

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu meliputi pembahasan yang serupa atau terkait dengan penelitian sebelumnya, baik untuk membedakannya dari apa yang telah dilakukan sebelumnya atau untuk menyempurnakan pekerjaan sebelumnya dalam rangka menemukan bidang yang masih kurang pengetahuannya dan belum dibahas dalam penelitian.

Tabel 2.1 Penelitian Terdahulu

NO	Judul Penelitian (Author, Tahun)	Abstrak	Metode	Kelebihan dan Kekurangan
1	Aplikasi Diagnosa Penyakit Kulit Menggunakan dengan Metode Convolutional Neural Networks (Widya dkk., 2023)	Penyakit yang paling umum beriklim tropis adalah penyakit kulit. Karena tidak berbahaya jadi masyarakat enggan untuk memeriksa kulit nya. Namun penyakit kulit juga dapat menyebar dan sulit diobati	<i>Convolutional Neural Network (CNN)</i>	Pada penelitian ini model konvolusional 4 layer digunakan untuk menganalisis hasil diagnostik dari kumpulan data di Edinburgh Dermofit Image Library. Aplikasi yang dikembangkan mencapai akurasi 100% pada data pelatihan dan pengujian.
2	Deteksi Objek pada Citra Makanan Sebagai Rekomendasi Diet Menggunakan	Mengidentifikasi pada foto jenis makanan seperti hamburger, ayam goreng, seblak, bakso	Mark R- (CNN) dan Faster Mask R- (CNN)	Pada penelitian ini telah mengimplementasi dari proses deteksi objek pada citra makanan dengan tingkat akurasi sebesar

	Metode Mask R-(CNN)(Ratri Enggar Pawening dkk., 2023)	aci, dan bakwan. Dengan berfungsi mengetahui makanan yang membantu menjaga tubuh tetap sehat dan menjamin metabolism berjalan dengan lancar		72% dengan menggunakan Mask R-(CNN). Namun dalam proses penelitian ini memiliki kekurangan dari ketersediaan dataset.
3	Prediksi Citra Makanan Menggunakan <i>Convolutional Neural Network</i> Untuk Menentukan Besaran Kalori Makanan (Darma Udayana & Cipta Nugraha, 2020)	Identifikasi foto makanan yang sering dimakan oleh masyarakat Indonesia sebagai pembelajaran dengan berbasis aplikasi <i>mobile</i>	<i>Convolutional Neural Network</i> (CNN)	Pada penelitian ini dengan klasifikasi citra makanan dengan metode (CNN) dengan hasil tingkat akurasi sebesar 66% sampai 98%. Dari Hasil akurasi memiliki akurasi yang rendah pada klasifikasi karena memiliki kemiripan sehingga sistem salah dalam mengklasifikasi pada makanan.
4	Tinjauan Literatur: Named Entity Recognition pada Resep Makanan Indonesia (Eko Saputro & Fathan Hidayatullah, t.t.)	Identifikasi studi literatur dari penerapan <i>Named entity recognition</i> (NER) pada resep makanan. Dari literatur dilakukan analisis dari metode, jenis entitas dan	Studi literatur dari metode <i>Named Entity Recognition</i>	Pada penelitian dapat mengkaji 5 literatur terkait NER. Dari 2 pendekatan literatur yang digunakan yaitu <i>deep-learning</i> dan <i>rule-based</i> . Dari segi performa rule-based lebih baik. Dari model yang sering digunakan yaitu Foodle dengan menggunakan dataset

		Bahasa yang digunakan		Entitas yang ditemukan meliputi nama, status, unit, kuantitas, ukuran, suhu, status kering atau segar, judul, kategori, bahan, langkah-langkah, makanan pembuka dan camilan, sarapan dan makan siang, makanan penutup, serta makan malam dan minuman.
5	PENYEBAB DAN PENCEGAHAN ALERGI(Abriyani dkk., 2022)	Identifikasi penyebab dari alergi yang sedikit diketahui oleh masyarakat dan pencegahan yang dilakukan.	Penyuluhan dalam bentuk informasi dalam bentuk materi	Pada penelitian ini masyarakat dapat memahami penyebab alergi dan dapat mencegah timbulnya alergi. Pada masyarakat juga mengabdi dalam meningkatkan pemahaman pada Kesehatan. Dalam pengabdian selanjutnya dapat memberikan informasi alergi agar berjalan secara berkesinambungan.
6	Fakultas Ilmu Komputer Pengembangan Aplikasi Deteksi Allergen pada Makanan Menggunakan <i>Convolutional Neural</i>	Alergi Makanan seringkali sebagai masalah Kesehatan di dunia. Dari wawancara pada pengidap alergi makanan ini masih kesulitan dalam	<i>Convolutional Neural Network</i> (CNN)	Pada Penelitian sistem berjalan sesuai yang diinginkan dimana, memiliki 10 dataset makanan dengan pengujian usability scale dengan 22 responden, Hasil skor rata-rata 84,2. Hal ini memenuhi kebutuhan

	<i>Network Berbasis Android</i> (Mahe swara dkk., 2017)	mengetahui kandungan yang ada pada makanan.		usability diatas 68. Hasil pengujian dengan ahli gizi menunjukkan bahwa keakuratan deteksi makanan mendapat persentase 76% dan keakuratan Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer alergen yang terkandung pada makanan mendapat persentase 70%. Namun perlu adanya peningkatan pada akurasi deteksi makanan.
7	Pengenalan Makanan Tradisional Nusantara dengan Menggunakan Metode <i>Convolutional Neural Network</i> (CNN)(Mahapu tri dkk., 2022)	Makanan tradisional diidentifikasi dengan adanya perbandingan dari 3 model (CNN). Hal ini diperlukan dalam <i>preprocessing</i> dalam <i>input</i> data pada citra makanan dengan <i>cropping</i> , <i>wrapping</i>	<i>Convolutional Neural Network</i> (CNN)	Penggunaan (CNN) untuk mempermudah mengidentifikasi gambar makanan tradisional. Namun perlu Keterbatasan Dataset dan ketergantungan Koneksi Internet menjadi suatu masalah di penelitian ini

Dari review penelitian terdahulu diatas, jadi research-gap nya adalah

1. Deteksi Berfokus pada pengenalan makanan.
2. Penggunaan *Convolutional Neural Network (CNN)* dengan model arsitektur *Mobilenetv2* untuk mendapatkan model *training* yang paling stabil.
3. Pada penelitian berfokus pada bahan makanan yang dapat mengandung alergi.

2.2 Teori Terkait

Teori dapat diartikan dengan sekumpulan proporsi yang berhubungan dan dapat diuji secara empiris (Marliyah, 2021) Teori terkait adalah elemen fundamental yang digunakan untuk menjelaskan fenomena atau perilaku dalam penelitian. Teori ini menyediakan struktur konseptual yang memudahkan dalam memahami, menginterpretasi, dan diprediksi berbagai aspek terkait subjek yang diteliti. Penggunaan teori-teori ini membantu peneliti dalam merumuskan hipotesis, mengembangkan metodologi penelitian, dan menganalisis data dengan lebih terstruktur dan fokus. Selain itu, teori terkait memungkinkan peneliti untuk mengaitkan temuan penelitian mereka dengan literatur yang ada, sehingga dapat berkontribusi pada pengembangan pengetahuan yang lebih komprehensif.

2.2.1 Allergen Makanan

Makanan adalah substansi yang diperlukan oleh organisme hidup untuk menyediakan energi dan nutrisi yang esensial bagi pertumbuhan, pemeliharaan fungsi tubuh, serta perbaikan jaringan. Nutrisi dalam makanan seperti protein, karbohidrat, lemak, vitamin, dan mineral, semuanya berperan penting dalam menjaga keseimbangan kesehatan tubuh. Setiap orang memiliki kebutuhan nutrisi yang berbeda, namun makanan juga dapat menjadi sumber risiko bagi individu yang memiliki sensitivitas atau alergi terhadap zat tertentu di dalamnya. Di sinilah peran alergen makanan menjadi signifikan.

Alergen makanan adalah protein yang terkandung dalam makanan dan mampu memicu reaksi imun tubuh pada individu yang peka. Hal tersebut menjadi suatu masalah bagi manusia pada Kesehatan yang dipengaruhi pada semua umur(Dewanty dkk., 2022). Sistem kekebalan tubuh orang yang alergi akan bereaksi terhadap alergen ini sebagai ancaman, meskipun bagi kebanyakan orang, komponen tersebut tidak menimbulkan masalah. Ketika seseorang dengan alergi makanan mengonsumsi hidangan yang mengandung alergen ini, tubuhnya akan memproduksi antibodi IgE yang memicu pelepasan histamin, menyebabkan berbagai gejala alergi. Reaksi ini bisa bervariasi dari ringan, seperti gatal-gatal atau ruam, hingga berat, seperti kesulitan bernapas dan anafilaksis. Berikut adalah tabel beberapa makanan dan potensi alergen yang terkandung serta reaksi yang mungkin terjadi:

Tabel 2.2 Jenis Makanan yang memiliki potensi alergi

NO	Makanan	Bahan Potensial Alergi	Reaksi yang Mungkin Terjadi
1	Ayam Betutu	Ayam (protein hewani), rempah-rempah seperti bawang putih	Ruam, gatal, pembengkakan, gangguan pencernaan
2	Sate Lilit	Ikan atau ayam (protein hewani), santan	Ruam, gatal, pembengkakan, kesulitan bernapas
3	Lawar	Daging babi atau ayam (protein hewani), kelapa, rempah-rempah	Ruam, gangguan pencernaan, gatal, reaksi anafilaksis
4	Nasi Jinggo	Ayam (protein hewani), sambal (cabai)	Gangguan pencernaan, gatal di mulut, ruam kulit
5	Babi Guling	Daging babi (protein hewani), bumbu rempah-rempah	Ruam, pembengkakan, gangguan pencernaan

6	Kerak Telor	Telur (protein hewani), kelapa parut	Ruam, pembengkakan mual, gatal, wajah,
7	Soto Betawi	Daging sapi (protein hewani), santan	Gatal, gangguan pencernaan, ruam kulit
8	Kue Cucur	Tepung terigu (gluten), santan	Ruam, gatal, gangguan pencernaan
9	Rawon	Daging sapi (protein hewani), bumbu seperti bawang putih	Ruam, gangguan pencernaan (diare), pembengkakan, kesulitan bernapas
10	Pecel	Kacang tanah dalam bumbu pecel	Gatal, ruam, pembengkakan wajah atau tenggorokan, anafilaksis
11	Rujak Cingur	Kacang tanah, udang dalam petis	Gatal, ruam, anafilaksis
12	Lontong Balap	Kecambah, tahu (kedelai), lenthong (kacang-kacangan)	Ruam, gangguan pencernaan, pembengkakan wajah
13	Sate Padang	Daging sapi atau ayam (protein hewani), tepung beras (gluten)	Ruam, gatal, pembengkakan, mual
14	Kalio Ayam	Ayam (protein hewani), santan	Ruam, gangguan pencernaan, gatal
15	Ayam Taliwang	Ayam (protein hewani), cabai	Ruam, gatal, pembengkakan, gangguan pencernaan
16	Plecing Kangkung	Sambal plecing (cabai, terasi/udang), kangkung	Ruam, gatal di mulut, gangguan pencernaan

Gejala alergi dapat muncul beberapa menit hingga beberapa jam setelah mengonsumsi makanan ini. Gejala ringan seperti gatal-gatal atau ruam kulit mungkin dapat dikelola dengan obat antihistamin, namun reaksi yang lebih serius, seperti kesulitan bernapas atau anafilaksis, memerlukan penanganan medis segera, termasuk penggunaan epinefrin.

Dengan banyaknya variasi bahan dalam makanan Indonesia, penting bagi individu yang memiliki alergi makanan untuk waspada terhadap bahan-bahan yang digunakan dalam makanan jadi dan memastikan makanan tersebut aman untuk dikonsumsi. Edukasi tentang bahan-bahan makanan yang sering digunakan, membaca label makanan (jika tersedia), dan bertanya langsung kepada penjual atau koki mengenai komposisi makanan dapat membantu mencegah insiden alergi.

2.2.2 Artificial Intelligence

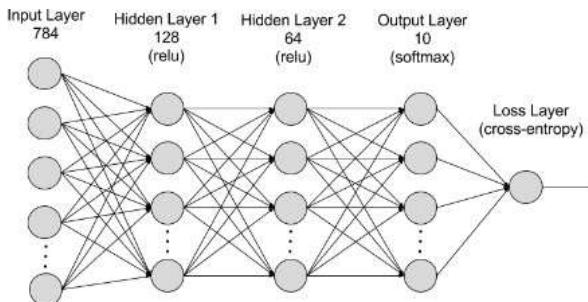
Artificial Intelligence atau dapat dikenal dengan kecerdasan buatan merupakan ilmu yang mempelajari kecerdasan dari manusia dengan memecahkan masalah dari mesin berupa model yang berpikir selayaknya manusia. Kecerdasan buatan sering diterapkan dan ditemui dalam kehidupan sehari-hari (Anggeli & Agung, 2021). Tujuan dari kecerdasan buatan dapat meniru pola pikir manusia. Penerapan dari kecerdasan buatan itu sendiri mencangkup pemrosesan Bahasa alami, sistem pakar, Computer vision dan pengenalan suara. Namun dari pencapaian kecerdasan buatan yang berkembang, konsepnya masih menjadi perdebatan dari kalangan peneliti. Sehingga memberikan upaya dengan dimensi kesadaran kepada mesin terus berlanjut. Salah satunya dengan melalui Jaringan Saraf Tiruan (JST). JST itu sendiri terinspirasi cara kerja otak manusia dengan model matematis dalam mengelola informasi, beradaptasi dan belajar dari pengalaman.

2.2.3 Deep Learning

Deep Learning merupakan bagian dari machine learning yang memiliki keterkaitan dengan kecerdasan buatan. *Deep Learning* telah

menyoroti keberhasilan pengguna dalam pengolahan citra sehingga menarik peneliti dalam beberapa tahun terakhir. *Deep Learning* berfokus pada penggunaan data dan algoritma untuk meniru cara manusia belajar, secara bertahap meningkatkan akurasinya (Azmi dkk., 2023). Model dari *Deep Learning* sendiri dengan mengenali pola dalam gambar, suara dan teks secara kompleks bertujuan menghasilkan wawasan dan akurasi pada prediksi. Pada metode ini juga mengontrol tindakan yang berkaitan dengan kecerdasan buatan, seperti mengimplementasikan gambar, menukar file suara menjadi teks tidak dengan campur tangan manusia.

Cara kerja dari *Deep Learning* sama hanya dengan kerja otak manusia dengan memiliki jutaan *neuron* dimana dengan mempelajari, menganalisis dan mengimplementasi informasi. Dengan hal ini *Deep Learning* memiliki lapisan *neuron* buatan yang bekerja sama di dalam komputer. *Neuron* ini berupa simpul sebagai modul perangkat lunak dengan menggunakan sebagai perhitungan secara matematika untuk memproses data. Jaringan saraf buatan sendiri menggunakan simpul tersebut sebagai memecahkan masalah. Inilah jaringan saraf *multi-layer* dapat dilihat pada gambar dibawah ini.



Gambar 2.1 *Multi-layer Neural Network*

Deep Learning mempunyai fungsi aktivasi dengan ketentuan dimana suatu *neuron* dapat diaktifkan atau tidak dan mengubah *output* menjadi *non-linier*. Fungsi aktivasi terdiri dari

1. *Relu*

Relu merupakan fungsi aktivasi yang popular karena sederhana dan efisien. Hal ini karena mengambil dari nilai maksimal antara 0 dan *input* itu sendiri, Sehingga *input negative* akan berubah menjadi nol dan *input* negative akan tetap. Dari persamaan dari *Relu* berupa:

$$f(x) = \max(0, x) \quad (2.1)$$

Gambar 2.2 Persamaan *Relu*

2. *Sigmoid*

Sigmoid memiliki sifat sederhana yang dimana *non-linier* sebagai *input* dan memberikan probabilitas antara 0 atau 1. Dari sifat *non-linier* itulah membuat pola menjadi kompleks. Namun dari *output* yang mendekati 0 atau 1 gradien menjadi kecil yang menyebabkan model sulit belajar. Persamaan *Sigmoid* sebagai berikut:

$$f(x) = \max(0, x) \quad (2.2)$$

Gambar 2.3 Persamaan *Sigmoid*

3. *Softmax*

Softmax merupakan fungsi aktivasi dengan mengubah vector angka ke probabilitas yang menjumlahkan sampai satu. Fungsi *softmax* dengan mengambil *input* dari setiap *neuron/layer* dihitung dengan menggunakan fungsi eksponensial. Persamaan dari *softmax* sebagai berikut:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (2.3)$$

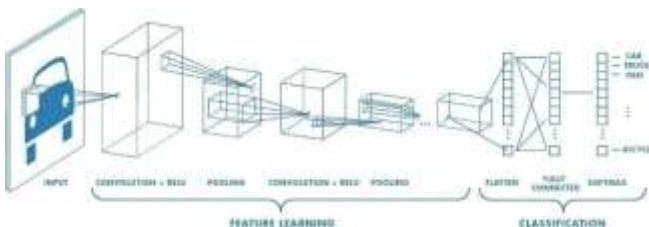
Gambar 2.4 Persamaan Soft Max

2.2.4 Klasifikasi

Klasifikasi adalah metode untuk mengelompokkan data ke dalam berbagai kelas atau kategori yang telah ditentukan sebelumnya. Dalam proses klasifikasi, model dibangun untuk mengenali pola dalam data pelatihan sehingga dapat memprediksi kelas yang benar untuk data baru. Teknik ini banyak diterapkan dalam bidang machine learning dan statistik, dengan tujuan seperti deteksi objek dalam gambar, penyaringan email spam, hingga mendiagnosis penyakit. Setiap kelas pada klasifikasi memiliki nilai diskrit dan tidak berhubungan dengan urutan atau skala nilai. Proses klasifikasi terbagi menjadi dua tahap utama: tahap pertama adalah pelatihan, di mana model dilatih untuk mempelajari pola dari data; dan tahap kedua adalah penerapan, dimana model yang telah terbentuk digunakan untuk memprediksi kelas atau label dari data yang baru.

2.2.5 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) atau dapat dikenal dengan salah satu jenis jaringan saraf tiruan sering digunakan dalam menganalisis dan memproses pada suatu gambar. (CNN) sangat efektif dalam pemrosesan gambar dibandingkan dengan jenis *neural network* lainnya(Maheswara dkk., 2017). (CNN) bekerja dengan sejumlah lapisan yang bertugas mengekstraksi berbagai fitur dari gambar. Setiap lapisan dalam jaringan ini melakukan transformasi pada data, memungkinkan jaringan secara otomatis mengenali elemen-elemen seperti pola, tepi, dan tekstur. Struktur berlapis yang kompleks membuat (CNN) mampu mendeteksi dan mempelajari karakteristik gambar dengan cara *non-linier*, sehingga mampu menjalankan tugas seperti klasifikasi, segmentasi, dan deteksi objek dengan tingkat akurasi yang tinggi.



Gambar 2.5 Cara kerja (CNN)

Dari gambar diatas dapat diketahui bahwa proses *Convolutional Neural Network* sama hanya dengan melihat seekor kucing, pandangan mata tidak langsung ke kucing. Dalam hal ini otak melakukan proses informasi visual secara bertahap dengan dimulai dari garis, tepi, tekstur, hingga dalam bentuk kucing. Hal ini yang (CNN) bekerja dengan cara yang mirip. Dari cara kerja (CNN) sebagai salah satu keunggulan dimana Efisien dalam memproses data gambar, (CNN) dapat mempelajari fitur-fitur yang penting dengan otomatis dan tidak harus melakukan ekstraksi fitur manual. Karena keunggulannya ini, (CNN) banyak diterapkan dalam berbagai aplikasi yang melibatkan data visual, seperti klasifikasi gambar, deteksi objek, pengenalan wajah, hingga segmentasi gambar, membuatnya menjadi pilihan utama dalam pemrosesan data gambar pada kecerdasan buatan.

2.2.5.1 Convolutional layer

Convolutional layer adalah sebuah proses dalam manipulasi citra yang menggunakan filter eksternal (mask) untuk menciptakan citra baru. Tujuan utamanya adalah untuk mengekstraksi fitur-fitur penting dari gambar—seperti tepi, warna, atau orientasi gradien—melalui operasi yang disebut konvolusi (Iskandar Mulyana dkk., t.t.). Proses ini bekerja dengan menggeser sebuah filter, yang memiliki bobot dan bias yang sama di setiap posisi, ke seluruh area citra. Hasil dari operasi ini adalah sebuah "peta fitur" (feature map), yaitu citra baru yang menonjolkan fitur-fitur spesifik yang dideteksi oleh filter dari citra aslinya. *Convolutional layer* memiliki dua komponen penting:

1. *Stides*

Strides merupakan sebuah parameter kunci dalam operasi konvolusi yang mengatur jarak lompatan sebuah filter (kernel) ketika menyusuri citra masukan. Parameter ini mendikte seberapa besar pergeseran filter pada sumbu horizontal dan vertikal dalam setiap langkahnya.

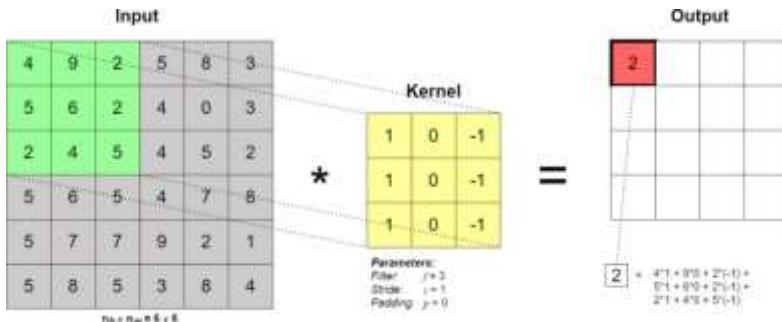
Sebagai ilustrasi, *stride* dengan nilai 1 akan menggerakkan filter secara bertahap, satu piksel demi satu piksel. Hal ini menghasilkan sebuah *feature map* keluaran yang ukurannya sangat detail dan hampir identik dengan ukuran citra masukan. Sebaliknya, jika *stride* diatur ke nilai 2, filter akan melompat sejauh dua piksel dalam sekali gerak. Akibatnya, dimensi *feature map* yang dihasilkan akan menyusut secara spasial, menjadi kira-kira setengah dari ukuran aslinya.

Penggunaan *stride* yang lebih besar secara signifikan mengurangi ukuran *feature map* keluaran, yang berarti jumlah parameter dan beban komputasi juga menurun. Namun, kelemahannya adalah potensi kehilangan informasi detail karena proses pengambilan sampel menjadi lebih jarang. Jadi, terdapat pilihan antara menggunakan *stride* kecil untuk mendapatkan hasil konvolusi yang kaya detail dengan konsekuensi beban komputasi yang berat, atau menggunakan *stride* besar untuk proses yang lebih efisien namun dengan hasil yang kurang detail.

2. *Padding*

Padding adalah sebuah metode untuk menambahkan bingkai buatan, biasanya berisi angka nol (*zero padding*), di sekeliling gambar input sebelum filter diaplikasikan. Tujuan utama dari teknik ini adalah untuk mengendalikan dimensi peta fitur (*feature map*) yang dihasilkan dan untuk mencegah ukuran gambar menyusut setelah melewati setiap lapisan. Ada beberapa jenis *padding*: *Same Padding* secara spesifik menghitung jumlah bingkai yang dibutuhkan agar dimensi output sama dengan input, yang sangat berguna untuk memastikan informasi di bagian tepi gambar tidak hilang. Sebaliknya, *Valid Padding* berarti tidak ada bingkai yang ditambahkan sama sekali, sehingga filter hanya bekerja pada area "*valid*" di dalam gambar asli, yang akibatnya membuat ukuran output menjadi lebih kecil. Peran *padding* menjadi sangat penting dalam arsitektur jaringan yang dalam, karena ia membantu mempertahankan ukuran spasial gambar dan

memastikan fitur-fitur di seluruh bagian citra, termasuk di tepian, tetap dipertimbangkan selama proses.



Gambar 2.6 Operasi Konvolusi menggunakan *Zero Padding*

Dari Proses konvolusi pada gambar diatas ,dapat dihitung berdasarkan formula:

$$\frac{(N - F + 2P)}{S} + 1 \quad (2.4)$$

N = Ukuran spasial

F = Kernel *Filter*

P = *Padding*

S = *Stride*

Secara keseluruhan, jika *input* pada sebuah *Convolutional layer* adalah citra berukuran $W1 \times H1 \times D1$, maka *output* dari *layer* tersebut akan berupa citra baru dengan ukuran $W2 \times H2 \times D2$, yang dapat dihitung menggunakan formula berikut:

$$W2 = \frac{(W1 - F + 2P)}{S} + 1 = H2 \quad (2.5)$$

$$(P = \frac{F - 1}{2}) \quad (2.6)$$

F = Ukuran Spasial *Filter* (lebar/tinggi).

S = *Stride*

P = *Padding*.

Beberapa hal penting yang perlu diperhatikan pada layer konvolusi adalah:

1. Ketebalan filter disesuaikan dengan ketebalan (depth) atau jumlah saluran dari citra input, sehingga dapat memproses seluruh informasi pada setiap saluran.
2. Ukuran filter (F) umumnya bernilai ganjil, karena secara intuitif lebih seimbang—mampu mencakup area kiri dan kanan objek dengan proporsi yang sama.
3. Semua filter dalam satu layer konvolusi memiliki ukuran yang seragam untuk menyederhanakan proses komputasi. Jumlah filter (K) biasanya dipilih sebagai kelipatan 2, karena banyak pustaka pemrograman telah mengoptimalkan perhitungan untuk ukuran tersebut guna meningkatkan efisiensi.
4. Zero padding sering diterapkan agar ukuran output secara spasial tetap sama dengan input, sehingga informasi tepi citra tetap terjaga selama proses konvolusi.

2.2.5.2. Pooling Layer

Pooling layer merupakan bagian penting dari arsitektur *Convolutional Neural Network* (CNN) yang berfungsi untuk mengurangi dimensi spasial dari *feature maps* yang dihasilkan oleh *layer* konvolusi. Pengurangan ini dilakukan melalui proses *down-sampling* seperti *max pooling* atau *average pooling*, dengan tujuan utama menurunkan jumlah parameter dan beban komputasi dalam jaringan, sekaligus mengurangi risiko *overfitting*.

Pooling layer menerima output dari *Convolutional layer* sebagai *input*, lalu menghasilkan *feature maps* yang lebih kecil, yang tetap mempertahankan informasi penting namun dalam bentuk yang lebih ringkas. Dua jenis metode pooling yang umum digunakan adalah:

1. *Max Pooling*

Metode ini bekerja dengan memilih nilai aktivasi tertinggi dari setiap area kecil (*receptive field*) dalam feature map. Hanya nilai terbesar dari setiap area yang diteruskan ke tahap selanjutnya, sehingga informasi paling menonjol dapat dipertahankan. *Max pooling* sering digunakan karena kemampuannya untuk menyoroti fitur paling penting.

2. *Average Pooling*

Berbeda dengan max pooling, average pooling menghitung nilai rata-rata dari semua aktivasi dalam setiap bidang reseptif lokal. Semua nilai dijumlahkan, lalu dibagi dengan jumlah elemen, sehingga hasilnya mencerminkan informasi rata-rata dari area tersebut. Metode ini cenderung menghasilkan representasi yang lebih halus dan stabil, serta kurang sensitif terhadap fluktuasi lokal.

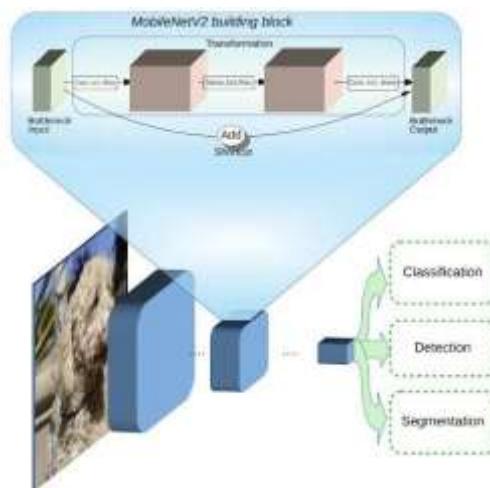
2.2.5.3 Fully Connected Layer

Feature map yang dihasilkan dari proses ekstraksi fitur memiliki bentuk array multidimensi. Agar data ini dapat diproses lebih lanjut oleh *fully connected layer*, diperlukan tahap *reshape* untuk mengubahnya menjadi vektor satu dimensi. Proses ini dikenal sebagai *flattening*, yang berfungsi untuk mengkonversi *feature map* menjadi bentuk vektor sehingga dapat digunakan sebagai input pada *fully connected layer*. Dengan demikian, *flattening* menjadi tahap penting sebagai penghubung antara bagian konvolusional dan bagian klasifikasi dalam arsitektur CNN.

2.2.6 MobileNetV2

Arsitektur *Convolutional Neural Network* (CNN) yang ringan, *MobileNetV2*. Secara khusus pada perangkat dengan keterbatasan sumber daya perangkat. Arsitektur ini memiliki kemampuan dengan memberikan

performa tinggi sambil menjaga kestabilitasan dari ukuran maupun akurasi model dalam komputasi. Dari kemampuan yang diperoleh dapat dijadikan dengan Arsitektur (CNN) lainnya. Arsitektur *Mobilenetv2* terdiri atas serangkaian lapisan konvolusional dengan diikuti oleh konvolusi yang dapat dipisahkan berdasarkan kedalaman, residu terbalik, desain kemacetan, kemacetan *linier*, dan *blok squeeze-and-excitation (SE)*. Pada komponen tersebut bekerja sama untuk mengurangi jumlah parameter dan komputasi yang diperlukan sambil mempertahankan kemampuan model untuk menangkap fitur yang kompleks.



Gambar 2.7 *Mobilenetv2*

Pada gambar diatas bekerja dengan mengambil gambar sebagai *input*, lalu memprosesnya melalui serangkaian lapisan untuk mengenali fitur-fitur penting. Pertama, gambar diperkecil ukurannya melalui lapisan konvolusi dan *pooling*. Kemudian, bagian inti *Mobilenetv2*, yaitu *blok bottleneck* dan *depthwise separable convolution*, berfungsi mengekstraksi fitur-fitur dengan lebih efisien. *Blok* ini membantu model mempelajari pola-pola dalam gambar dengan mengurangi ukuran data tanpa kehilangan informasi penting. Setelah itu, hasilnya diproses di lapisan *fully connected*

untuk menghubungkan semua fitur yang telah dipelajari, lalu melewati lapisan *Softmax* untuk menentukan kategori gambar. Dengan cara ini, *Mobilenetv2* dapat mengklasifikasikan gambar dengan cepat dan efisien, bahkan saat dijalankan di perangkat seluler. Ini memastikan aplikasi tidak hanya efektif dalam mengidentifikasi alergen, tetapi juga dapat diakses dengan nyaman oleh pengguna sehari-hari pada berbagai perangkat.

2.2.7 *Black Box Testing*

Pengujian *Black box* cara menguji program tanpa melihat atau memeriksa kode sumbernya. Metode ini berfokus pada bagian luar perangkat lunak, seperti bagaimana ia menerima masukan dan menghasilkan keluaran. Pengujian ini memeriksa apakah sistem berfungsi sebagaimana mestinya berdasarkan persyaratan yang diberikan, tanpa memperhatikan bagaimana sistem tersebut dibangun secara internal. Dalam pengujian *Black Box* menggunakan teknik analisis nilai batas, setiap fungsi yang akan diuji diberi nilai batas atas dan nilai batas bawah untuk melihat apakah *input* dan *outputnya* sesuai atau tidak (Permatasari dkk., 2023). Hasil dari pengujian ini memberikan keyakinan bahwa perangkat lunak mampu menangani berbagai skenario dengan benar, meningkatkan keandalan serta kepuasan pengguna.

2.2.8 *Confusion Matrix Testing*

Confusion Matrix adalah cara untuk memeriksa kinerja model klasifikasi. *Matrix* ini membandingkan prediksi model dengan label aktual untuk menghitung ukuran seperti presisi, perolehan kembali, dan akurasi. (Woro Isti Rahayu dkk., 2021). *Confusion Matrix* menampilkan hasil prediksi dalam bentuk tabel yang menunjukkan empat kemungkinan hasil: True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). *Confusion Matrix* memberikan gambaran tentang jumlah prediksi benar dan salah untuk setiap kelas, membantu dalam mengidentifikasi kesalahan dan memahami dimana model mengalami kebingungan dalam membuat

prediksi. *Confusion Matrix* berguna untuk memahami kinerja model dalam konteks masalah klasifikasi dengan dua kelas atau lebih.

Dalam kinerja *Confusion Matrix*, digunakan untuk mengukur hasil dari algoritma yang telah diterapkan. Beberapa pengukuran kinerja dalam *Confusion Matrix* meliputi *Accuracy*, *Precision*, *Recall*, dan *F1-score* atau F-Measure. Berikut adalah rumus-rumus untuk mengukur kinerja algoritma menggunakan *Confusion Matrix*.

1. *Accuracy*

Akurasi dihitung dengan menilai tingkat ketepatan prediksi yang benar terhadap keseluruhan data.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.7)$$

2. *Precision*

Precision diukur untuk menilai keakuratan antara data yang diharapkan dengan hasil prediksi dari algoritma yang digunakan. Berdasarkan pernyataan ini, *Precision* adalah rasio prediksi yang dihitung menggunakan persamaan dari keseluruhan hasil yang diprediksi sebagai positif.

$$Presisi = \frac{TP}{TP + FP} \quad (2.8)$$

3. *Recall*

Recall diukur untuk menilai seberapa baik algoritma berhasil menemukan kembali informasi yang benar. Berdasarkan pernyataan ini, *Recall* adalah rasio yang menggambarkan keseluruhan data yang benar-benar positif.

$$Recall = \frac{TP}{TP + FN} \quad (2.9)$$

4. *F1-Score*

Nilai F1-Score, atau F-Measure, dihitung dari kombinasi *Precision* dan *Recall* yang membandingkan kategori prediksi dengan kategori aktual.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (2.10)$$

Akurasi memiliki penilaian sebagai berikut.

1. Rentang nilai 0.90-1.00 dianggap sebagai klasifikasi yang sangat baik (excellent classification).
2. Rentang nilai 0.80-0.90 dianggap sebagai klasifikasi yang baik (good classification).
3. Rentang nilai 0.70-0.80 dianggap sebagai klasifikasi yang cukup (fair classification).
4. Rentang nilai 0.60-0.70 dianggap sebagai klasifikasi yang kurang baik (poor classification).
5. Rentang nilai 0.50-0.0 dianggap sebagai kegagalan (failure).