Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Perpustakaan Univeristas Multimedia Nusantara

BAB 2
LANDASAN TEORI

Survei literatur mengenai diagnosis penyakit COVID-19 dengan menggunakan gambar CT-Scan / X-Ray paru-paru menunjukkan bahwa sejumlah besar studi telah dilakukan oleh berbagai penulis untuk menyelesaikan permasalahan ini. Permasalahan dapat diselesaikan dengan berbagai macam metode dan beragam arsitektur yang diterapkan.

2.1 COVID-19

Pada akhir Desember 2019, wabah pneumonia baru yang ditandai dengan demam, batuk kering, dan kelelahan muncul di pasar basah grosir makanan laut, Pasar Grosir Makanan Laut Huanan, Wuhan, Hubei, Cina pada Desember 2019 dan menginfeksi sekitar 66% staf di sana. Pasar ditutup pada 1 Januari 2020 setelah pengumuman peringatan epidemiologis oleh otoritas kesehatan setempat pada 31 Desember 2019. Namun, pada bulan berikutnya (Januari), ribuan orang di China, termasuk banyak provinsi dan kota-kota diserang oleh penyebaran penyakit yang merajalela [1].

Penyakit ini disebabkan oleh salah satu mutasi dari virus SARS-CoV sehingga dinamakan SARS-CoV-2. Enam asam amino *Receptor-Binding Domain* (RBD) yang menempel pada reseptor cenderung memiliki ikatan yang kuat terhadap *Angiotensin Converting Enzyme* 2 (ACE2) [9]. ACE2 merupakan enzim yang menempel pada permukaan luar di beberapa sel organ manusia seperti sel paru-paru, arteri, jantung, ginjal, dan usus. Berdasarkan penelitian terkait [10], SARS-CoV memiliki afinitas ikatan yang kuat sehingga ikatan inilah yang mempermudah dalam virus SARS-CoV untuk berpindah dari satu inang ke inang lainnya (dalam hal ini manusia). Yang menjadi pembeda pada mutasi SARS-CoV-2 adalah virus SARS-CoV-2 dapat mengenali reseptor ACE2 manusia secara lebih efisien dari pada SARS-CoV sehingga kemampuan untuk menular antar manusia pun semakin tinggi [11].

Meluasnya penyebaran virus SARS-CoV-2 ke seluruh dunia menyebabkan banyak orang harus dikarantina, yakni dipisahkan dari masyarakat lainnya untuk mencegah penularan antar manusia. Belum lagi dampak yang berimbas ke perekonomian, yakni lumpuhnya industri sehingga banyak pegawai yang

dirumahkan dan berbagai sektor esensial pun tersendat operasionalnya. Untuk mengatasi meluasnya penyebaran virus SARS-CoV-2, dilakukan beberapa strategi seperti deteksi orang yang terpapar virus ini. Deteksi menjadi penting untuk mengendalikan penyebaran dan melakukan tindakan pencegahan penyebarannya [2].

Kurangnya alat deteksi virus memperlambat proses deteksi orang yang terpapar virus ini. Keterlambatan dalam diagnosis menyebabkan seseorang lambat dalam mendapatkan penanganan medis lanjut terutama orang yang mengalami komplikasi atau komorbid. Ini pun meningkatkan jumlah pasien dan korban jiwa. Tingkat penyebaran akan menurun apabila orang yang terpapar diketahui lebih cepat dan dilakukan karantina.

Diagnosis COVID-19 pada umumnya dilakukan dengan tes *Reverse-Transcription Polymerase Chain Reaction* (RT-PCR) dengan mengambil spesimen DNA pada saluran pernapasan. Tes ini hanya dapat dilakukan pada laboratorium yang dilengkapi infrastruktur yang lengkap. Selain itu, dalam beberapa kasus tes COVID-19 perlu diulang sehingga biaya yang diperlukan tidak murah dan waktu yang dibutuhkan untuk mengetahui hasil diagnosis pun lama yakni satu atau dua hari.

Menurut WHO [3], virus SARS-CoV-2 atau disebut COVID-19 ini memiliki gejala utama yakni gangguan pernapasan. Gangguan pernapasan artinya virus telah menginfeksi paru-paru sehingga gambar X-Ray akan mampu memberikan informasi terkait adanya virus tersebut atau tidak [4]. Ketersediaan perangkat pemindai X-Ray di rumah sakit dan laboratorium mampu digunakan sebaik mungkin untuk mendeteksi penyakit ini. Namun, metode pemeriksaan biasanya dilakukan manual oleh dokter tanpa melibatkan penggunaan teknologi.

Dengan adanya perkembangan teknologi dan didukung dengan dataset yang besar, mengembangkan sebuah model dengan memanfaatkan *pre-trained* model yang mampu memberikan hasil yang akurat akan sangat menjanjikan karena dapat mempercepat waktu diagnosis yang biasanya dilakukan oleh dokter secara manual.

2.2 Convolutional Neural Network

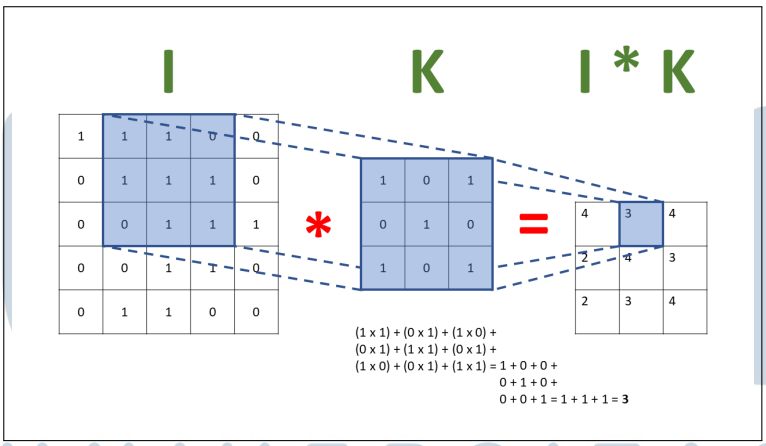
Convolutional Neural Network atau CNN merupakan metode yang cukup populer dalam *deep neural network*. Hasil yang diperoleh dari CNN selalu menjadi yang terbaik selama beberapa dekade terakhir dalam berbagai bidang yang berkaitan dengan pengenalan pola, mulai dari pemrosesan gambar hingga pendeteksi suara. Pada pemrosesan gambar, penggunaan CNN sendiri dinilai memberikan

manfaat dikarenakan fitur dalam gambar tidak perlu tergantung secara *spasial*. Contohnya pada tugas pengenalan wajah. Hal yang perlu diperhatikan adalah bagaimana mesin dapat mendeteksi adanya wajah dalam gambar tanpa memedulikan lokasi wajah pada gambar. Aspek lain dari CNN yakni mendapatkan fitur abstrak dari tiap lapisan saat input disebar ke lapisan yang dalam. Contohnya pada klasifikasi wajah, dimana pada lapisan pertama mendeteksi ujung gambar. Kemudian, pada lapisan berikutnya mendeteksi bentuk sederhana hingga menemukan fitur tingkat tinggi seperti wajah [12].

Berdasarkan Bartosz Szabłowski [13], cara kerja CNN adalah:

1. Convolutional Layer

   Pada *layer* ini, *pixel* dari gambar input akan dipadankan dengan sebuah filter yang disebut *kernel*. *Kernel* sendiri adalah matriks yang memiliki tinggi, lebar, dan panjang yang ditujukan untuk mendeteksi fitur/objek yang ingin dicari dalam gambar. *Pixel* input akan dikalikan dengan *kernel* dari kiri atas hingga kanan bawah. Contoh: Input gambar sebesar 5x5x3 dengan *kernel* sebesar 3x3x3 dan *stride* atau pergeseran sebesar 1 *pixel*. Simulasinya dapat dilihat pada gambar 2.1.



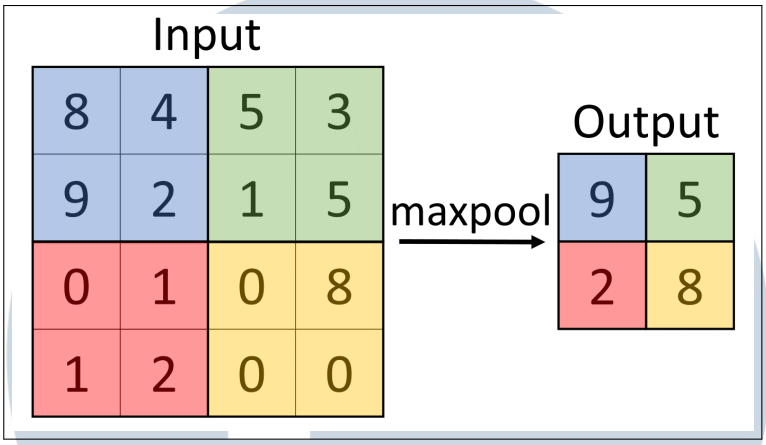
Gambar 2.1. Simulasi cara kerja pada *Convolutional Layer*

Sumber: [13]

2. Pooling Layer

   Di *layer* ini, terjadi proses pengurangan input. Metode *pooling* yang paling sering digunakan yakni *max-pooling* dimana nilai terbesar akan mewakili areanya. Contohnya adalah input 4x4 berikut yang dilakukan *max-pooling* 2x2 dengan *stride* sebesar 2 pada Gambar 2.2. Terlihat bahwa di tiap area warna

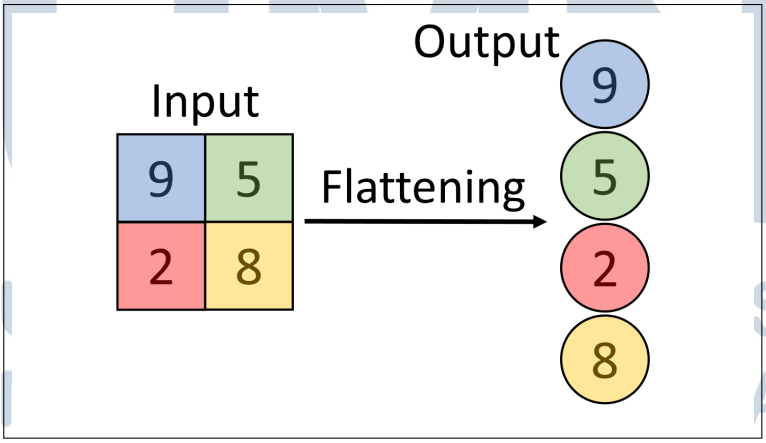
input akan diambil nilai paling maksimal. Dengan demikian, matriks input pun akan berkurang.



Gambar 2.2. Simulasi cara kerja *max-pooling*
Sumber: [13]

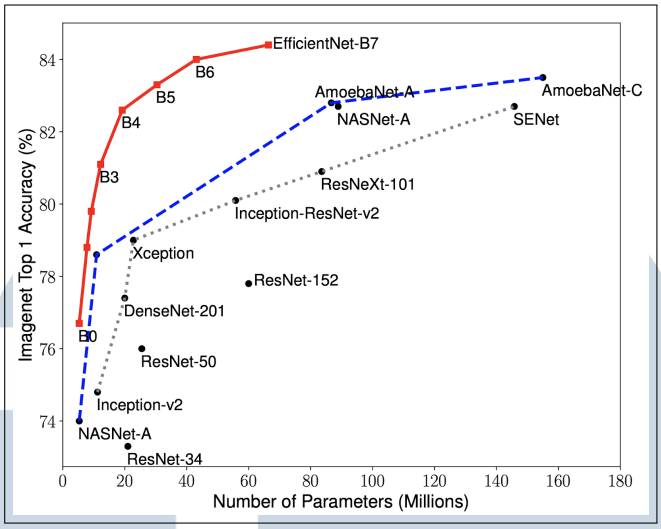
3. Fully Connected Layer

Input data dari *fully connected layer* merupakan hasil dari *layer* sebelumnya. Pada *layer* ini, matriks akan diratakan menjadi berupa vektor seperti pada Gambar 2.3. *Layer* ini menjadi penentu dari hasil CNN itu sendiri dimana pada tugas klasifikasi akan memberikan hasil berupa kelas dari gambar tersebut.



Gambar 2.3. Simulasi cara kerja pada *Fully Connected Layer*
Sumber: [13]

CNN sendiri memiliki banyak arsitektur mulai dari ResNet hingga EfficientNet. Perbandingan antar arsitektur CNN dapat dilihat pada Gambar 2.4.



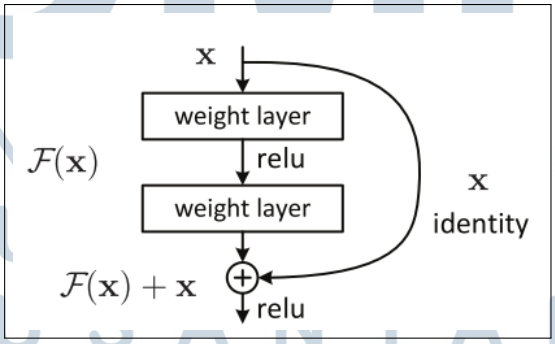
Gambar 2.4. Perbandingan jumlah parameter yang digunakan dengan akurasi yang dihasilkan berdasarkan arsitektur CNN

Sumber: [7]

2.3 ResNet

ResNet atau kepanjangannya *Residual Network* merupakan arsitektur CNN yang dirancang oleh Kaiming He yang mana menjadi juara satu dalam ILSVRC 2015 dalam lomba kompetisi klasifikasi ImageNet yang merupakan dataset terbesar dalam klasifikasi gambar dengan menggunakan *layer* 8 kali lebih dalam dibanding dengan VGG-16 [14].

ResNet terdiri dari tumpukan "*Residual Unit*". ResNet sendiri memperkenalkan fitur yang mampu mengabaikan satu atau lebih *layer* seperti pada Gambar 2.5.



Gambar 2.5. Fitur ResNet

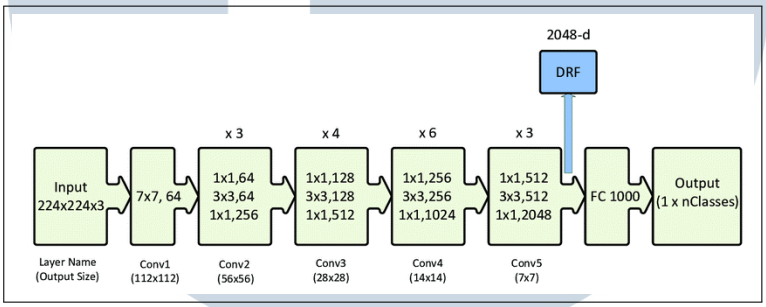
Sumber: [5]

Terlihat bahwa arsitektur ini memiliki koneksi langsung yang mengabaikan

beberapa layer. Ini dinamakan "*skip connections*" dan merupakan bagian utama dari *residual block*. Metode ini menyelesaikan masalah dimana jika *network* yang dibangun semakin dalam, akurasi pun stagnan atau tidak berkembang.

Arsitektur ini digunakan peneliti [8] untuk diagnosis infeksi COVID-19. Pada jurnal tersebut, metode yang digunakan adalah *supervised learning* dimana data latih yang diberikan kepada mesin diberi label. Model ini dilatih sebanyak 3 tahap dimana pada masing-masing tahap, mesin akan dilatih dengan dataset yang berbeda sehingga memberikan tingkat akurasi yang semakin tinggi dan mampu mendeteksi *noise* pada gambar dan menghiraukannya [15].

Arsitektur ResNet 50 dapat dilihat pada Gambar 2.6.



Gambar 2.6. Arsitektur ResNet 50
Sumber: [4]

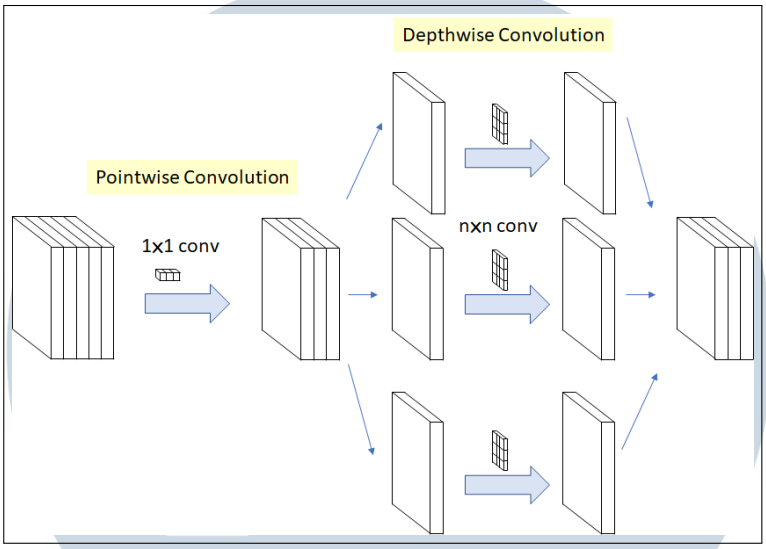
ResNet sendiri terus berkembang. Setahun setelah perilisan jurnal ResNet [14], dikeluarkan jurnal baru dengan penulis yang sama yang mengeluarkan versi kedua dari ResNet itu sendiri yakni ResNet50V2 dan ResNet101V2 yang tentunya memberikan hasil yang lebih baik dibanding pendahulunya [16].

ResNet versi kedua ini menggunakan *pre-activation* dari *residual block* yang telah dibuat pada versi pertamanya. *Pre-activation* ini merupakan *layer* yang ditambahkan sebelum dimasukkan ke textitresidual block. Penambahan fitur ini pun menghasilkan hasil yang menjanjikan dibanding dengan versi pertamanya.

2.4 Xception

Xception yang kepanjangannya Extreme version of Inception merupakan arsitektur yang dikembangkan dari Inception oleh Francois Chollet yang merupakan pegawai Google. Xception dikembangkan dengan memodifikasi *depthwise separable convolution*. Xception diklaim mengungguli arsitektur Inception V3 yang menjadi pendahulunya dalam dataset ImageNet. Secara singkat, jumlah parameter

yang digunakan sama dengan Inception V3 hanya saja yang dicapai adalah efisiensi dari penggunaan parameter model [6].

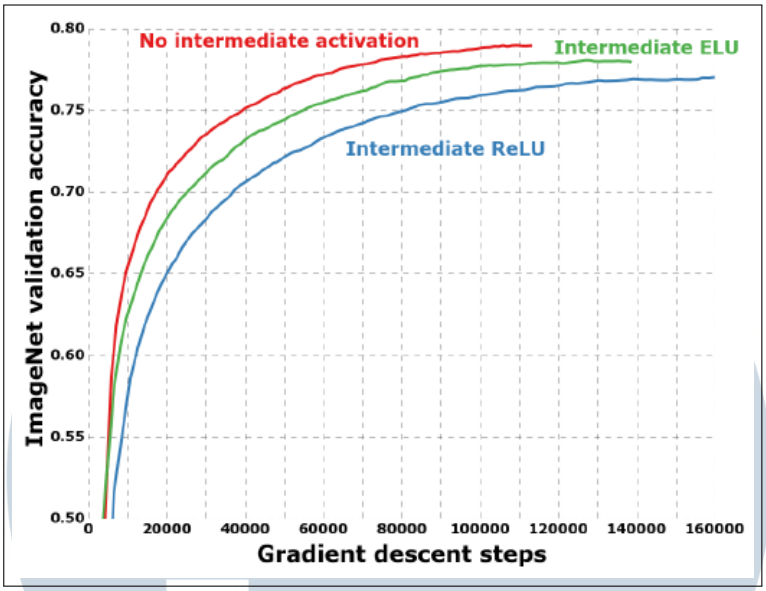


Gambar 2.7. Modifikasi *depthwise separable convolution* pada Xception
Sumber: [17]

Implementasi modifikasi *depthwise separable convolution* dalam arsitektur Xception dapat dilihat pada Gambar 2.7. Terlihat bahwa *pointwise convolution* (1 x 1 *Convolution*) dilakukan terlebih dahulu sebelum *depthwise convolution* (n x n *Spatial Convolution*). Ini merupakan kebalikan dari model Inception V3 yang mana *depthwise convolution* dilakukan dahulu kemudian *pointwise convolution*. Dalam gambar, n berjumlah 3 dikarenakan mengikuti model Inception V3.

Perbedaan lainnya antara Xception dan Inception adalah Xception tidak menggunakan *intermediate ReLU non-linearity* sehingga memberikan akurasi yang tinggi dibanding yang menggunakan ELU atau ReLU [17]. Ini terbukti meningkatkan akurasi dari model yang dihasilkan seperti pada Gambar 2.8.

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 2.8. Perbandingan hasil akurasi pada Xception
Sumber: [17]

2.5 EfficientNet

EfficientNet adalah arsitektur CNN yang dipublikasikan oleh Google pada salah satu jurnalnya di tahun 2019. Model ini tidak hanya menaikkan akurasi, tapi juga meningkatkan ketepatan model dengan mengurangi parameter dan *Floating Point Operations Per Second* (FLOPS) seperti yang dilakukan pada arsitektur GPipe yang mana GPipe sendiri menggunakan *Pipeline Parallelism* [7].

EfficientNet menskalakan dari aspek lebar (*width*), kedalaman (*depth*), dan resolusi jaringan (*resolution*) secara seragam dengan koefisien tetap yang didefinisikan sebagai simbol ϕ. Metode ini disebut *Compound Scaling* [18]. Rumusnya dapat dituliskan sebagai berikut:

$$depth : \delta = \alpha^{\phi} \tag{2.1}$$

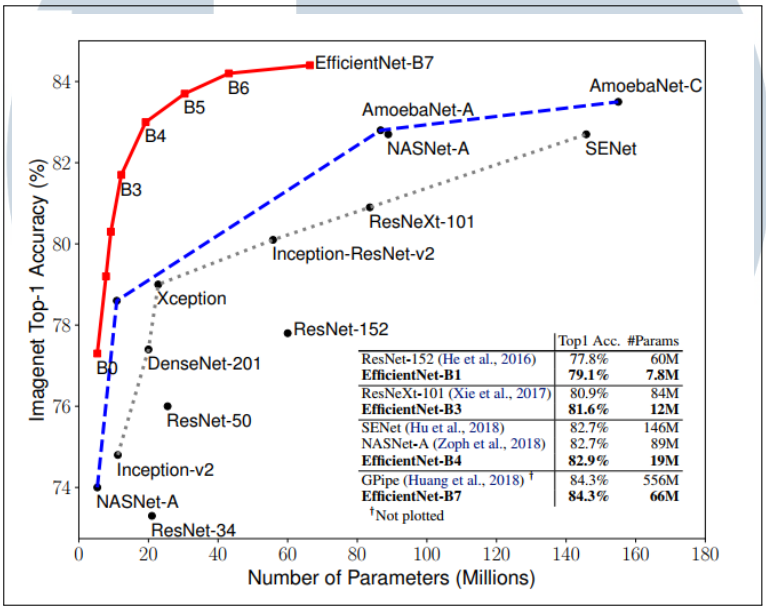
$$width : \omega = \beta^{\phi} \tag{2.2}$$

$$resolution : \tau = \gamma^{\phi} \tag{2.3}$$

Koefisien tetap ini adalah koefisien yang mengontrol sumber daya komputasi. Misalnya, jika kita ingin menggunakan sumber daya komputasi 2^{ϕ} kali lebih banyak, maka kita cukup meningkatkan kedalaman jaringan dengan α^{ϕ}, lebar de-

ngan β^{ϕ}, dan ukuran gambar dengan γ^{ϕ}. Nilai *alpha*, *beta*, dan *omega* merupakan koefisien konstan yang ditentukan oleh penelusuran petak kecil pada model kecil aslinya.

EfficientNet sendiri terdiri dari beberapa model yakni B0, B1, B2, B3, B4, B5, B6 dan B7. Dari semua model tersebut, B0 merupakan model yang menggunakan parameter paling sedikit dibanding yang lainnya.



Gambar 2.9. Jumlah Parameter vs Akurasi EfficientNet

Sumber: [19]

Pada penelitian ini, arsitektur EfficientNet B0 akan digunakan untuk diagnosis infeksi COVID-19. Secara garis besar, arsitekturnya dapat diuraikan sebagai berikut:

UNIVERSITAS MULTIMEDIA NUSANTARA

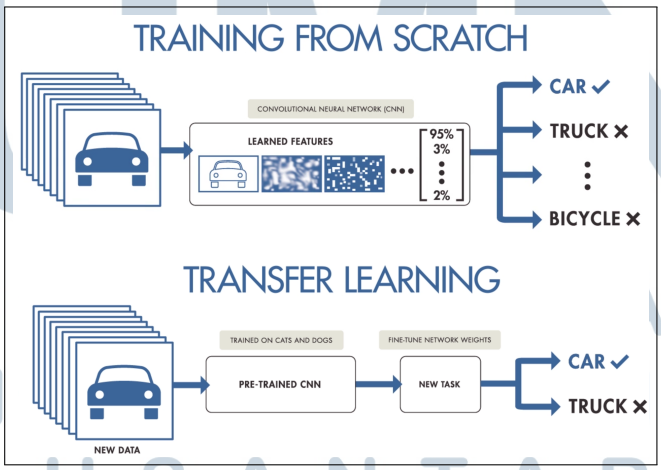
Tabel 2.1. Arsitektur EfficientNet B0

Layer	Operator	Resolution	Channels	Layers
1	Conv 3x3	224x224	32	1
2	MBConv1 3x3	112x112	16	1
3	MBConv6 3x3	112x112	24	2
4	MBConv6 5x5	56x56	40	2
5	MBConv6 3x3	28x28	80	3
6	MBConv6 5x5	14x14	112	3
7	MBConv6 5x5	14x14	192	4
8	MBConv6 3x3	7x7	320	1
9	Conv 1x1 & Pooling & FC	7x7	1280	1

Sumber: [19]

2.6 Transfer Learning

Transfer learning adalah algoritma *machine learning* yang menggunakan model sebelumnya untuk memecahkan tugas berikutnya. Tugas di sini merupakan tugas untuk menemukan bobot parameter yang memengaruhi output pada tiap kelas-kelas gambar yang ada. Metode ini sangat sering digunakan pada *Image Processing*. Tujuan penggunaannya yakni untuk mengurangi operasi yang dilakukan untuk melatih model yang akan dibuat [20].



Gambar 2.10. Gambaran *Transfer learning*
Sumber: [20]

Pada Gambar 2.10, *transfer learning* tidak melakukan perhitungan ulang,

namun menggunakan bobot (*weight*) yang ada sebelumnya untuk melakukan klasifikasi terhadap gambar tersebut sehingga mengurangi operasi yang dilakukan.

2.7 Fine Tuning Model

Untuk mendapatkan hasil akurasi model yang lebih tinggi dengan meminimalisir *loss* [13], maka dilakukan *fine-tuning* model dengan memperhatikan beberapa objek sebagai berikut:

2.7.1 Flatten

Flatten adalah fungsi yang mengubah nilai *array convolutional* dua dimensi (2D) atau *max pooling* dua dimensi menjadi *array* satu dimensi (1D) [21]. Misalnya inputnya adalah (8, 8). Maka, hasil setelah fungsi *flatten* adalah (64) dimana *array* diubah ke dalam bentuk satu dimensi.

2.7.2 Dense

Dense merupakan *hidden layer* yang menerima tiap input *neuron* dari *layer* sebelumnya dengan membawa bobotnya masing-masing. *Dense layer* menunjukkan adanya pergantian dimensi vektor dimana termasuk operasi seperti rotasi, *scaling*, maupun translasi [22].

2.7.3 Rectified Linear Unit

Rectified Linear Unit atau disingkat ReLU merupakan fungsi aktivasi yang menghasilkan inputnya apabila inputnya bernilai positif. Apabila inputnya adalah negatif, maka outputnya menjadi 0. Secara matematika, dapat dituliskan sebagai berikut:

$$yi = \begin{cases} xi & \text{if } xi \geq 0 \\ 0 & \text{if } xi < 0 \end{cases} \tag{2.4}$$

xi melambangkan input dari layer sebelumnya. Sementara itu, *yi* melambangkan hasil dari fungsi aktivasi ReLu tersebut [23].

2.7.4 Dropout

Dropout adalah teknik regularisasi dimana hasil *layer* yang dipilih secara acak akan dibuang nilainya supaya model menjadi kurang sensitif terhadap suatu bobot di *network*. Ini untuk mengurangi risiko model *overfitting* [21].

2.7.5 Sigmoid

Sigmoid yang biasa disebut *logistic function* adalah salah satu fungsi aktivasi yang digunakan untuk memprediksi hasil dari probabilitas terutama untuk masalah klasifikasi yang bersifat *binary*. Aktivasi ini juga dinilai mudah dimengerti dan kebanyakan digunakan untuk membangun *network* yang dangkal [24]. Secara matematika, dapat dituliskan sebagai berikut:

$$f(x) = \frac{1}{(1 + exp^{-x})} \tag{2.5}$$

xi melambangkan input dari *layer* sebelumnya.

2.7.6 Callback

Callback digunakan dalam penelitian guna menjalankan suatu fungsi setelah tiap epoch selesai. Ini berfungsi untuk mengontrol proses pelatihan model yang dilakukan. Beberapa *callback* yang digunakan, yakni:

1. *EarlyStopping*

   Pada penelitian pembelajaran mesin, model yang *overfitting* tidak baik digunakan. *Overfitting* adalah keadaan dimana model belajar terlalu detail pada data *training* sehingga berdampak buruk terhadap performa model dalam mengevaluasi data baru yang belum dipelajari sebelumnya [25]. Untuk menghindari *overfitting*, dilakukan penghentian proses latih model lebih awal. Dalam *library* keras, terdapat fungsi *EarlyStopping* yang akan menghentikan latihan jika metrik yang ditentukan tidak ada perubahan selama jumlah *epoch* yang ditentukan [26].

2. *ModelCheckpoints*

Agar model dapat disimpan dan digunakan kembali, maka fungsi *ModelCheckpoints* digunakan pada *callback* dengan argumen "save_best_only" yang hanya melakukan penyimpanan model dengan hasil yang terbaik [27].

3. *ReduceLROnPlateau*

   Fungsi ini digunakan untuk mengurangi *learning rate* saat metrik yang ditentukan berhenti berkembang [28].

2.8 Confusion Matrix

Confusion matrix adalah pengukuran performa dari tugas klasifikasi oleh *machine learning* [29]. Empat nilai utama pada *confusion matrix*, yakni:

1. *True Positive* / TP

   Hasil prediksi mengeluarkan nilai positif dan hasil sebenarnya adalah positif pula. Ini artinya hasil prediksi bernilai benar.

2. *False Positive* / FP

   Hasil prediksi mengeluarkan nilai positif dan hasil sebenarnya adalah negatif. Ini artinya hasil prediksi bernilai salah.

3. *True Negative* / TN

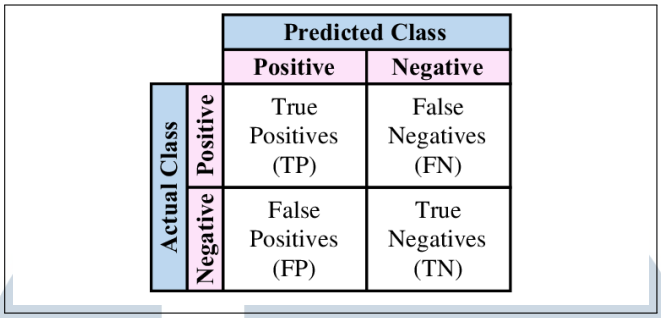
   Hasil prediksi mengeluarkan nilai negatif dan hasil sebenarnya adalah negatif. Ini artinya hasil prediksi bernilai benar.

4. *False Negative* / FN

   Hasil prediksi mengeluarkan nilai negatif dan hasil sebenarnya adalah positif. Ini artinya hasil prediksi bernilai salah.

Confusion matrix disajikan berbentuk tabel dengan kombinasi antara hasil prediksi dan hasil sebenarnya seperti pada Gambar 2.11.

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 2.11. Tabel *Confusion Matrix*

Sumber: [29]

Dari nilai yang dihasilkan oleh *confusion matrix*, maka dapat dihitung nilai akurasinya. Rumus akurasi sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

Pada penelitian, akurasi digunakan untuk menguji performa yang dihasilkan oleh masing-masing *pre-trained* model dimana data yang digunakan akan diseimbangkan antara kelas positif dan kelas negatif yang mana kedua kelas tersebut sama-sama penting sehingga keakuratan deteksi kedua kelas tersebut sangat dibutuhkan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA