

BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Berdasarkan kajian literatur, deteksi penyakit berbasis citra digital telah banyak diterapkan pada berbagai kasus medis seperti deteksi penyakit kulit, hingga deteksi penyakit kuning pada bayi. Beberapa penelitian memanfaatkan pengolahan citra kulit untuk mendeteksi kelainan, dengan fokus pada pola dan warna kulit. Model berbasis *Machine Learning* dan *Deep Learning* sering digunakan untuk mengolah data citra dan memberikan akurasi yang cukup akurat, meskipun performa model sering dipengaruhi oleh keberagaman dataset.

Pendekatan berbasis *Convolutional Neural Network* (CNN) semakin banyak digunakan karena kemampuannya yang sangat baik untuk mengenali citra. Namun, penerapan CNN dalam mendeteksi penyakit kuning pada bayi belum banyak dilakukan. Penelitian ini mengembangkan model berbasis CNN dengan arsitektur DeepLabv3 untuk segmentasi dan arsitektur EffcientNetB3 untuk klasifikasi. Dengan memanfaatkan algoritma CNN, diharapkan hasil deteksi dapat lebih akurat dalam diagnosis dini penyakit kuning pada bayi. Berikut adalah *State of the art* penelitian terkait yang dapat dilihat pada tabel 2.1.

Tabel 2.1 Penelitian Terkait

No	Judul	Metode/Algoritma	Hasil Penelitian
1	<i>Comparative Analysis of Classification of Neonatal Bilirubin by Using Various Machine Learning Approaches</i> (Bhagat et al., 2024)	NuSVC, <i>Random Forest</i> , ADABOost, dan <i>Extra Trees</i> .	Algoritma NuSVC mencapai akurasi paling tinggi yaitu 62,50%, <i>precision</i> 61,90%, dan <i>recall</i> 56,52% dalam mengklasifikasi penyakit kuning.

No	Judul	Metode/Algoritma	Hasil Penelitian
2	<i>Detection of Jaundice in Neonates Using Artificial Intelligence (Nihila et al., 2021)</i>	SVM	Sistem deteksi <i>jaundice</i> non-invasif menggunakan algoritma SVM mencapai akurasi 81,1%. Menunjukkan kinerja yang baik dalam klasifikasi penyakit kuning.
3	<i>Deep Learning Network with Spatial Attention Module for Detecting Acute Bilirubin Encephalopathy in Newborns Based on Multimodal MRI (Zhang et al., 2023)</i>	MRI multimodal menggunakan arsitektur ResNet18	Studi ini menunjukkan bahwa model klasifikasi MRI multimodal menggunakan ResNet18 menjadikan klasifikasi terbaik dicapai dengan kombinasi gambar T1 dan T2, menghasilkan akurasi 80,80%.
4	<i>Classification of Lung Disease in X-Ray Images Using Gray Level Co Occurrence Matrix Method and Convolutional Neural Network (Nurcahyati et al., 2024)</i>	CNN dan GLCM	Hasilnya menunjukkan bahwa model CNN mencapai akurasi rata-rata 92%, sedangkan metode GLCM menghasilkan metrik yang lebih rendah 87%.
5	Pengembangan Model <i>Convolutional Neural Network (CNN)</i> untuk Klasifikasi Penyakit Kulit Berbasis Citra Digital (Imam Fathurrahman et al., 2025)	CNN	Studi ini dirancang untuk mengklasifikasikan penyakit kulit menggunakan CNN, dan mencapai akurasi sebesar 87,14%.
6	<i>Enhanced Skin Cancer Diagnosis Using Optimized CNN Architecture and Checkpoints for Automated Dermatological Lesion Classification (Musthafa et al., 2024)</i>	CNN	Studi ini mencapai akurasi yang luar biasa sebesar 97,78% untuk model CNN, precision 97,9%, recall 97,9%, dan <i>F1-score</i> 97,8%.
7	Deteksi Penyakit Kulit Wajah Menggunakan Tensorflow dengan Metode <i>Convolutional Neural Network</i> (Prastika et al., 2021)	CNN	Studi deteksi penyakit kulit wajah dengan jumlah 20 jenis penyakit diperoleh akurasi 99,91% dan persentase rata-rata ketepatan 80%.
8	<i>An automated detection of Scabies skindisease Using Image Processing and CNN (Halder et al., 2022)</i>	CNN	Studi deteksi penyakit <i>scabies</i> menggunakan CNN menunjukkan akurasi tinggi yaitu 97,25%.
9	<i>Human Skin Diseases Detection and</i>	CNN	Studi ini mencapai akurasi 89,45% dalam mendeteksi penyakit kanker kulit.

No	Judul	Metode/Algoritma	Hasil Penelitian
	<i>Classification using CNN</i> (Ahmed et al., 2023)		
10	Deteksi Otomatis Jerawat Wajah Menggunakan Metode <i>Convolutional Neural Network</i> (CNN) (Sudana Putra et al., 2021)	CNN	<i>Convolutional Neural Network</i> (CNN) mencapai kisaran akurasi 99,8% hingga 100%, menunjukkan kinerja yang sangat baik dalam mengklasifikasikan gambar jerawat.
11	<i>A Lightweight Complex-Valued DeepLabv3+ for Semantic Segmentation of PolSAR Image</i> (Yu et al., 2022)	DeepLabv3+ (CNN)	Segmentasi citra PolSAR, menggunakan arsitektur CNN DeepLabv3+ akurasi rata-rata yang didapatkan yaitu 93,89%.
12	<i>An improved semantic segmentation algorithm for high-resolution remote sensing images based on DeepLabv3+</i> (Wang et al., 2024)	DeepLabv3+ (CNN)	Segmentasi citra penginderaan jauh menggunakan arsitektur CNN DeepLabv3+ akurasi keseluruhan yang didapatkan yaitu 86,11%.
13	<i>Leveraging EfficientNet-B3 with Advanced Fine-Tuning for Precise Breast Cancer Classification</i> (George & S, 2024)	EfficientNetB3 (CNN)	Klasifikasi kanker payudara menggunakan arsitektur CNN EfficientNetB3 menghasilkan akurasi sebesar 95,25%.
14	<i>Deep Learning Approach for Enhancing Oral Squamous Cell Carcinoma with LIME Explainable AI Technique</i> (Islam et al., 2024)	EfficientNetB3 (CNN)	Penelitian ini membandingkan beberapa arsitektur untuk kalsifikasi (OSCC) Karsinoma sel skuamosa oral, yaitu EfficientNetB3, DenseNet121, ResNet101, dan VGG16. Akurasi tertinggi didapatkan oleh EfficientNetB3 dengan akurasi 99,64%.

2.2 Landasan Teori

2.2.1 Penyakit Kuning pada Bayi

Penyakit kuning atau bisa disebut juga *jaundice* atau *ikterus* merupakan keadaan kulit dan bagian putih mata menjadi berwarna kuning yang diakibatkan tingginya kadar bilirubin, kondisi ini umum terjadi pada bayi. Pada bayi, kadar bilirubin yang tinggi sering terjadi karena fungsi hati yang belum matang. Pada bayi

dengan fungsi hati yang belum optimal, bilirubin dapat menumpuk dalam darah sehingga menyebabkan warna kuning pada tubuh. Ikterus bisa muncul jika kadar bilirubin serum total $\geq 5\text{mg / dl}$ ($86\text{ }\mu\text{mol/L}$) (Rufaindah et al., 2022).

Penyakit kuning pada bayi dapat disebabkan oleh peningkatan kadar bilirubin, baik yang tak terkonjugasi maupun terkonjugasi. Pada bayi, ikterus dapat terjadi karena beberapa faktor, seperti prematuritas, pengaruh ASI, atau gangguan medis tertentu. Beberapa penyebab utama meliputi penyakit yang memicu kelebihan produksi bilirubin, seperti penyakit hemolitik atau defisiensi enzim, serta obstruksi saluran empedu yang mengganggu kemampuan hati untuk mengeluarkan bilirubin terkonjugasi. Gangguan dalam produksi dan ekskresi bilirubin juga dapat menjadi faktor penyebab, seperti defisiensi enzim *glucose-6-phosphate dehydrogenase* (G6PD), hipotiroidisme, galaktosemia, atau bayi dari ibu dengan diabetes (Hockenberry et al., 2015).

2.2.2 Citra Digital

Fungsi dua dimensi $f(x,y)$ adalah citra digital dengan ukuran M baris dan N kolom, dimana x dan y adalah posisi koordinat spasial dan amplitudo f di titik koordinat (x,y) , yang disebut intensitas atau tingkat keabuan citra pada titik tersebut (Gonzalez & Woods, 2008). Proses digitasi dapat diartikan sebagai proses konversi data analog ke format digital. Pengolahan citra digital adalah ilmu yang mempelajari cara memperbaiki kualitas gambar, seperti meningkatkan kontras, transformasi warna, restorasi citra, mengubah rotasi, translasi, skala, dan transformasi geometrik. Pengolahan citra juga mempelajari bagaimana memilih gambar dengan karakteristik fitur yang tepat untuk dianalisis. Secara keseluruhan

pengolahan citra berarti suatu cara mengolah suatu citra menjadi citra lain yang lebih sempurna atau yang diinginkan. Dengan kata lain, pengolahan citra merupakan proses dengan masukan citra dan menghasilkan keluaran berupa citra yang diinginkan (Sulistiyan et al., 2016).

Citra digital dipetakan menjadi elemen-elemen piksel berbentuk matriks dua dimensi. Setiap piksel memiliki nilai numerik yang mempresentasikan warna. Warna yang terlihat pada citra merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru atau biasa dikenal RGB. Kombinasi nilai dari ketiga komponen RGB tersebut menciptakan warna yang khas untuk setiap piksel. Pada citra dengan 256 warna, setiap piksel direpresentasikan dengan panjang 8 bit. Rentang RGB berkisar antara 0 hingga 255, yang didasarkan pada cara komputer mempresentasikan angka dalam 8 digit biner. Citra digital dapat dipresentasikan sebagai matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel, yaitu elemen terkecil dari sebuah citra. Setiap piksel memiliki dua parameter, yaitu koordinat dan nilai intensitas atau warna. Nilai pada koordinat (x,y) dilambangkan sebagai $f(x,y)$, yang menunjukkan tingkat intensitas atau warna piksel di lokasi tersebut (Kirana, 2021).

2.2.3 Machine Learning

Machine Learning atau pembelajaran mesin adalah bagian dari Kecerdasan Buatan, *Machine Learning* ini merupakan studi ilmiah tentang algoritma dan model statistik yang digunakan sistem komputer untuk melakukan tugas tertentu tanpa menggunakan instruksi eksplisit, dengan mengandalkan pola dan inferensi sebagai gantinya (Pratama, 2020). Istilah *Machine Learning* pertama kali didefinisikan oleh

Arthur Samuel pada tahun 1959. Menurut Arthur, Machine Learning merupakan suatu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrograman yang jelas (Wiederhold & McCarthy, 2010).

Menurut (Wira & Putra, 2020) Machine Learning adalah teknik untuk melakukan inferensi terhadap data dengan pendekatan matematis. Inti *Machine Learning* adalah membuat model (matematis) yang merefleksikan pola-pola data.

2.2.4 Deep Learning

Deep learning atau pembelajaran mendalam merupakan sub-bidang yang spesifik dari *machine learning* dengan penerapan algoritma *artificial neural network* (jaringan syaraf tiruan) untuk mempelajari representasi data yang tidak terstruktur dalam jumlah yang besar (Pramestya, 2018). Pada *deep learning* lapisan representasi dipelajari melalui model yang disebut *neural network* (jaringan syaraf) dalam lapisan literal yang ditumpuk satu sama lain secara terstruktur.

Arsitektur *deep learning* disebut sebagai *multi layer perceptron*, yang memiliki 4 node *neuron* pada *input layer* (lapisan masukan) berfungsi sebagai jalur untuk mentransmisikan sinyal masukan ke *node* berikutnya, 2 *hidden layer* (lapisan tersembunyi) yang merupakan perantara antara *input layer* dengan *output layer* dan berfungsi untuk menambah instruksi logika, serta 3 node *neuron* pada *output layer* (lapisan keluaran) yang menjadi hasil akhir dari *neural network*.

Jumlah *neuron* yang diterapkan pada sebuah lapisan tersembunyi sangat mempengaruhi performa *deep learning*. Jika jumlah *node neuron* yang diterapkan pada arsitektur *deep learning* terlalu sedikit maka dapat menyebabkan model

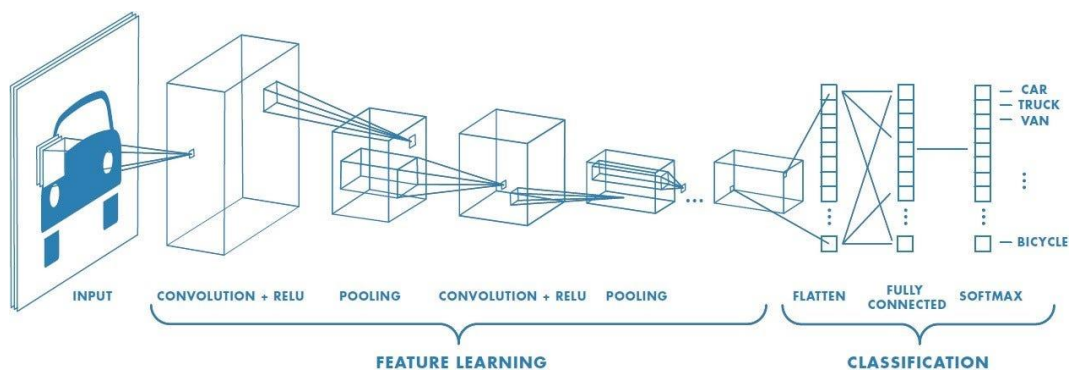
menjadi *underfitting*. Jika jumlah *node neuron* yang diterapkan terlalu banyak maka dapat menyebabkan model menjadi *overfitting*.

2.2.5 *Convolutional Neural Network (CNN)*

Convolution Neural Network bermula dari Yann LeCun dan teman temannya berhasil melakukan klasifikasi citra kode zip menggunakan kasus khusus dari Feed Forward Neural Network Pada tahun 1989. Mulai pada tahun 2012, dimana perangkat keras *Graphical Processing Unit* (GPU) berkembang, DNN juga ikut berkembang dan bahkan sebuah CNN dapat melakukan pengenalan citra dengan akurasi yang sangat baik dan dapat menyaingi manusia pada dataset tertentu (Sugiarto et al., 2018).

Convolutional Neural Network (CNN) 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 (Nurfita & Ariyanto, 2018). *Convolutional Neural Network (CNN)* adalah pengembangan dari *Multilayer Perceptron (MLP)* yang didesain 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 (Suartika et al., 2016). CNN hampir sama dengan neural network pada umumnya yang memiliki neuron yang memiliki bobot dan bias. CNN memiliki 1 tahap *training (Supervised Backpropagation)*.

Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa array yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi *layer* dan *pooling layer*. Berikut adalah jaringan arsitektur *Convolutional Neural Network* berdasarkan penelitian (Nurhikmat, 2018) dapat dilihat pada gambar 2.1.



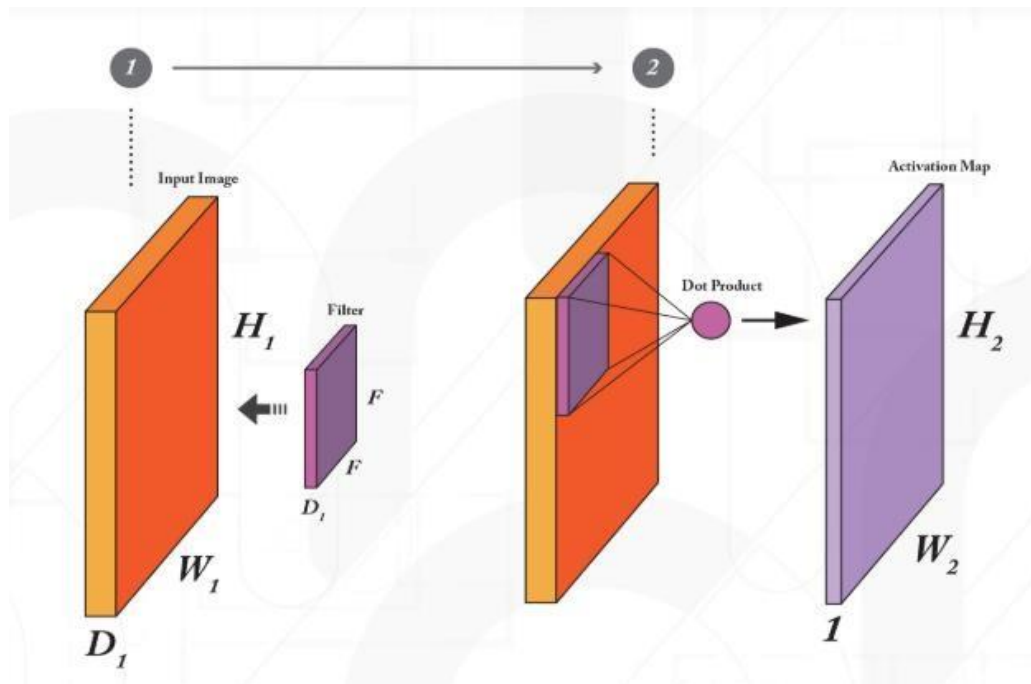
Gambar 2.1 Arsitektur *Convolutional Neural Network*

Pada gambar 2.1 Tahap pertama pada arsitektur CNN adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*). Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

1. *Convolution Layer*

Pada proses konvolusi, filter menjadi peran utama untuk proses pelatihan mesin. Seperti layaknya gambar, filter memiliki ukuran tinggi, lebar, dan tebal

tertentu. Filter ini diinisialisasi dengan nilai tertentu (*random* atau menggunakan teknik tertentu seperti Glorot), dan nilai dari filter inilah yang menjadi parameter yang akan di-*update* dalam proses *learning*. Proses konvolusi dapat dilihat pada gambar 2.2.



Gambar 2.2 Proses pada Convolutional Layer

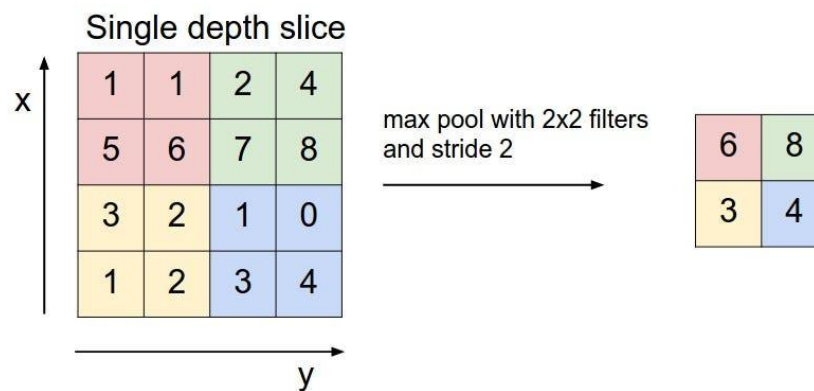
Gambar 2.2 menjelaskan input CNN selalu berbentuk kotak. Proses untuk gambar *non-rectangular* masih belum diketahui. Filter pun mengikuti karakteristik kotak tersebut. Parameter lapisan konvolusi terdiri dari satu set filter yang dapat dipelajari. Setiap filter kecil secara spasial (Panjang x lebar x tinggi), tetapi meluas melalui kedalaman dari volume input.

2. Operasi *Pooling*

Setelah melewati proses konvolusi kemudian hasilnya akan melalui tahap *pooling*. Fungsi dari *pooling* ini adalah untuk mereduksi input secara spasial (mengurangi jumlah parameter) dengan operasi *down-sampling*. Umumnya,

metode *pooling* yang digunakan adalah *max pooling* atau mengambil nilai terbesar dari bagian tersebut.

Berdasarkan penelitian (Suartika et al., 2016) *Max Pooling* adalah proses untuk meningkatkan invariansi posisi dari fitur menggunakan operasi *Max*. *Max Pooling* membagi *output* dari *Convolutional Layer* menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi seperti pada gambar 2.3 Grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran).



Gambar 2.3 Operasi Max Pooling

Secara umum, operasi pada pooling layer dimulai dengan menerima sebuah volume berukuran:

$$W1 \times H1 \times D1 \quad (1)$$

Membutuhkan dua hiperparameter yakni:

F = ukuran bidang *receptif*

S = stride

Menghasilkan sebuah volume dengan ukuran:

$$W2 \times H2 \times D2 \quad (2)$$

dimana:

$$W2 = (W1 - F)/S + 1 \quad (3)$$

$$H2 = (H1 - F)/S + 1 \quad (4)$$

$$D2 = D1 \quad (5)$$

Memperkenalkan *zero* parameter karena menghitung fungsi input yang tetap, dan *Pooling layer* biasanya tidak memasukkan input menggunakan *zero-padding*. Pada praktiknya, hanya ada dua variasi yang terlihat pada *max pooling layer* yaitu sebuah *pooling layer* dengan $F=3$, $S=2$ (disebut juga sebagai *overlapping pooling*) dan lebih umum lagi $F=2$, $S=2$. Ukuran *pooling* dengan bidang reseptif yang lebih besar sangat merusak.

3. Fully-Connected Layer

Feature map yang dihasilkan dari *feature extraction layer* masih berbentuk *multidimensional array*, sehingga kita harus melakukan “*flatten*” atau *reshape* *feature map* menjadi sebuah vector agar bisa kita gunakan sebagai *input* dari *fully-connected layer*. Neuron pada *Fully Connected Layer* memiliki hubungan yang lengkap pada semua aktivasi dalam layer sebelumnya. Aktivasi tersebut kemudian di komputasi dengan sebuah perkalian matriks diikuti oleh bias *offset*. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

Activation map yang dihasilkan dari *feature extraction layer* masih berbentuk *multidimensional array*, sehingga mau tidak mau harus dilakukan

reshape activation map menjadi suatu vektor agar bisa digunakan sebagai input dari *fully-connected layer*. Layer ini memiliki *hidden layer*, *activation function*, *output layer*, dan *loss function*. Layer ini adalah layer yang biasanya digunakan dalam penerapan *multi layer perceptron* dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

Setiap neuron pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully-connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, sedangkan *fully-connected layer* hanya dapat diimplementasikan di akhir jaringan.

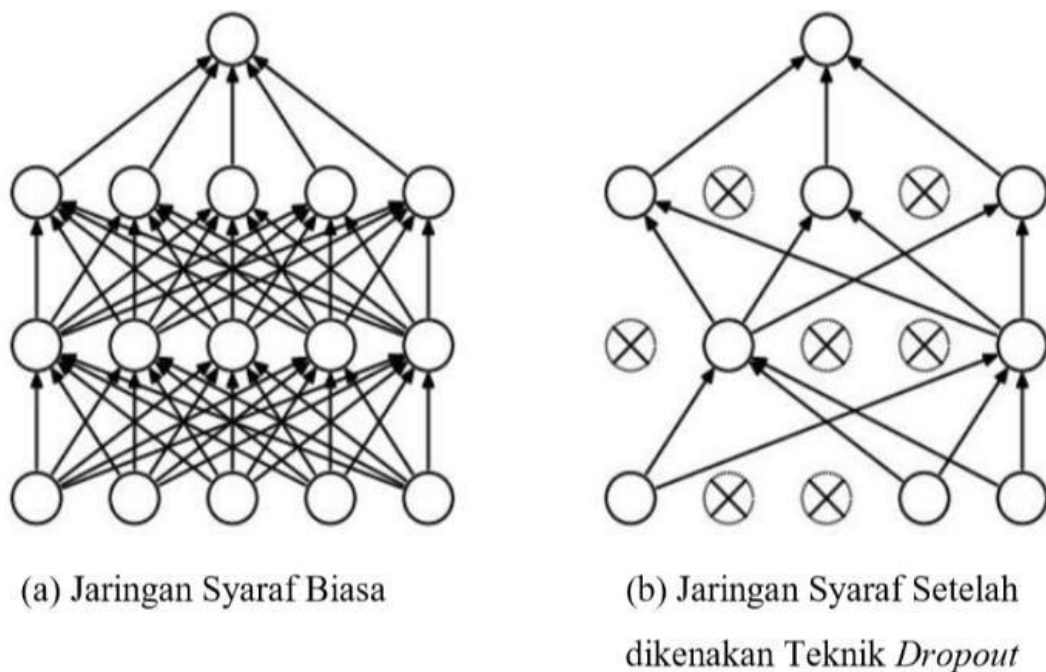
Convolution layer dengan ukuran kernel 1 x 1 melakukan fungsi yang sama dengan *fully-connected layer* namun dengan tetap mempertahankan karakter spasial dari data. Sehingga hal tersebut membuat penggunaan layer ini pada CNN sekarang tidak banyak dipakai.

Selain arsitektur yang telah dipaparkan, masih banyak arsitektur lain yang dapat digunakan untuk berbagai karakteristik data maupun arsitektur yang cocok untuk karakteristik/permasalahan tertentu.

4. *Dropout Regularization*

Dropout adalah teknik regularisasi jaringan syaraf dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. Neuron-neuron ini dapat dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada neuron pada saat melakukan *backpropagation*.

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* mengacu kepada menghilangkan *neuron* yang berupa *hidden* maupun layer yang *visible* di dalam jaringan. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak. Setiap neuron akan diberikan probabilitas yang bernilai antara 0 dan 1. *Dropout* dapat dilihat pada gambar 2.4.



Gambar 2.4 *Dropout*

Pada gambar 2.4 jaringan syaraf (a) merupakan jaringan syaraf biasa dengan 2 lapisan tersembunyi. Sedangkan pada bagian (b) jaringan syaraf sudah diaplikasikan teknik regularisasi *dropout* dimana ada beberapa neuron aktivasi yang tidak dipakai lagi. Teknik ini sangat mudah diimplementasikan pada model CNN dan akan berdampak pada performa model dalam melatih serta mengurangi *overfitting*.

5. *Softmax Layer*

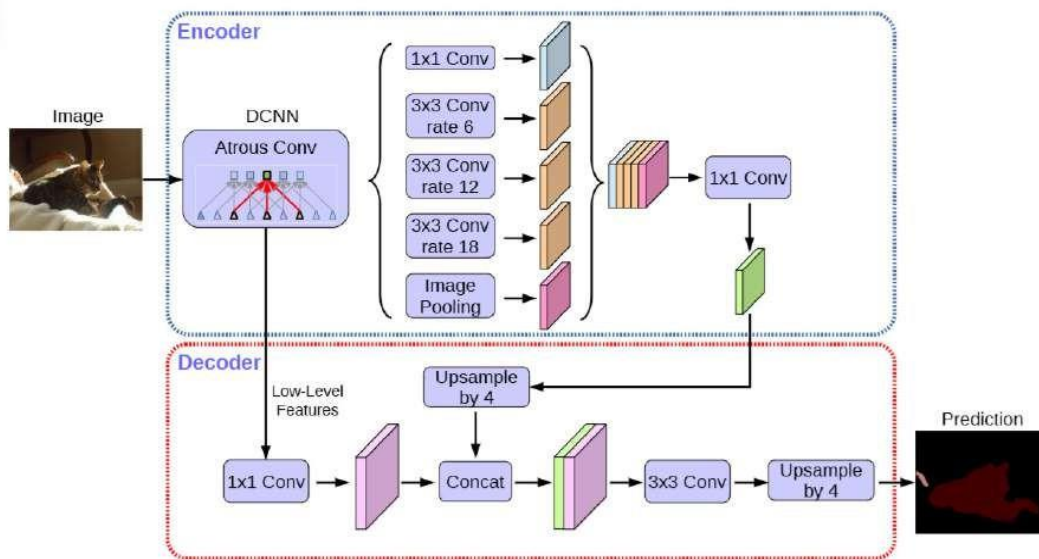
Fungsi *softmax* digunakan untuk metode klasifikasi dengan jumlah kelas yang banyak, seperti regresi logistik multinomial, analisis diskriminan *linear multiclass*, *Naive Bayes Classifier*, dan *Artificial Neural Network (ANN)*. *Softmax* adalah sebuah fungsi yang mengubah K-dimensi vektor 'x' yang berupa nilai sebenarnya menjadi vektor dengan bentuk yang sama namun dengan nilai dalam rentang 0-1, yang jumlahnya 1. Fungsi *softmax* digunakan dalam *layer* yang terdapat pada *neural network* dan biasanya terdapat pada layer terakhir untuk mendapatkan output. Tidak jauh berbeda dengan neuron pada umumnya *softmax neuron* menerima input lalu melakukan pembobotan dan penambahan bias. Tetapi setelah itu neuron pada *softmax layer* tidak menerapkan fungsi aktivasi melainkan menggunakan fungsi *softmax*. Dapat disimpulkan bahwa *softmax layer*, menentukan probabilitas terbesar untuk hasil kelasnya.

2.2.6 DeepLabv3+

DeepLabv3+ adalah model segmentasi gambar semantik berbasis *convolutional neural network (CNN)* yang dirancang untuk meningkatkan akurasi segmentasi, khususnya pada batas-batas objek. Model ini memanfaatkan *Atrous Spatial Pyramid Pooling (ASPP)* untuk menangkap informasi kontekstual dari berbagai skala menggunakan *atrous convolution*, yang memungkinkan model mempertahankan resolusi tinggi tanpa kehilangan detail spasial akibat *pooling* (Chen et al., 2018). Selain itu, penggunaan *depthwise separable convolution* mengurangi kompleksitas komputasi dan jumlah parameter, sehingga model tetap efisien pada berbagai skala. *Encoder* menghasilkan *feature map* dengan resolusi

rendah tetapi kaya informasi semantik, sementara *decoder* memulihkan detail objek dengan menggabungkan fitur tingkat rendah dari backbone seperti ResNet-101.

Sebagai pengembangan dari CNN, DeepLabv3+ memaksimalkan kemampuan CNN untuk menangkap pola visual dengan meningkatkan resolusi fitur melalui *atrous convolution*. Pendekatan ini memungkinkan model menangkap informasi pada berbagai tingkat hierarki tanpa peningkatan signifikan dalam beban komputasi. Dengan mekanisme *decoder* yang dirancang khusus, model ini menghasilkan segmentasi yang lebih detail, terutama di area batas objek (Chen et al., 2018). DeepLabv3+ telah diuji pada beberapa dataset benchmark seperti PASCAL VOC 2012 dengan akurasi 89.0% dan Cityscapes dengan akurasi 82.1%. Modifikasi pada arsitektur CNN seperti penggantian *max pooling* dengan *depthwise separable convolution* turut berkontribusi pada peningkatan kinerja, menjadikannya salah satu solusi unggul dalam segmentasi semantik berbasis CNN. Gambar 2.5. menampilkan arsitektur model DeepLabv3+ dan juga proses untuk menghasilkan label prediksi.



Gambar 2.5 Arsitektur Model DeepLabv3+

Gambar 2.5 menunjukkan arsitektur DeepLabv3+, yang merupakan pengembangan dari DeepLabv3 dengan menambahkan modul *decoder* untuk meningkatkan segmentasi, terutama pada batas objek. *Encoder* pada DeepLabv3+ menggunakan *Atrous Spatial Pyramid Pooling* (ASPP) dengan *atrous convolution* pada tingkat dilasi berbeda (*rate* 6, 12, dan 18) untuk menangkap informasi multi-skala, ditambah *pooling* gambar untuk menyediakan informasi global. *Low-level feature* dari *backbone*, seperti lapisan awal ResNet sebelum *downsampling*, juga dimanfaatkan untuk mempertahankan detail spasial.

Modul *decoder* menerima keluaran dari ASPP dan *low-level feature* yang telah melalui *convolution* 1x1 untuk mengurangi dimensi saluran. Setelah penggabungan (*concatenate*), fitur tersebut diproses melalui *convolution* 3x3 untuk meningkatkan kualitas segmentasi, diikuti dengan upsampling bilinear untuk mengembalikan resolusi ke ukuran asli. Pendekatan ini memungkinkan DeepLabv3+ menghasilkan segmentasi semantik yang lebih baik, terutama pada

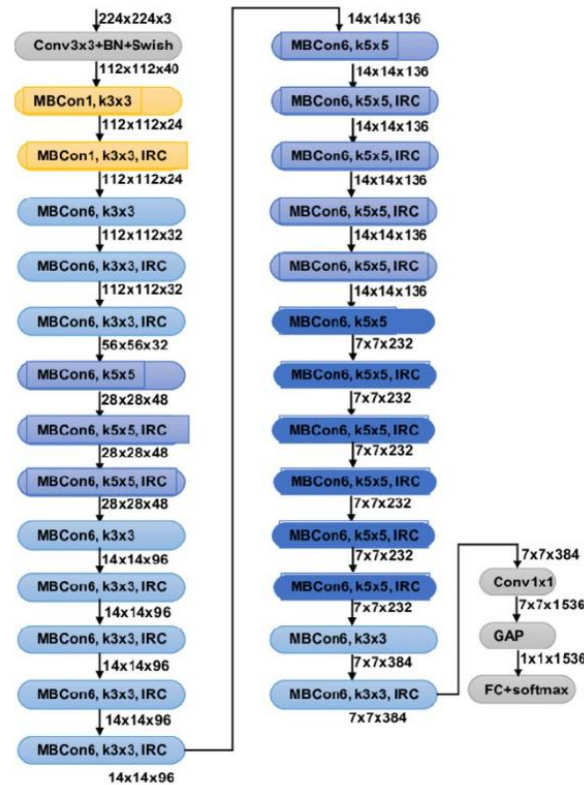
batas objek, dengan memadukan informasi multi-skala dari ASPP dan kemampuan pemulihan detail dari struktur *encoder-decoder*.

2.2.7 EfficientNetB3

EfficientNet-B3 adalah model *convolutional neural network* (CNN) yang merupakan bagian dari keluarga EfficientNet, yang terkenal karena pendekatan inovatifnya dalam mencapai efisiensi tinggi dengan kinerja yang kompetitif. Inti dari EfficientNet-B3 adalah metode penskalaan gabungan (*compound scaling*), yang secara simultan menyesuaikan lebar (jumlah filter), kedalaman (jumlah lapisan), dan resolusi input gambar menggunakan koefisien terpadu (Ba Alawi & Bozkurt, 2023). Pendekatan ini berbeda dari metode penskalaan tradisional yang biasanya hanya fokus pada satu dimensi jaringan, memungkinkan EfficientNet-B3 untuk mengalokasikan sumber daya komputasi secara lebih efektif dan mencapai keseimbangan yang lebih baik antara akurasi dan efisiensi.

Model EfficientNet-B3 dibangun di atas arsitektur dasar yang menggunakan blok *Mobile Inverted Bottleneck Convolution* (MBConv). Blok ini dikenal karena efisiensinya dalam melakukan operasi konvolusi dengan jumlah parameter dan komputasi yang lebih sedikit dibandingkan dengan blok konvolusi standar. Selain itu, beberapa blok MBConv dalam EfficientNet-B3 dilengkapi dengan modul *Squeeze-and-Excitation* (SE), yang memungkinkan jaringan untuk mempelajari kepentingan relatif dari berbagai saluran fitur, sehingga meningkatkan kemampuan representasi model (Ba Alawi & Bozkurt, 2023). Kombinasi dari blok MBConv dan pendekatan penskalaan gabungan memungkinkan EfficientNet-B3 untuk mencapai

kinerja yang baik dengan sumber daya yang lebih terbatas. Arsitektur EfficientNet-B3 dapat dilihat pada gambar 2.6.



Gambar 2.6 Arsitektur EfficientNetB3

Gambar 2.6 merupakan arsitektur EfficientNet-B3, alur pemrosesan dimulai dengan input gambar 224x224 RGB yang melewati lapisan konvolusi awal (Conv3x3) diikuti oleh *Batch Normalization* dan aktivasi *Swish*. Ini menghasilkan fitur peta awal dengan dimensi 112x112 dan 40 saluran. Selanjutnya, jaringan terdiri dari serangkaian tahapan yang berisi tumpukan blok MBConv dengan konfigurasi yang bervariasi dalam ukuran kernel (3x3 atau 5x5) dan faktor ekspansi. Perhatikan bagaimana resolusi spasial fitur peta secara bertahap berkurang sementara jumlah saluran meningkat seiring dengan kedalaman jaringan, mencerminkan hierarki fitur yang dipelajari oleh model. Keberadaan *Inverted*

Residual Connection (IRC) dalam beberapa blok MBConv juga terlihat, yang berperan penting dalam memfasilitasi pelatihan jaringan yang lebih dalam dan mencegah masalah vanishing gradient.

Di bagian akhir arsitektur, setelah melalui serangkaian blok MBConv, fitur peta dengan resolusi 7x7 dan 384 saluran diproses oleh lapisan konvolusi 1x1 (Conv1x1) untuk memperluas jumlah saluran menjadi 1536. Operasi *Global Average Pooling* (GAP) kemudian mereduksi dimensi spasial menjadi vektor 1x1536. Akhirnya, vektor fitur ini diumpankan ke lapisan *Fully Connected* (FC) yang diikuti oleh fungsi aktivasi *softmax* untuk menghasilkan probabilitas kelas. Konfigurasi arsitektur yang spesifik ini, dengan variasi dalam jenis blok MBConv dan jumlah filternya di setiap tahapan, adalah hasil dari proses pencarian arsitektur yang dirancang untuk mengoptimalkan kinerja dan efisiensi EfficientNet-B3.

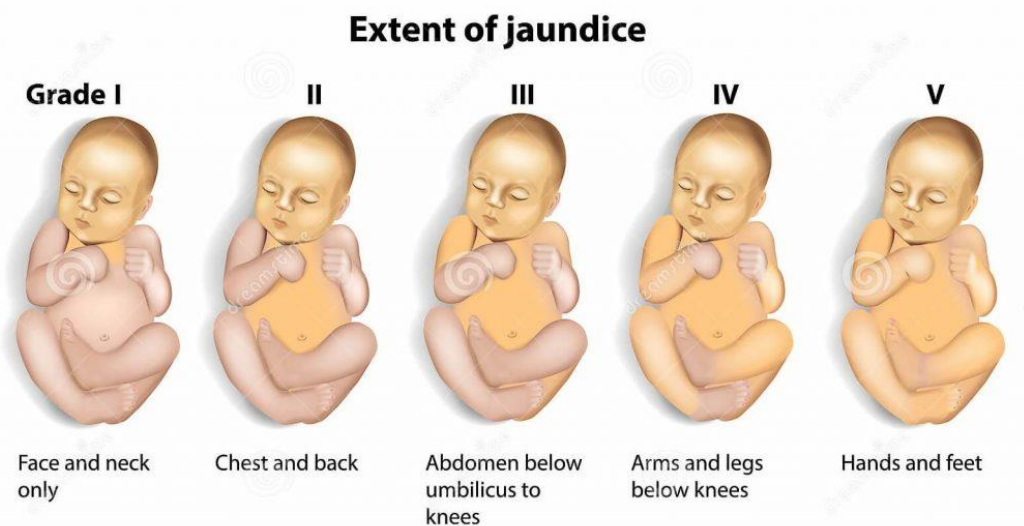
2.2.8 Roboflow

Roboflow adalah *platform* berbasis web yang mendukung pengembangan aplikasi *computer vision*, khususnya dalam pengelolaan dataset untuk deteksi objek. *Platform* ini menyediakan fitur anotasi objek menggunakan *bounding box*, pra pemrosesan data seperti konversi *grayscale*, serta augmentasi untuk memperkaya variasi dataset. Roboflow mempermudah proses pengelolaan dan berbagi dataset secara kolaboratif, yang merupakan langkah penting dalam membangun model berbasis *deep learning*. Selain itu, platform ini membantu mengatasi tantangan dalam tahap akuisisi data, memastikan dataset yang digunakan berkualitas tinggi dan relevan. Dengan integrasi *pipeline* untuk anotasi, augmentasi, dan pelatihan

model, Roboflow menjadi alat yang efektif dalam mendukung penelitian dan pengembangan sistem deteksi objek (Hayati et al., 2023).

2.2.9 Metode Kramer

Metode Kramer adalah pendekatan yang digunakan untuk menilai derajat penyakit kuning pada bayi dengan mengamati perubahan warna kulit akibat peningkatan kadar bilirubin dalam darah. Dikembangkan oleh Dr. H. Kramer pada tahun 1960-an, metode ini membagi derajat ikterus menjadi beberapa kategori berdasarkan lokasi dan intensitas warna kuning yang terlihat pada kulit. Penilaian visual ini didasarkan pada prinsip bahwa peningkatan kadar bilirubin akan menyebabkan perubahan warna yang dapat diamati secara klinis (Kramer, 1969). Metode Kramer membagi tubuh bayi menjadi 5 bagian yang dapat dilihat pada gambar 2.7.



Gambar 2.7 Tingkat Penyakit Kuning berdasarkan Metode Kramer

Gambar 2.7 merupakan tingkat penyakit kuning berdasarkan metode Kramer, Zona 1 mencakup kepala dan leher (bilirubin <5 mg/dL), Zona 2 meliputi

batang tubuh hingga pusar (bilirubin 5–10 mg/dL), Zona 3 mencakup area tubuh dari pusar hingga lutut (bilirubin 10-12 mg/dL), Zona 4 meliputi lengan dan tungkai bawah (bilirubin 12-15 mg/dL), dan Zona 5 mencakup telapak tangan dan kaki (bilirubin >15 mg/dL). Perubahan warna kuning yang muncul di zona tubuh tertentu memberikan gambaran awal tingkat keparahan penyakit.

Penyakit kuning berdasarkan metode kramer dibagi menjadi 5 *grade*, dimana setiap *gradenya* mempunyai rekomendasi penanganan berbeda, apabila bayi ada di *grade* 1 rekomendasi penanganannya yaitu pemberian ASI yang kuat, *grade* 2 rekomendasi penanganannya yaitu jemur dibawah sinar matahari, *grade* 3 bayi memerlukan fototerapi, apabila ada di *grade* 4 bayi memerlukan fototerapi yang intensif, dan apabila ada di *grade* 5 maka bayi harus menjalani transfusi tukar (Ramadhanty & Puspitasari, 2022).