcoy secara organik dengan menggunakan *vermikompos* dapat dijadikan sebagai alternatif yang baik mengingat potensi kotoran cacing tanah sebagai bahan organik mempunyai kandungan hara cukup tinggi yaitu: C 20,20%, N 1,58%, C/N 13, P 70,30 mg/100g, K 21,80 mg/ 100g, Ca 34,99 mg/100g, Mg 21,43 mg/100g [12]

### 2.2.3 Citra

Kematangan tumbuhan biasanya ditentukan oleh beberapa parameter, diantaranya parameter ukuran, berat, ciri warna, keharuman dari tumbuhan tersebut dan lain-lain. Parameter kematangan tumbuhan dari sisi warna kulit tumbuhan itu sendiri merupakan salah satu faktor penting di dalam identifikasi kematangan tumbuhan. Umumnya, klasifikasi kematangan tumbuhan dilakukan dengan cara manual yaitu menggunakan indera penglihatan untuk membedakan kematangan itu tersendiri, namun metode tersebut memiliki banyak kelemahan seperti penilaian oleh manusia yang bersifat subyektif dan tidak konsisten[2].

Perkembangan teknologi memungkinkan identifikasi kematangan suatu tumbuhan menggunakan metode pengolahan citra. Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat *continue* seperti gambar pada monitor televisi, foto sinar X, dan lain-lain. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer [7]. Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat seperti optik berupa foto, analog berupa sinyal video seperti gambar pada monitor televisi, serta digital yang dapat langsung disimpan pada suatu pita magnetik.

### 2.2.4 Pengolahan Citra (*Image Processing*)

Citra didefinisikan sebagai dua dimensi  $f(x,y)$ , dimana parameter  $x$  dan  $y$  adalah koordinat spasial, dan  $f$  pada setiap pasangan koordinat  $(x,y)$  disebut sebagai intensitas bayangan pada titik tersebut [7]. Ketika nilai  $x,y$  dan  $f$  berhingga disebut dengan citra digital. Dengan kata lain, sebuah citra dapat didefinisikan dengan *array* dua dimensi yang secara khusus diatur dalam baris dan kolom. Citra digital terdiri dari sejumlah elemen yang terbatas, yang masing-masing elemen memiliki nilai tertentu [2]. Pengolahan citra merupakan metode untuk melakukan beberapa operasi pada gambar, guna mendapatkan gambar yang disempurnakan atau untuk mengekstrak beberapa informasi yang berguna. Pengolahan citra merupakan jenis pemrosesan sinyal di mana *input* adalah gambar dan *output* dapat berupa gambar

atau karakteristik atau fitur yang terkait dengan gambar tersebut. Saat ini, pengolahan citra adalah salah satu teknologi yang berkembang pesat [8].



Gambar 2.2 Pengolahan Citra

Pengolahan Citra (*Image Processing*) pada dasarnya mencakup tiga langkah sebagai berikut :

- a. Mengimpor gambar melalui alat akuisisi gambar
- b. Menganalisis dan memanipulasi gambar
- c. Keluaran yang hasilnya dapat berupa gambar atau laporan yang diubah berdasarkan analisis gambar.

Ada dua jenis metode yang digunakan untuk pengolahan citra yaitu, pengolahan citra analog dan digital. Pengolahan citra analog dapat digunakan untuk *hard copy* seperti cetakan dan foto. Analisis gambar menggunakan berbagai dasar interpretasi saat menggunakan teknik visual tersebut. Teknik pengolahan citra digital mampu membantu manipulasi gambar digital dengan menggunakan komputer. Tiga fase umum yang harus dilalui semua jenis data saat menggunakan teknik digital adalah pemrosesan, peningkatan dan tampilan, serta ekstraksi informasi [8].

### **2.2.5 Segmentasi Citra**

Segmentasi Citra merupakan proses dimana citra digital dipartisi menjadi berbagai subkelompok (piksel) yang disebut objek citra, dimana dapat mengurangi kompleksitas citra, sehingga dapat menganalisis citra menjadi lebih sederhana [9]. Proses segmentasi citra bersifat eksperimental, subjektif, dan tergantung pada tujuan yang ingin dicapai. Segmentasi dalam Pengolahan Citra sedang digunakan

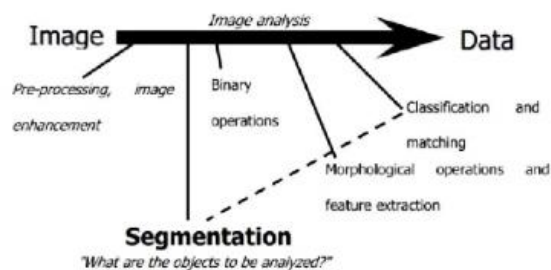
dalam industri medis untuk diagnosis yang efisien dan lebih cepat, mendeteksi penyakit, tumor, dan pola sel dan jaringan dari berbagai citra medis yang dihasilkan dari radiografi, MRI, endoskopi, termografi, ultrasonografi, dll.

Citra satelit diproses untuk mengidentifikasi berbagai pola, objek, kontur geografis, informasi tanah, yang nantinya dapat digunakan untuk pertanian, pertambangan, *geo-sensing*. Segmentasi citra memiliki area aplikasi yang sangat besar dalam robotika, seperti RPA, self-mengemudi mobil. Implementasi segmentasi gambar dalam python, matlab, dan bahasa lainnya digunakan secara ekstensif untuk proses tersebut, teknik Segmentasi Citra:

- a. *Threshold Method*
- b. *Edge Based Segmentation*
- c. *Region Based Segmentation*
- d. *Clustering Based Segmentation*
- e. *Watershed Based Method*
- f. *Artificial Neural Network Based Segmentation*

Pemrosesan citra secara umum telah diimplementasikan dalam berbagai bahasa pemrograman baik java, matplotlib, C++.

Dengan sifat dasar modularitas, implementasi dan penggunaan serbaguna dalam tumpukan ilmu data, pembelajaran mesin, dan pembelajaran mendalam, Python juga memiliki perpustakaan yang kuat untuk mengimplementasikan berbagai teknik. digunakan dalam Segmentasi Citra. Pustaka Python seperti *scikit-image*, *OpenCV*, *Mahotas*, *Pillow*, *matplotlib*, *SimpleITK* dll. Terkenal digunakan untuk mengimplementasikan pemrosesan gambar secara umum dan segmentasi citra pada khususnya.



Gambar 2.3 Ilustrasi Analisis Segmentasi Citra

### 2.2.6 *Object Detection*

*Object detection* menentukan keberadaan suatu objek dan ruang lingkungannya serta lokasi pada sebuah gambar. Hal ini dapat diperlakukan sebagai pengenalan objek kelas dua, dimana satu kelas mewakili kelas objek dan kelas lain mewakili kelas non-objek. Deteksi objek dapat dibagi lagi menjadi *soft detection* dan *hard detection*. *Soft detection* hanya mendeteksi adanya objek sedangkan *hard detection* mendeteksi adanya objek serta lokasi objek[21].

Google merilis *Tensorflow Object Detection API* untuk mempermudah pengembangan aplikasi deep learning menggunakan *Tensorflow Object Detection API*. *TensorFlow Object Detection API* adalah *platform open source* yang dapat Anda gunakan untuk mengembangkan, melatih, dan *men-deploy* model deteksi objek. Sistem ini telah banyak diterapkan di berbagai produk Google termasuk pencarian gambar, deteksi wajah dan plat nomor di Google Streetview, Google Assistant, Waymo atau *Self Driving Car* ("*Tensorflow Object Detection API*").

### 2.2.7 *Artificial Intelligence (Kecerdasan Buatan)*

*Artificial Intelligence* merupakan ilmu dan teknik pembuatan mesin cerdas, khususnya program komputer cerdas. Hal ini terkait dengan tugas yang sama dengan menggunakan komputer untuk memahami kecerdasan manusia, tetapi *Artificial Intelligence* tidak harus membatasi dirinya terhadap metode yang diamati secara biologis [24]

Menurut pengertian Dobrev yang mengatakan bahwa *Artificial Intelligence* sebagai pembelajaran bagaimana membuat komputer melakukan hal-hal yang dimana saat ini masih lebih baik dilakukan oleh manusia [25].

### 2.2.8 *Deep Learning*

*Deep learning* merupakan metode pembelajaran data yang bertujuan untuk merepresentasikan data secara bertahap dengan menggunakan sejumlah lapisan pengolahan data. Menurut LeCun, Bengio dan Hinton (2015) representasi data dibuat berdasarkan hasil algoritma pembelajaran yang tidak secara eksplisit dilakukan oleh manusia. *Deep learning* merupakan teknologi kecerdasan buatan yang mampu membuat keputusan yang akurat berdasarkan data [19]. Penggunaan



*deep learning* cocok untuk data yang kompleks dan kumpulan data yang besar. Pembelajaran mendalam membuatnya lebih mudah untuk mengidentifikasi suatu objek.

Pembelajaran mendalam dapat digunakan untuk mengklasifikasikan dan meregresi gambar, deret waktu, dan data tekstual. Algoritma *neural network* digunakan dalam teknik *deep learning* yang menggunakan *input* berupa metadata yang kemudian diolah dengan *hidden layer* transformasi *nonlinier* dari *input* data untuk mendapatkan nilai *output*. Dalam *deep learning* terdapat struktur berlapis atau jaringan syaraf tiruan (JST) *Deep learning* terdiri dari beberapa jenis yaitu *fully connected neural network*, *convolutional neural network*, *neural network recursur*, *a generative adversarial network* dan *deep reinforcement learning* [20].

Jaringan syaraf tiruan yang memiliki konsep sejumlah besar lapisan dapat ditanggihkan dari algoritma pembelajaran mesin yang ada sehingga komputer dapat belajar lebih cepat dan lebih akurat. Dengan berkembangnya *deep learning* dalam penggunaannya, sering digunakan untuk penelitian atau industri dapat membantu memecahkan banya masalah data besar seperti visi komputer, pengenalan suara, dan pemrosesan bahasa alami.

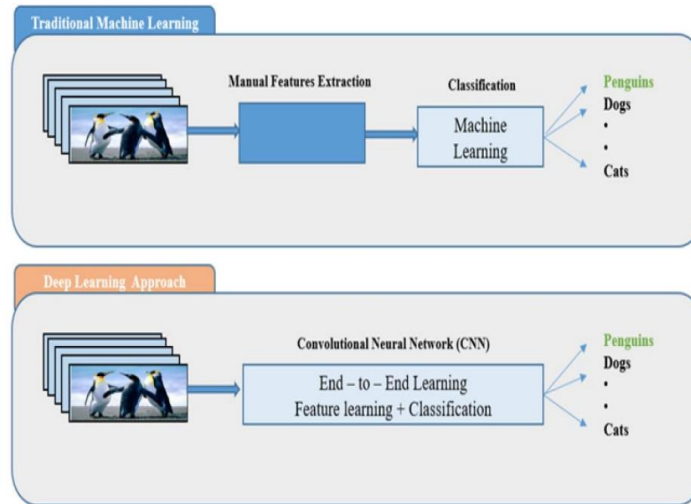
Salah satu fitur utama *Deep Learning* yang membantu mengekstraksi pola yang berguna dari data dan memudahkan untuk membedakan kelas berdasarkan model adalah rekayasa fitur, yang merupakan teknik prediksi penting untuk mendapatkan skor yang baik [5] .

### **2.2.9 Convolutional Neural Network (CNN)**

Algoritma yang digunakan dalam *Feature Engineering* dapat menemukan pola-pola umum yang penting untuk membedakan kelas-kelas dalam *deep learning*, metode CNN atau *Convolutional Neural Network* sangat efisien untuk menemukan fitur-fitur yang baik pada citra ke level selanjutnya untuk melatih asumsi *nonlinier* yang dapat meningkatkan kompleksitas suatu model. Yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama, sehingga dalam dunia *deep learning* penggunaan GPU sudah sangat umum [5].

CNN dapat secara otomatis mempelajari fitur dari data (gambar) dan mendapatkan hasil dari output. Ini adalah salah satu perbedaan utama antara CNN

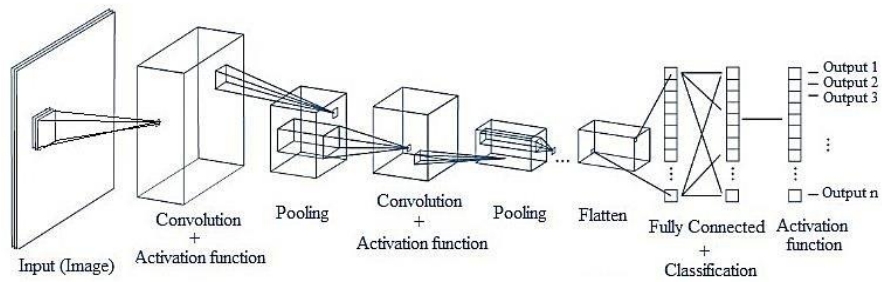
dan pembelajaran mesin sederhana. CNN dapat berjalan lebih cepat dari GPU, misalnya Nvidia Tesla K80. CNN juga mencapai kinerja tinggi dari berbagai sistem visi komputer seperti deteksi objek, pencarian pengenalan dan klasifikasi gambar [27].



Gambar 2.4 Perbedaan antara CNN dan pembelajaran mesin

Pada gambar, dapat diketahui bahwa pembelajaran mesin tradisional selalu menggunakan ekstraksi manual untuk mendapatkan fitur dari data *input* sebelum klasifikasi. Mengenai pendekatan pembelajaran yang mendalam, pembelajaran fitur dan klasifikasi saat ini di CNN, dan tidak dipisahkan untuk proses seperti pembelajaran mesin tradisional.

Pada jaringan CNN memiliki tipe *feed-forward* (tidak berulang). Arsitektur CNN terdiri dari dua bagian, yaitu *Feature Extraction Layer (Feature Learning)* dan *Fully Connected Layer (MLP)* atau klasifikasi. Dimana pada *Feature Extraction Layer* terjadi proses *encoding* dari sebuah citra menjadi *features* berupa angka yang merepresentasikan gambar tersebut. *Fully Connected Layer (MLP)* pada bagian ini digunakan untuk mengklasifikasikan data dengan mengubah *features* menjadi sebuah *vector* agar dapat dijadikan *input* dalam MLP [21]. Arsitektur yang digunakan pada CNN dapat dilihat pada gambar 2.5 berikut ini



Gambar 2.5 Tampilan Arsitektur CNN [24]

Struktur dari CNN terdiri dari *input*, proses fitur ekstraksi, proses klasifikasi dan *output*. Proses fitur ekstraksi terdiri dari beberapa lapisan tersembunyi (*hidden layer*), yaitu lapisan konvolusi, ReLu (fungsi aktivasi), dan lapisan *pooling*. Secara 13 hirarki, CNN bekerja sehingga *output* yang dihasilkan pada layer konvolusi pertama dapat digunakan pada layer konvolusi selanjutnya. Sedangkan pada proses klasifikasi terdiri dari *fully connected*, *flatten*, dan *softmax* (fungsi aktivasi) yang *output*-nya berupa klasifikasi [24].

### 2.2.10 Feature Extraction Layer

#### 1. Input Layer

*Input layer* digunakan untuk mewadahi nilai *pixels* citra yang selanjutnya akan dijadikan nilai input [21]. *Input layer* terdiri dari citra masukan dengan, lebar, tinggi, dan dimensi. Misalnya yang menjadi masukan *pixels array* adalah  $32 \times 32 \times 3$  artinya yang menjadi masukan adalah citra dengan ukuran  $32 \times 32$  dengan 3 *channel* warna RGB.

#### 2. Convolutional Layer

*Convolutional layer* adalah lapisan pertama pada *feature extraction layer*. Pada *convolutional layer* nilai *input* akan digabungkan dan di-filter lalu diteruskan ke *layer* selanjutnya. Hal ini mirip dengan fungsi respon *neuron* pada korteks. Proses filter dilakukan dengan menggeser filter yang melintasinya melalui *dot product* antara data *input* dan nilai filter sehingga diperoleh *output feature map*. Parameter *stride* yang akan menentukan pergeseran filter sepanjang *pixels* gambar. Jika nilai *stride* semakin kecil maka hasil yang diperoleh akan semakin detail [25].

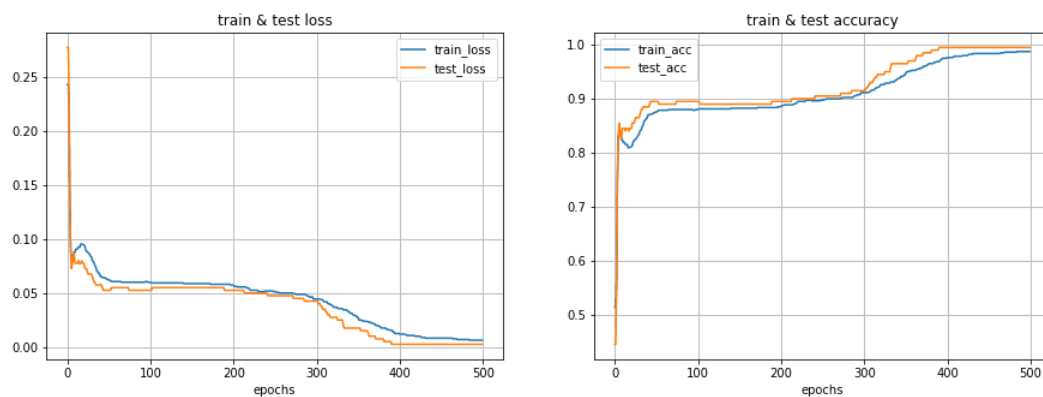
#### 3. Pooling Layer

*Pooling layer* merupakan bentuk dari *non-linear down sampling* [21]. Fungsi dari lapisan ini untuk mengurangi ukuran spasial dari representasi secara progresif, mengurangi jumlah parameter, dan mengontrol *overfitting*. Lapisan ini bekerja secara *independent* pada tiap irisan kedalaman *input* (tidak mengubah kedalaman) dan mengubah ukuran secara spasial (mengurangi tinggi dan lebar) dengan menggunakan operasi Max.

### 2.2.11 Loss dan Accuracy

*Loss* atau kerugian menunjukkan tingkat kerugian pada klasifikasi citra, dimana semakin rendah *loss* maka model yang dibangun semakin baik. Namun hal tersebut tidak berlaku jika model telah mengalami *overfitting*. Nilai *loss* didapatkan dari perhitungan pada *training* dan *validation* dan hasilnya menginterpretasikan seberapa bagus model ketika melakukan *training* dan *validation*.

Untuk nilai *loss* tidak dinyatakan dalam persen seperti *accuracy*, hal tersebut karena berasal dari penjumlahan *error* dari setiap contoh *epoch* atau *validation sets*. Pada dasarnya tujuan model pembelajaran digunakan untuk mengurangi nilai *loss* sesuai dengan parameter model dengan mengganti nilai *weight vector* melalui metode optimasi yang berbeda seperti *backpropagation* pada *Neural Network*. Jika disimpulkan maka nilai *loss* menggambarkan seberapa bagus atau buruknya suatu model setelah dilakukan iterasi dari optimasi.



Gambar 2.6 Tampilan *loss* dan *accuracy*

Pada Gambar 2.6 menunjukkan grafik hubungan antara *loss*, *accuracy* dan *epoch*. Nilai akurasi dapat diperoleh setelah mempelajari parameter model dan pelatihan lebih lanjut (tidak ada pelatihan yang dilakukan). Sampel uji kemudian dimasukkan ke dalam model, jumlah kesalahan dalam model (kehilangan lingkaran

nol) dicatat, dan kemudian dibandingkan dengan target yang sebenarnya. Kemudian tingkat kesalahan klasifikasi dihitung. Misalnya, jika jumlah sampel uji 1000 dan model yang dihasilkan memberikan nilai benar 952, maka nilai akurasi yang diperoleh adalah 95,2% [16].

### **2.2.12 Training, Validation, dan Testing**

Dalam membangun model terdapat dua istilah penting yaitu *training* dan *testing*. *Epoch* merepresentasikan proses konstruksi model sedangkan *testing* digunakan untuk menguji model yang telah dibuat. Dalam proses *training* dan *testing* diperlukan adanya *dataset* yang berupa kumpulan data (sampel dalam statistik). Untuk membangun model dan mengevaluasi model dibutuhkan adanya sampel data.

Berikut merupakan tiga jenis dataset yang tidak saling beririsan himpunannya.

- a. *Training set* yang penggunaannya dibutuhkan untuk membangun ataupun melatih model.
- b. *Validation set* dibutuhkan untuk optimasi saat training model agar model mampu mengenali pola secara generik.
- c. *Testing set* yang berfungsi untuk menguji atau testing model.

Untuk membagi dataset biasanya digunakan perbandingan rasio sebesar 80%:10%:10% atau 90%:5%:5% untuk *training* : *validation* : *testing*. Namun apabila *validation set* tidak digunakan karena ukuran dataset yang kecil, maka perbandingan rasio pembagiannya berubah menjadi 90%:10%, 80%:20%, 70%:30% atau 50%:50% untuk *epoch:testing* [17].

### **2.2.13 Confusion Matrix**

Suatu metode yang biasanya digunakan untuk membantu menghitung akurasi hasil klasifikasi pada deteksi objek. Dalam *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi.

Keempat istilah tersebut adalah *True Positive* (TP), *False Positive* (FP), *False Negative* (FN) dan *True Negative* (TN).

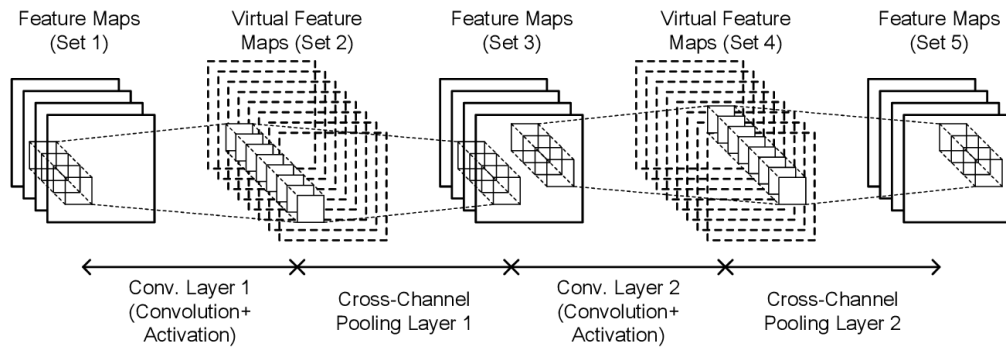
Keterangan :

1. *True Positive* (TP) adalah kondisi dimana objek manusia berhasil dideteksi oleh sistem
2. *False Positive* (FP) adalah kondisi dimana sistem menganggap objek atau kondisi lain sebagai objek manusia
3. *False Negative* (FN) adalah kondisi dimana ada objek manusia yang tidak terdeteksi oleh sistem
4. *True Negative* (TN) adalah kondisi dimana sistem mampu mengabaikan objek selain manusia Berdasarkan *confusion matrix*, dapat ditentukan nilai *Accuracy*, *Precision*, *Recall* dan *F-Score*

#### **2.2.14 Efficient Net Model Type**

*Efficient Net*, merupakan salah satu *state of art* dengan akurasi mencapai 84,4% dengan parameter 66M. Dalam masalah klasifikasi *ImageNet*, dapat dianggap sebagai kelompok model CNN. Grup *Efficient Net* terdiri dari 8 model antara B0 dan B7, dan seiring bertambahnya nomor model, jumlah parameter yang dihitung tidak meningkat banyak, sementara akurasi meningkat secara signifikan. Tidak seperti model CNN lainnya, *Efficient Net* menggunakan fungsi aktivasi baru yang disebut *Swish* fungsi aktivasi *Rectified Linear Unit* (ReLU) [14].

Tujuan dari arsitektur *deep learning* adalah untuk mempresentasikan hasil dengan lebih efisien dan dengan pendekatan model yang lebih kecil. *Efficient Net* tidak seperti *state-of-art* model lainnya, mencapai hasil yang lebih efisien dengan kedalaman penskalaan yang seragam, lebar, dan resolusi sambil memperkecil model. Langkah pertama dalam metode penskalaan majemuk adalah mencari grid untuk menemukan hubungan antara dimensi penskalaan yang berbeda dari jaringan dasar di bawah kendala sumber daya tetap. Dengan cara ini, faktor penskalaan yang cocok untuk kedalaman, lebar, dan dimensi resolusi dapat ditentukan. Koefisien ini kemudian diterapkan untuk menskalakan jaringan *baseline* ke target jaringan yang diinginkan [14].



Gambar 2.6 Model Dasar *Efficient Net* [14]

Pada gambar 2.6, merupakan model dasar dari *Efficient Net* yang berawal dari Conv 3x3 dan berakhir di MBC v6 3x3. Blok bangunan utama untuk *Efficient Net* adalah *bottleneck* yang merupakan kebalikan (*inverse*) dari MBConv, yang pertama kali diperkenalkan di *Mobile Net V2*. Tetapi karena peningkatan nilai FLOPS (*floating 31 point operations per second*), blok ini tetap digunakan meskipun sedikit lebih banyak daripada *Mobile Net V2*.

Di MBCONV, *block* terdiri dari lapisan pertama yang mengambang kemudian menempatkan saluran, jadi koneksi langsung digunakan antara *bottleneck* yang menghubungkan saluran lebih sedikit daripada lapisan ekspansi. Arsitektur ini memiliki kedalaman konvolusi yang dapat dipisahkan dan mengurangi perhitungan dengan hampir k2 faktor dibandingkan dengan lapisan tradisional dimana k adalah ukuran kernel yang menunjukkan lebar dan tinggi jendela konvolusi 2D [14].

### 2.2.15 Python

*Python* merupakan bahasa pemrograman dengan tujuan umum yang dikembangkan secara khusus untuk membuat *source code* mudah dibaca. *Python* juga memiliki *library* yang lengkap sehingga memungkinkan *programmer* untuk membuat aplikasi yang mutakhir dengan menggunakan *source code* yang tampak sederhana [26].

### 2.2.16 Tensorflow js

*Tensorflow* merupakan perpustakaan perangkat lunak yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian Mesin Cerdas Google, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam.

*Tensorflow* menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan. Fitur utamanya meliputi:

1. Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan *array multidimension (tensors)*.
2. Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin.
3. *Graphics Processing Unit (GPU)* transparan menggunakan, mengotomatiskan manajemen dan optimalisasi memori yang sama dan data yang sama digunakan.

### **2.2.17 Android Studio**

*Android Studio* adalah *Software Intregated Development Environment* (IDE) untuk pengembangan aplikasi sistem operasi Android. Pada Android Studio bahasa pemrograman yang digunakan adalah Java dan Kotlin. Android Studio memiliki banyak fitur yang berguna untuk pengembangan aplikasi android. Beberapa fitur yang membantu pembuatan aplikasi Android seperti Gradle yang fleksibel, *Emulator* Android, dan lingkungan pengembangan yang menjadi satu bisa untuk semua macam Android [26]

Android Studio adalah sebuah *environment* yang dibuat untuk membantu *developer* dalam mengembangkan aplikasi yang berbasis android. Android Studio menawarkan banyak fitur untuk meningkatkan produktivitas pengguna saat membuat aplikasi Android, seperti [20] :

- a. Sistem versi berbasis Gradle yang fleksibel
- b. Emulator yang cepat dan kaya fitur
- c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- d. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- e. Template kode dan *integrasi* GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh



- f. Alat pengujian dan kerangka kerja yang ekstensif

### 2.2.18 Java

Java merupakan salah satu bahasa pemrograman yang diciptakan oleh James Gosling pada tahun 1991 dengan nama pertama yaitu Oak dan berganti pada tahun 1995 menjadi Java. Bahasa pemrograman Java adalah salah satu bahasa pemrograman dengan paradigma *Object Oriented Programming* (OOP). Java mampu membuat sebuah aplikasi mulai dari *mobile*, *desktop* dan *web* dan mampu dijalankan pada sistem operasi Linux, Windows, DOS, dan Unix, selain itu banyak kelebihan bahasa pemrograman Java yaitu sederhana, berorientasi objek, mudah dalam distribusi, *interpreter*, aman, *architecture neutral*, *portable*, *multithreaded*.

### 2.2.19 Data Augmentation

Salah satu cara untuk menghadapi *overfitting* adalah dengan mengumpulkan banyak data. Pada data gambar biasanya hanya satu gambar yang tersedia untuk masing-masing data. Tetapi dengan adanya *data augmentation*, data dapat diperbanyak dengan menggunakan beberapa teknik transformasi gambar. *Data augmentation* akan menghasilkan gambar baru dari data yang sudah ada dengan menerapkan beberapa jenis transformasi pada data gambar dan menggunakannya sebagai data tambahan pada pelatihan [27].

Beberapa transformasi yang biasa digunakan sebagai berikut :

- a. Menggeser gambar dengan beberapa *pixel* secara horizontal atau vertikal
- b. Memutar gambar
- c. Mengubah kecerahan
- d. Memperbesar gambar
- e. Mengubah kontras
- f. Memotong gambar

Dengan menambah data, maka dapat membantu dalam mengurangi *overfitting* saat melatih model *machine learning*