

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Banyak penelitian yang telah dilakukan mengenai optimasi *hyperparameter* pada *convolutional neural network* dan klasifikasi aksara. Pada penelitian-penelitian sebelumnya yang menggunakan CNN, banyak variasi penggunaan *hyperparameter* untuk mendapatkan akurasi yang tinggi. Penelitian tersebut dijadikan sumber informasi dalam penelitian selanjutnya untuk mendapatkan hasil yang lebih baik. Berikut beberapa penelitian terdahulu yang relevan dengan penelitian ini.

Pertama, pada penelitian yang berjudul “Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah” [15] terdapat masalah peningkatan jumlah sampah yang ada di Indonesia pada tahun 2019. Untuk itu dibuat sebuah model klasifikasi sampah menggunakan CNN yang di optimasi. Didapat nilai akurasi yang telah dioptimasi menjadi lebih tinggi dibandingkan sebelum dilakukan optimasi .

Kedua, penelitian yang berjudul “Klasifikasi Akasara Jawa Dengan CNN” [16] dengan permasalahan mulai menurunnya penggunaan bahasa Jawa di jaman sekarang. Untuk itu dilakukan klasifikasi aksara Jawa menggunakan CNN. Didapatkan hasil model CNN yang digunakan mendapatkan nilai presisi sekitar 0.5 hingga 0.6 pada setiap kelasnya.

Ketiga, penelitian “Klasifikasi COVID-19 menggunakan FilterGabor dan CNN dengan HyperparameterTuning”[17] terdapat penelitian sebelumnya menggunakan Reverse-Transcription Polymerase Chain Reaction (RT-PCR), namun metod tersebut kurang efisien. Dari penelitian ini didapatkan hasil akurasi 97,9% dari pelatihan model CNN dengan *hyperparameter* tuning dan *Gabor filter*[17].

Keempat, penelitian yang berjudul “Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi” [6] dengan permasalahan utama pada penelitian ini adalah penurunan jumlah produksi padi dalam negeri. Salah satu penyebab dari menurunnya produksi padi adalah rusaknya

tanaman padi karena di serang penyakit. Akibat terburuk yang akan dialami adalah petani akan gagal panen. Untuk itu dilakukan klasifikasi penyakit pada padi menggunakan CNN serta dilakukan optimasi *hyperparameter*. Pada penelitian ini didapatkan hasil nilai *accuracy* 97,56%, *precision* 97,64%, *recall* 97,57% dan *f1-score* 97,57%. Hasil ini

didapatkan dengan pelatihan menggunakan 100 epoch, batch size 32 dan learning rate 0,001.

Kelima, penelitian “Human Skin Disease Detection using Convolutional Neural Network Method with Hyperparameter Tuning to Determine the Best Parameter Combination”. [7] dalam penelitian ini terdapat permasalahan penyakit kulit yang menduduki peringkat ketiga sebagai penyakit dengan penularan secara langsung di Indonesia. Penyebab penularan penyakit kulit ini disebabkan beberapa faktor seperti kurangnya kesadaran mengenai identifikasi penyakit kulit. Itu itu dilakukan identifikasi jenis penyakit kulit menggunakan *convolutional neural network* yang di optimasi *hyperparameter* untuk mendapatkan hasil yang lebih optimal. Pada penelitian ini didapatkan hasil nilai *accuracy* 97,81%.

Tabel 2.1 Penelitian Sebelumnya

| No | Judul Penelitian | Metode | Masalah | Hasil | Perbedaan |
|----|---|---|--|--|---|
| 1. | Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah (Rima Dias Ramadhani, Afandi Nur Aziz Thohari, Condro Kartiko, Apri Junaidi, Tri Ginanjar Laksana, Novanda dan Alim Setya Nugraha, 2021) | <i>Convolutional Neural Network</i> dengan penyesuaian <i>hyperparameter dropout, padding, stride</i> | Peningkatan jumlah sampah yang ada di Indonesia dari tahun sebelumnya, dimana pada tahun 2019 sampah yang ada di Indonesia mencapai 64 juta ton. Dari 64 juta ton ini hampir 60% sampah tersebut adalah sampah organik. Peningkatan jumlah sampah tidak membuat kesadaran masyarakat untuk memilah sampah sesuai dengan golongannya, | Dari hasil pelatihan model yang dilakukan didapatkan model CNN tanpa adanya optimasi <i>hyperparameter</i> hanya mendapatkan nilai akurasi 67,7% pada klasifikasi jenis sampah. Sedangkan pada model yang dilakukan optimasi <i>hyperparameter</i> mendapatkan akurasi yang jauh lebih tinggi, yakni 91,2% | Perbedaan dengan penelitian ini adalah penggunaan parameter <i>hyperparameter</i> . <i>Hyperparameter</i> yang digunakan kernel filter, <i>dropout</i> dan <i>learning rate</i> . Selain itu pada penelitian ini menggunakan <i>accuracy, recall, precision</i> dan <i>f1-score</i> untuk evaluasi model. |

| No | Judul Penelitian | Metode | Masalah | Hasil | Perbedaan |
|----|---|-------------------------------------|---|--|--|
| 2. | Klasifikasi Aksara Jawa Dengan CNN (Edo Prasetyo N.A. Wijaya, Hendrawan Armanto,Lukman Zaman,2020) | <i>Convolutional Neural Network</i> | Pergunaan aksara jawa ngalegena mulai kurang diminati seiring kemajuan zaman. Untuk itu dilakukan klasifikasi aksara jawa agar bahasa | Hasil dari penelitian ini didapatkan hasil dari pelatihan model dari setiap kelas dengan rata-rata keseluruhan sebagai berikut : presisi = | Perbedaan dengan penelitian ini adalah dilakukan optimasi <i>hyperparameter</i> pada CNN. Dilakukan <i>preprocessing</i> data. Selain itu pada |

| No | Judul Penelitian | Metode | Masalah | Hasil | Perbedaan |
|----|---|--|--|---|---|
| | | | jawa tidak hilang seiring kemajuan zaman. | 0.56, recall = 0.57 dan F-measure = 0.57 | penelitian ini menggunakan <i>accuracy</i> , <i>recall</i> , <i>precision</i> dan <i>f1-score</i> untuk evaluasi model. |
| 3. | Klasifikasi menggunakan FilterGabor dan CNN dengan HyperparameterTuning(Afis Julianto, Andi Sunyoto, dan Ferry Wahyu Wibowo) | <i>Filter galbor</i> dan <i>convolutional neural network</i> dengan <i>hyperparameter tuning</i> | Penyebaran virus Corona atau COVID-19 menyebar dengan sangat cepat. Untuk menekan jumlah penyebaran COVID-19 dilakukan deteksi dan diagnosis sejak dini menggunakan Reverse-Transcription Polymerase Chain Reaction (RT-PCR). Namun, metode tersebut | Hasil dari penelitian ini mendapat hasil terbaik dengan nilai akurasi sebesar 97.7% dan AUC sebesar 99% . | Perbedaan dengan penelitian ini adalah terdapat pada optimasi <i>hyperparameter</i> dan pada penelitian ini tidak menggunakan filter gabor. Selain itu dataset yang digunakan juga berbeda dengan penelitian ini. |

| No | Judul Penelitian | Metode | Masalah | Hasil | Perbedaan |
|----|---|---|---|--|--|
| 4 | Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi(Afis Julianto, Andi Sunyoto, dan Ferry Wahyu Wibowoyang, tahun 2022) | CNN dengan optimasi <i>hyperparameter</i> | Permasalahan utama pada penelitian ini adalah penurunan jumlah produksi dalam negeri. Salah satu penyebab dari menurunnya produksi padi adalah rusaknya tanaman padi karena di serang penyakit. Akibat terburuk yang akan | Pada penelitian ini didapatkan hasil nilai accuracy 97,56%, precision 97,64%, recall 97,57% dan f1-score 97,57%. Hasil ini didapatkan dengan pelatihan menggunakan 100 <i>epoch</i> , <i>batch size</i> 32 | Perbedaan terdapat pada objek yang diteliti, penggunaan <i>hyperparameter</i> yang berbeda. Dimana dalam penelitian ini menggunakan penyesuaian <i>learning rate</i> , <i>kernel filter</i> dan <i>dropout</i> . |

| No | Judul Penelitian | Metode | Masalah | Hasil | Perbedaan |
|----|--|---|---|-------------------------|--|
| 5 | <i>Human Skin Disease Detection using Convolutional Neural Network Method with Hyperparameter Tuning to Determine the Best Parameter Combination</i> (Riki Martua Aritonang, Mangaras Yanu) | CNN dengan optimasi <i>hyperparameter</i> (3 <i>convolutional layer</i> , 0,05 <i>APL dropout</i> , 0,5 <i>dropout layer</i> , 64 <i>dense layer</i> dan <i>optimizer RMSProp</i>) | dialami adalah petani akan gagal panen. Untuk itu dilakukan klasifikasi penyakit pada padi menggunakan CNN serta dilakukan optimasi <i>hyperparameter</i> Kurangnya kesadaran mengenai identifikasi penyakit kulit. Itu itu dilakukan identifikasi jenis penyakit kulit menggunakan <i>convolutional neural network</i> yang di optimasi <i>hyperparameter</i> untuk | dan learning rate 0,001 | Perbedaan terdapat pada objek yang diteliti, serta penggunaan <i>hyperparameter</i> yang berbeda. Dimana dalam penelitian ini menggunakan penyesuaian learning |

| No | Judul Penelitian | Metode | Masalah | Hasil | Perbedaan |
|----|---|--------|--|----------------------------------|-------------------------------------|
| | Florestiyanto dan Bambang, tahun 2023) | | mendapatkan hasil yang lebih optimal. | tuning dengan akurasi 97,81%. | rate, kernel filter dan dropout. |

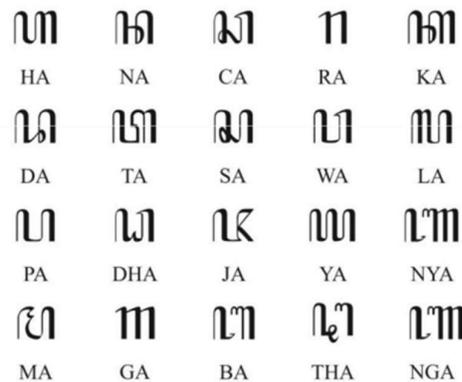
2.2 Landasan Teori

2.2.1 Aksara

Aksara merupakan sistem informasi yang dituangkan dalam bentuk simbol visual pada media kertas maupun media lainnya seperti kayu, batu, kain dan lainnya[18]. Aksara merupakan salah satu bentuk representasi bahasa, karena aksara merupakan salah satu cara manusia dalam menyampaikan informasi. Aksara kuat kaitannya dengan bahasa daerah yang ada di Indonesia, namun tidak semua bahasa daerah memiliki aksara. Contoh beberapa bahasa daerah yang memiliki aksara adalah aksara Jawa, aksara Bali, aksara Sunda dan aksara Lontara [19]

1. Aksara Jawa

Aksara Jawa, merupakan aksara tradisional dari Indonesia yang cara penulisannya menggunakan bahasa Jawa dan merupakan perkembangan dari turunan aksara Brahmi [20]. Huruf Jawa juga dikenal sebagai Hancaraka dan Carakan yang digunakan dalam penulisan aksara Jawa [21]. Aksara Jawa terdiri dari beberapa bentuk, yaitu aksara Carakan, aksara Pasangan, Sandhangan, aksara Murda, aksara Rekan, aksara Swara dan angka Jawa [20].



Gambar 2.1 Aksara Jawa

2. Aksara Bali

Aksara Bali, merupakan aksara yang memiliki aksara alfabilitas yang mewajibkan aksara segmental pada konsonan dengan notasi vokal namun bersifat sekunder. Aksara Bali dibagi menjadi dua bagian, yaitu aksara suci dan aksara biasa. Pada aksara suci terdapat aksara wijaksana dan aksara modre

yang biasanya digunakan dalam hal magis dan agamis . Kemudian pada aksara biasa meliputi aksara wreastra dan aksara swalalita yang sering digunakan pada kehidupan sehari-hari masyarakat Bali [22].



Gambar 2.2 Aksara Bali

3. Aksara Sunda

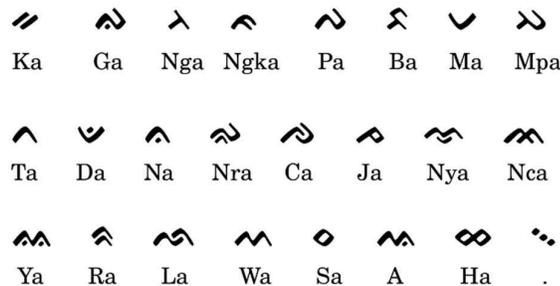
Aksara Sunda, merupakan aksara yang sudah ada sejak sekitar abad 5M dan berasal dari tanah sunda [23]. Aksara sunda dilestarikan oleh suku-suku sunda yang berada di sekitaran Provinsi Jawa Tengah. Aksara Sunda terdiri dari empat(4) bagian, yakni Aksara Swara, Aksara Ngalagena, Aksara Angka dan Aksara Rarangkeun. Ada beberapa daerah di Indonesia dimana masyarakatnya dominan menggunakan bahasa sunda, seperti Bogor, Cianjur dan Bandung [24].



Gambar 2.3 Aksara Sunda

4. Aksara Lontara

Aksara Lontara, merupakan aksara tradisional yang digunakan oleh suku bugis yang berasal dari Makasar. Aksara lontaran merupakan perkembangan dari huruf kawi yang digunakan di Indonesia [25]. Aksara Lontara terdiri dari dua bagian, yakni huruf konsonan yang berjumlah dua puluh tiga(23) dan huruf vokal yang berjumlah lima(5). Pada umumnya aksara Lontara digunakan untuk menuliskan tata aturan pemerintah dan kemasyarakatan [19].



Gambar 2.4 Aksara Lontara

2.2.2 Deep Learning

Deep learning merupakan salah satu cabang ilmu dari *machine learning* yang bekerja seperti jaringan otak pada manusia dalam memahami dan memproses data yang kompleks [3]. *Deep learning* melakukan adaptasi dengan menggunakan pendekatan *multilayer* ke lapisan *neural network* sehingga dapat mengerjakan berbagai tugas dengan kompleksitas yang tinggi seperti gambar beresolusi tinggi, sinyal suara, dan teks yang panjang [20] [21]. Dengan kemampuan tersebut *deep learning* mampu untuk memproses data yang sangat besar dan kompleks sehingga membuatnya sangat berarti dalam menyelesaikan tugas-tugas yang sulit. Selain itu *deep learning* memberikan kemajuan yang sangat signifikan dalam memajukan komputer untuk memahami dan memproses data yang membuat kemajuan besar dalam berbagai aspek kehidupan. Terdapat beberapa model *deep learning*, seperti *Convolutional Neural Network (CNN)*, *Recurrent Neural Networks (RNNs)*,

Transformers, Autoencoders, Deep Belief Networks (DBNs), Siamese Networks dan Hybrid Models.

2.2.3 Preprocessing

Preprocessing merupakan tahapan mengubah data yang sebelumnya masih berupa data mentah menjadi data dengan format yang sesuai dengan pelatihan model yang digunakan [17]. Tujuan dari *preprocessing* dalam pengolahan citra adalah untuk memastikan kualitas dan konsistensi data untuk pelatihan [9]. Dalam pengolahan citra, terdapat beberapa teknik dalam *preprocessing* data, seperti *noise reduction, filtering, normalization, contrast enhancement, edge detection, image segmentation dan feature extraction.*

2.2.4 Augmentasi

Augmentasi adalah tahapan mengubah atau menambahkan data yang sebelumnya didapatkan dengan tujuan meningkatkan jumlah dan karakteristik sehingga data yang dimiliki menjadi lebih beragam. Tahapan ini dapat membantu meningkatkan kinerja model dalam melakukan pelatihan sehingga bisa terhindar dari *overfitting* dan *underfitting* terutama pada penelitian dengan data yang sedikit [28]. Namun, penggunaan teknik augmentasi yang tidak tepat dapat mengakibatkan meningkatnya kompleksitas model serta meningkatkan bias model. Ada beberapa teknik yang dapat digunakan dalam pengolahan citra, seperti rotasi, refleksi, pembesaran, pengurangan, penambahan noise, perubahan pencahayaan, perubahan warna dan lain sebagainya.

2.2.5 Feature Extraction

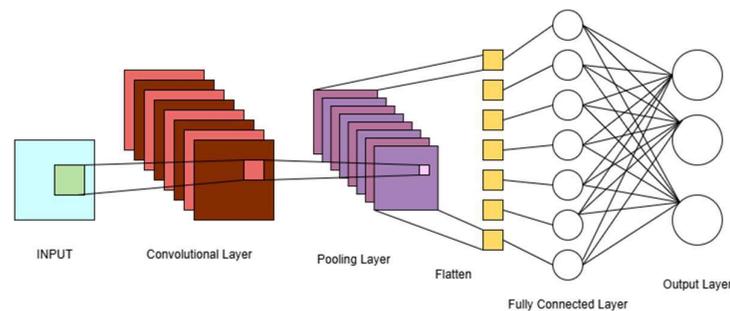
Feature extraction merupakan sebuah proses mengidentifikasi dan mengekstraksi fitur-fitur penting dari data menggunakan filter-filter tertentu. Ini merupakan tahap awal dari pengolahan citra yang memungkinkan model untuk melakukan ekstraksi pada data citra dan menghasilkan output yang nantinya akan digunakan pada tahap berikutnya. Hasil dari proses *feature extraction* nantinya akan digunakan untuk berbagai hal dalam *deep learning*, seperti klasifikasi, pengenalan

pola dan deteksi objek. Proses *feature extraction* akan menghasilkan karakteristik penting dari data, seperti seperti tepi objek, tekstur, atau pola-pola visual lainnya [29]. Hal ini akan sangat membantu model dalam memahami informasi yang terkandung dalam data, terutama pada data berbentuk citra [30].

2.2.6 Convolutional Neural Network (CNN)

Convolutional neural networks merupakan bagian dari *deep learning* dengan *multi layer perceptron* (MLP) yang sering digunakan dalam klasifikasi dan pengolahan citra. *Multi layer perceptron*(MLP) adalah arsitektur jaringan saraf tiruan *feedforward network*, dimana informasi mengalir dari input ke output tanpa ada loop dalam jaringan. Secara sederhana CNN bekerja dengan memanfaatkan proses konvolusi untuk memindahkan filter berukuran tertentu ke sebuah gambar. Hasil perkalian bagian gambar dengan filter menghasilkan informasi representatif baru.

Convolutional neural network terdiri atas kumpulan neuron atau *layer* yang memiliki bobot dan bias. Pada umumnya CNN terdiri dari beberapa lapisan seperti *Convolutional Layer*, *Pooling Layer*, *Flatten* dan *Fully Connected Layer* [31].



Gambar 2.5 Arsitektur Convolutional Neural Network

2.2.6.1 Convolutional Layer

Convolutional layer atau biasa disebut juga dengan kernel terdiri dari kumpulan filter yang dapat dipelajari [32]. Setiap kernel memiliki ukuran dua dimensi, dimana pada umumnya berdimensi 3x3, 5x5 atau 7x7. Selain itu kernel

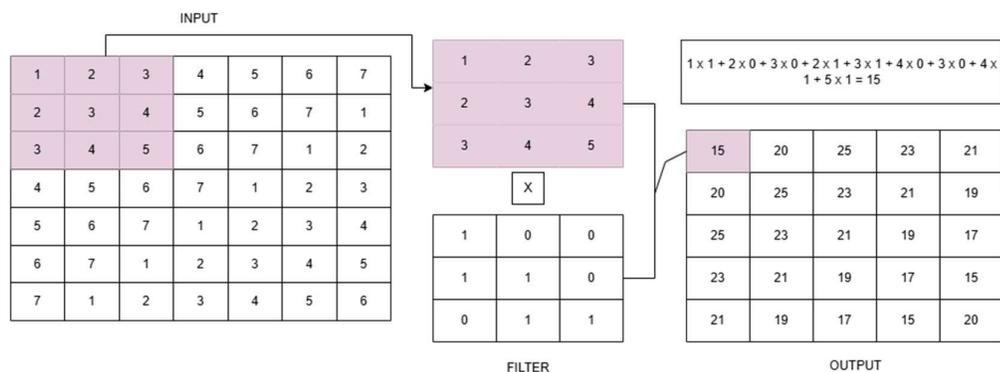
juga memiliki dimensi tambahan atau dimensi ketiga yang merupakan saluran warna pada gambar. Pada gambar berwarna, dimensi ketiga pada kernel akan bernilai tiga (3) yang mewakili saluran warna *red*, *green*, *blue* (RGB), contoh 3x3x3. Sedangkan pada gambar yang berwarna hitam putih dimensi ketiga pada kernel akan bernilai satu(1) yang mewakili saluran warna grayscale, contoh 3x3x1.

Pada prosesnya, filter pada kernel akan bergeser ke seluruh bagian pada gambar dan melakukan operasi AND pada setiap fitur antara nilai filter dan input. Hasil atau output dari proses ini akan menghasilkan *feature map* atau *activation map* yang nantinya akan digunakan untuk inputan pada proses selanjutnya. Perhitungan pada *convolutional layer* untuk mendapatkan output sesuai dengan input dapat dituliskan sebagai berikut:

$$output = \left(\frac{w-k+2p}{s} \right) + 1 \quad (1)$$

Dimana :

- w = ukuran gambar / citra
- k = ukuran filter
- p = padding (penambahan nilai nol pada tepi data)
- s = stride (jarak pergeseran filter)

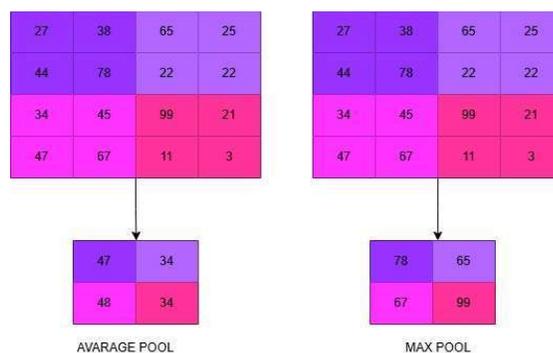


Gambar 2.6 Convolutional Layer

Gambar 2.6, merupakan contoh proses konvolusi antara matrik dari inputan dengan matriks filter untuk menghasilkan citra baru. Nilai yang terdapat pada matrix filter pada awalnya diperoleh secara acak atau dengan metode tertentu seperti *He initialization* dan *Xavier initialization* [33]. Kemudian pada tahap pelatihan selanjutnya nilai pada matrix filter akan dioptimalkan dengan melibatkan *forward pass*(untuk menghitung keluaran) dan *backward pass*(menghitung gradien dari fungsi kerugian terhadap setiap parameter).

2.2.6.2 Pooling Layer

Lapisan pooling berfungsi sebagai pengurang ukuran spasial dari konvolusi. Hal ini dilakukan agar mengurangi penggunaan sumber daya komputasi selama memproses data [34]. Pengurangan ukuran spasial dilakukan dengan cara mengurangi dimensi dari feature map(*downsampling*). Dalam proses ini jumlah parameter yang diperbarui akan berkurang, sehingga dapat mempercepat komputasi. Lapisan pooling juga dapat mengekstrak fitur dominan agar pelatihan model menjadi lebih efektif [35]. *Pooling layer* terdiri dari dua jenis yakni, *max pooling layer* dan *average pooling layer*.



Gambar 2.7 Pooling Layer

2.2.6.3 Activation Function

Fungsi aktivasi adalah jenis fungsi non-linear yang digunakan oleh jaringan saraf tiruan untuk mengubah data masukan ke dimensi yang lebih tinggi [36]. Hal ini bertujuan agar jaringan dapat melakukan pemotongan sederhana pada fitur-fitur aktivasi yang signifikan untuk meningkatkan pemahaman citra.. Fungsi dari

activation function adalah untuk melakukan pemetaan pada input ke output. Terdapat beberapa *activation function* yang sering digunakan, diantara :

- Sigmoid

Sigmoid adalah fungsi aktivasi yang menerima bilangan real sebagai input dan menghasilkan output antara 0 dan 1 [37]. Fungsi ini sederhana dan baik untuk proses klasifikasi, tetapi membutuhkan waktu komputasi yang lama.

$$f(x)_{\text{sigm}} = \frac{1}{1+e^{-x}} \quad (2)$$

Dimana :

- e = basis logaritma
- x = input dari fungsi sigmoid (berupa nilai hasil dari neuron atau unit dalam jaringan)

- *Softmax*

Softmax adalah fungsi aktivasi yang sering digunakan pada kasus klasifikasi multi kelas [38]. Fungsi ini menghasilkan probabilitas untuk setiap kelas, dengan probabilitas totalnya selalu sama dengan 1.

$$\text{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^k e^{z_k}} \quad (3)$$

Dimana :

- $\text{Softmax}(z_j)$ = probabilitas kelas j
- e = bilangan euler
- z_j = keluaran kelas j dari fully connected layer
- K = jumlah total elemen dalam vektor Z

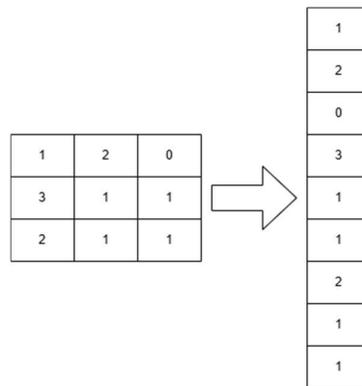
- ReLu

ReLU adalah fungsi aktivasi yang paling umum digunakan pada jaringan saraf konvolusional (CNN). ReLU mengubah nilai input menjadi

nilai positif, sehingga dapat mempercepat proses pembelajaran jaringan saraf tiruan [39]. Aktivasi ini akan merubah nilai dari setiap piksel yang kurang dari nol dari suatu citra menjadi bernilai nol [38]. Probabilitas yang dihasilkan oleh ReLU mirip dengan sigmoid, tetapi ReLU memiliki kinerja yang lebih baik. Rumus ReLU sangat sederhana, yaitu $f(x)_{ReLU} = \max(0, z)$.

2.2.6.4 Flattening

Flattening merupakan proses transformasi representasi data dari bentuk matriks multidimensial ke dalam vektor satu dimensi. Proses ini diterapkan agar mengubah inputan agar sesuai dengan format pada proses selanjutnya [39]. Hal ini bertujuan agar data masukkan berubah menjadi data yang dapat digunakan pada *fully connected layer*.



Gambar 2.8 Flattening

2.2.6.5 Fully Connected Layer

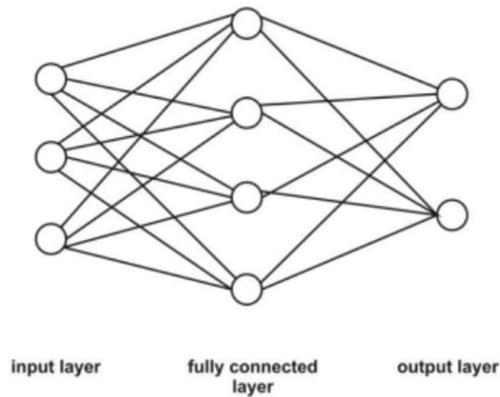
Fully connected layer atau dikenal juga dengan lapisan *Dense*. *Fully Connected* terhubung ke setiap neuron dalam lapisan sebelumnya. Ini berarti setiap input dari lapisan sebelumnya dihubungkan ke setiap neuron di lapisan ini [38]. Konektivitas penuh ini mengizinkan lapisan *fully Connected* untuk mengakses seluruh informasi yang telah diambil dari lapisan sebelumnya. Pada *fully connected layer* akan memproses inputan berupa vektor dari *pooling layer* atau *convolutional layer* akhir dan menghasilkan Output yang merupakan output terakhir dalam proses.

Proses *fully connected* dapat ditampilkan dengan perkalian matriks sederhana yang dapat dilihat sebagai berikut :

$$output = f(w^t x + b) \quad (4)$$

Dimana :

- f = fungsi aktivasi
- w = matriks bobot yang menghubungkan setiap neuron
- t = operasi transposisi
- x = vektor input / output dari proses sebelumnya
- b = bias



Gambar 2.9 Fully Connected Layer

2.2.7 Optimasi Hyperparameter

Optimasi *hyperparameter* merupakan sebuah proses penyesuaian parameter-parameter tertentu dalam model *machine learning* dan *deep learning* untuk mendapatkan hasil yang lebih baik [15]. Optimasi *hyperparameter* dilakukan karena pengaruh dari penyesuaian *hyperparameter* sangat signifikan dalam meningkatkan akurasi dari model yang digunakan. Selain itu dengan penyesuaian *hyperparameter* dapat mencegah model mengalami *overfitting* dan *underfitting*. Penentuan nilai dalam optimasi *hyperparameter* harus ditentukan sebelum proses

training dan dilakukan secara manual, ini dikarenakan model tidak dapat mempelajari dalam menentukan parameter *hyperparameter* [6].

Dalam CNN, terdapat beberapa parameter hyperparameter yang dapat digunakan untuk meningkatkan performa model CNN dalam mengelola citra. Beberapa diantaranya :

1. *Learning rate*

Learning rate merupakan parameter yang digunakan dalam pelatihan model jaringan saraf tiruan yang berfungsi dalam menghitung setiap langkah pada iterasi guna meminimalkan *loss function* [40]. Nilai *learning rate* terbaik dapat dicapai dengan menggunakan vektor gradien yang bernilai negatif. Hal ini karena vektor gradien negatif menunjukkan bahwa model pembelajaran mesin sedang bergerak ke arah nilai error yang lebih rendah. Nilai error yang lebih rendah berarti bahwa model pembelajaran mesin telah belajar dengan baik dan dapat menghasilkan output yang akurat [41]. Fungsi *learning rate* untuk memfokuskan model pembelajaran mesin pada komponen-komponen penting dari data masukan, sekaligus mengurangi pengaruh dari varian-varian yang tidak relevan.

2. *Dropout*

Dalam upaya pencegahan *overfitting* pada CNN penerapan *dropout* menjadi salah satu solusi. *Dropout* merupakan teknik regulasi dalam mencegah terjadinya *overfitting* pada jaringan saraf tiruan. Untuk mencegah terjadinya *overfitting* dropout akan menghilangkan beberapa neuron secara acak pada lapisan CNN selama pelatihan model [41]. Untuk setiap neuron yang hilang akan digantikan dengan nilai nol. Hal ini dilakukan agar model tidak bergantung pada satu set neuron saja. Secara umum nilai *dropout* yang digunakan pada model CNN adalah 0,2 atau 0,5.

3. Kernel filter

Filter kernel adalah parameter-parameter yang dapat diubah oleh pengguna untuk mengontrol kinerja model convolutional neural network (CNN) [39]. Penyesuaian hyperparameter filter kernel merupakan proses

eksperimental yang melibatkan percobaan dan pemantauan kinerja model selama pelatihan. Ini penting untuk menyesuaikan hyperparameter ini agar sesuai dengan tugas klasifikasi.

2.2.8 Confusion Matrix

Confusion matrix merupakan table yang digunakan untuk mengevaluasi kinerja model dalam klasifikasi. *Confusion matrix* digunakan untuk mengetahui kinerja dan tingkat kebenaran suatu model dalam melakukan klasifikasi. *Confusion matrix* berbentuk sebuah matrix dan terdiri empat komponen utama. *True positive* (TP), jumlah data yang benar-benar positif dan diprediksi positif oleh model. Model dapat dengan tepat mengidentifikasi data yang positif. *False negative* (FN), jumlah data yang benar-benar positif tetapi diprediksi negatif oleh model. Model gagal mengidentifikasi data yang positif. *False Positive* (FP), jumlah data yang benar-benar negatif tetapi diprediksi positif oleh model. Model salah mengidentifikasi data yang negatif. *True negative* (TN), jumlah data yang benar-benar negatif dan diprediksi negatif oleh model. Model dapat dengan tepat mengidentifikasi data yang negative [43] [31].

Tabel 2.2 Confusion Matrix

| | | Kelas hasil deteksi | |
|--------------|---------|---------------------|---------|
| | | Positif | Negatif |
| Kelas aktual | Positif | TP | FN |
| | Negatif | FP | TN |

Dari kolom *confusion matrix* dapat ditentukan perhitungan *accuracy*, *precision*, *recall* dan *f1-score* untuk mengetahui performa dan keakuratan model dalam melakukan klasifikasi.

1. *Accuracy*

Nilai dari *accuracy* menunjukkan keakuratan model dalam mengklasifikasi data dengan benar atau tingkat kebenaran antara nilai deteksi dengan nilai sebenarnya [42] [43]. Dengan kata lain nilai dari akurasi akan menampilkan seberapa akurat hasil prediksi dari suatu model dengan data aktual atau sebenarnya. Secara matematis dapat ditulis sebagai berikut :

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

2. *Precision*

Precision adalah ukuran seberapa akurat model dalam memprediksi kelas benar positif (*true positive*). Semakin tinggi nilai presisi, semakin akurat model dalam memprediksi kelas positif. Nilai presisi yang tinggi berarti model tidak sering salah memprediksi kelas positif [42] [43]. Selain itu nilai presisi yang tinggi menunjukkan bahwa model dapat diandalkan. Secara matematis, presisi dirumuskan sebagai:

$$precision = \frac{TP}{TP+FP} \quad (6)$$

3. *Recall*

Recall adalah ukuran yang menunjukkan berapa banyak model mendeteksi kelas aktual positif yang terdeteksi benar. Semakin tinggi nilai recall, semakin baik model dalam mendeteksi kelas positif. Nilai *recall* menunjukkan model berhasil dalam menentukan Kembali sebuah informasi. Nilai recall yang tinggi berarti model tidak sering gagal mendeteksi kelas positif [43], [44]. Secara matematis, recall dirumuskan sebagai:

$$recall = \frac{TP}{TP+FN} \quad (7)$$

4. *F1-score*

F1-Score adalah sebuah metrik yang menggambarkan nilai rata-rata dari *precision* dan *recall*, sehingga memberikan gambaran komprehensif tentang kualitas *precision* dan *recall* secara bersamaan. Nilai maksimal *F1-Score* diperoleh ketika *precision* dan *recall* memiliki nilai yang sama, sehingga semakin mendekati nilai 1, semakin baik kinerja model klasifikasi [8]. Dalam rumus matematis, *F1-Score* dapat diungkapkan sebagai:

$$f1 - score = 2 \times \frac{precision \times recall}{precision + recall}$$

(8)