

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian sebelumnya dapat digunakan sebagai referensi untuk melakukan penelitian ini.

Table I. Penelitian Terdahulu

NO	JUDUL PENELITIAN	PENULIS DAN TAHUN	METODE	HASIL AKURASI
1	Klasifikasi Citra Pigmen Kanker Kulit Menggunakan <i>Convolutional Neural Network</i>	Luqman Hakim, dkk, 2021	Convolutional Neural Network (CNN) dengan arsitektur model menggunakan 8 layer Convolutional 2D.	Laporannya menunjukkan bahwa hasil pengujian dan evaluasi menunjukkan akurasi sebesar 75%, dengan nilai precision dan recall tertinggi masing-masing sebesar 0.80 dan 0.82 di kelas benign, dan nilai f1_ sebesar 0.81.
2	Klasifikasi Kanker Kulit menggunakan Metode <i>Convolutional Neural Network</i> dengan Arsitektur VGG-16.	Regita, dkk, 2022	CNN Arsitektur VGG16	Hasilnya adalah 99.70% akurasi, 0.0055 loss, precision 0.9975, recall 0.9975, dan skor f1 0.9950.
3	<i>Skin Cancer Classification and Comparison of Pre-trained Models Performance using Transfer Learning</i>	Ssingha, dkk, 2022	ResNet, VGG16, dan MobileNetV2	Dalam pengujian validasi berhasil mencapai tingkat akurasi sekitar 98,39% dengan ketiga model yang dilatih sebelumnya.

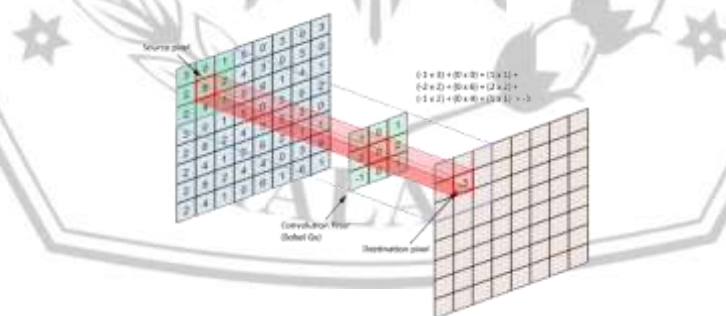
2.2 CNN

Yann LeCun adalah pencipta pertama dari Convolutional Neural Network (CNN) pada sekitar tahun 1980. Untuk memproses data visual, Metode deep learning menggunakan convolutional neural networking (CNN) [7]. CNN telah memainkan peran penting dalam perkembangan deep learning sejak awal, dan sekarang menjadi alat yang berhasil untuk menerapkan konsep kerja otak ke dunia machine learning. Jika digunakan secara komersial, model ini juga merupakan peran penting.

Convolutional Neural Network (CNN) dapat mengatasi masalah MLP, terutama ketika gambar yang digunakan untuk pelatihan berbeda dari gambar yang digunakan untuk validasi. MLP biasanya memiliki kemampuan untuk mengklasifikasikan gambar validasi dengan cara yang berbeda dari gambar yang dilatih. Dengan menggunakan filter deteksi yang dapat mengenali gambar di mana pun, CNN dapat mengatasi masalah ini. Struktur CNN terdiri dari tiga jenis lapisan: lapisan convolution, lapisan pooling, dan lapisan yang penuh terhubung.

2.2.1 Convolution Layer

Neuron yang terdiri dari lapisan convolution memiliki kemampuan untuk membuat filter dengan dimensi yang panjang dan tinggi. Input yang akan diterima oleh lapisan ini akan melewati filter, yang kemudian akan menghasilkan output berupa feature map.



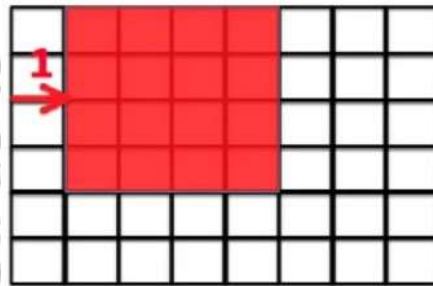
Gambar 2.1 Convolution Layer

(Sumber: freecodecamp.org)

Output yang akan dihasilkan oleh convolution layer akan diteruskan ke pooling layer untuk dikurangi dimensinya.

2.2.1.1 Stride

Stride adalah suatu parameter pada convolution layer yang dapat menentukan seberapa jauh filter akan bergeser melalui convolution Layer tersebut. Semakin besar nilai parameter pada stride, semakin banyak pergeseran yang akan dilakukan oleh filter. Ini dapat mempersingkat waktu komputasi tetapi juga dapat mengurangi jumlah informasi yang diperoleh.

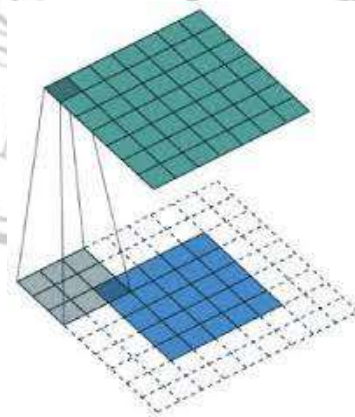


Gambar 2.2 Stride

(Sumber: medium.com)

2.2.1.2 Padding

Penambahan Padding dilakukan untuk meningkatkan dimensi convolution Layer. Tujuannya adalah untuk menjaga agar setiap piksel pada lapisan pergeseran dapat dikunjungi dengan jumlah yang sama, sehingga lapisan pergeseran tidak berbeda satu sama lain.



Gambar 2.3 Padding

(Sumber: towardsdatascience.com)

Penerapan Padding dapat membantu mengurangi resiko pada overfitting karena setiap piksel dikunjungi dengan merata, penggunaan padding juga dapat menghasilkan feature map yang lebih besar pada output.

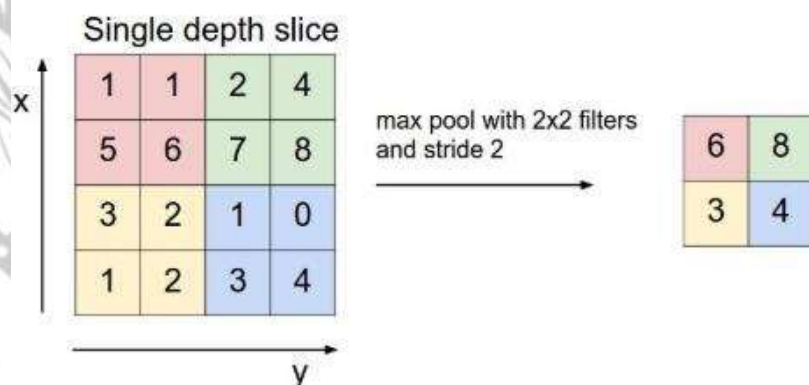
$$Output = \frac{w - N + 2P}{s} + 1$$

Dimana:

- W = Input panjang/tinggi
- N = Filter panjang/tinggi
- P = Tidak ada padding
- S = Stride

2.2.2 Pooling Layer

Pooling Layer memiliki prinsip kerja yang serupa dengan convolution layer, perbedaanya terletak pada tujuan filter yang akan digunakan oleh pooling layer. Fitur pada pooling layer tidak digunakan untuk mendeteksi sebuah fitur, melainkan untuk menemukan nilai maksimum atau rata-rata. Proses ini dikenal sebagai Max Pooling dan Average Pooling.



Gambar 2.4 Pooling Layer dengan Max Pooling

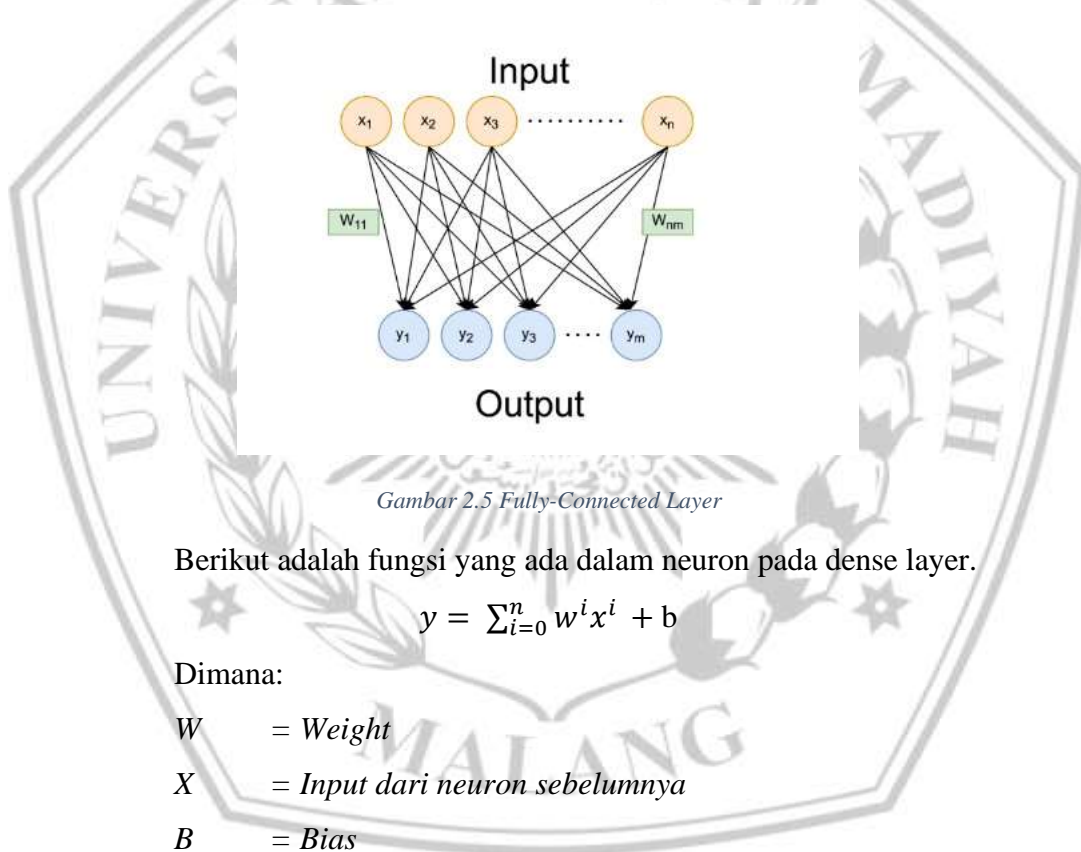
(Sumber: machinelearning.mipa.ugm.ac.id)

Pooling bertujuan untuk mengurangi dimensi output dari convolution layer yang juga disebut sebagai feature map, dengan demikian dapat mengurangi jumlah dari parameter yang perlu diolah dan mempercepat proses pelatihan model.

2.2.3 Fully-Connected Layer

Jaringan neural Fully-Connected (FC) lapisan mengategorikan data menggunakan output lapisan convolution dan pooling. FC layer dapat menerima masukan dalam bentuk larik 1D, Oleh karena itu output yang dikeluarkan oleh layer sebelumnya perlu diubah menjadi bentuk larik 1D array terlebih dahulu [10].

Setiap Neuron dalam Fully-Connected (FC) terdiri dari fungsi regresi dan fungsi activation. Fungsi regresi bertanggung jawab untuk memprediksi output, sementara fungsi activation bertanggung jawab untuk menentukan output yang akan diteruskan ke neuron berikutnya.



Gambar 2.5 Fully-Connected Layer

Berikut adalah fungsi yang ada dalam neuron pada dense layer.

$$y = \sum_{i=0}^n w^i x^i + b$$

Dimana:

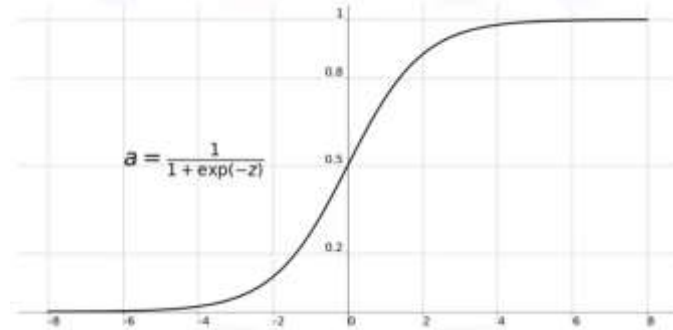
W = Weight

X = Input dari neuron sebelumnya

B = Bias

Terdapat beberapa jenis activation function pada FC layer, 3 di antaranya yang paling sering digunakan adalah *Sigmoid*, *Softmax*, dan *Relu*.

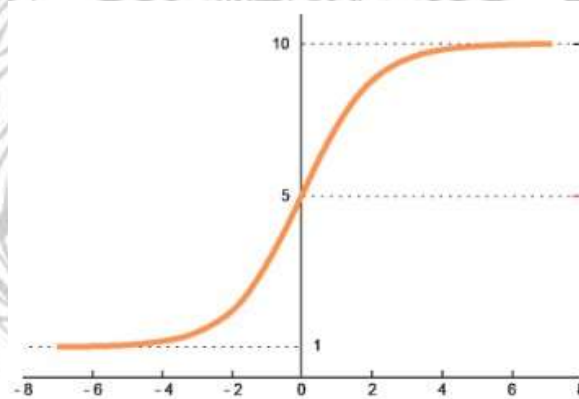
Sigmoid



Gambar 2.6 Sigmoid Function

Sigmoid berfungsi untuk mengubah logits menjadi 0 dan 1, Oleh karena itu dengan adanya sigmoid dapat dengan mudah untuk mengklasifikasikan dataset binary.

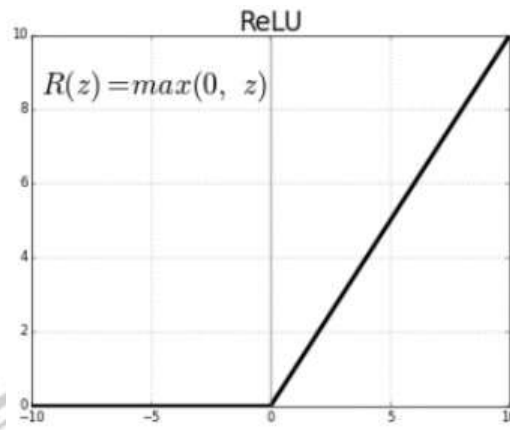
Softmax



Gambar 2.7 Softmax Function

Karena Softmax bertanggung jawab untuk mengubah nilai logit menjadi tiga output, Softmax adalah pilihan yang baik sebagai fungsi aktivasi untuk kumpulan data dengan tiga kategori atau lebih. Selain itu Softmax juga dapat menghasilkan keluaran berupa probabilitas yang jumlahnya selalu sama dengan 1.

Relu

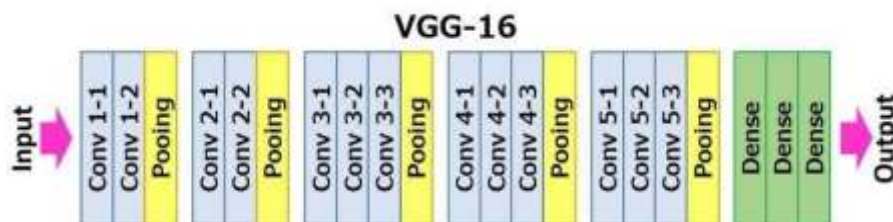


Gambar 2.8 Relu Function

Relu atau Rectified Linear Unit adalah fungsi aktivasi yang banyak digunakan dan populer. Relu biasanya diterapkan pada convolutional layer dan FC layer dalam model neural network. Relu juga dapat berfungsi untuk mengubah output negatif menjadi 0, Sehingga hanya output positif yang dihasilkan. Keuntungan menggunakan Relu adalah sering digunakan untuk melatih model AI karena performanya yang cepat dan mudah dipahami.

2.3 VGG16

VGG 16 adalah salah satu model dari VGGnet yang dibangun dengan menggunakan 16 lapisan sebagai arsitektur modelnya. Secara umum, VGG16 terdiri dari 5 blok konvolusional yang terkoneksi ke 3 MLP Classifiers. Pada lapisan output, VGG16 menggunakan fungsi aktivasi sigmoid jika terdapat 2 kategori atau kurang, dan fungsi aktivasi softmax jika terdapat 3 kategori atau lebih dalam dataset[8].

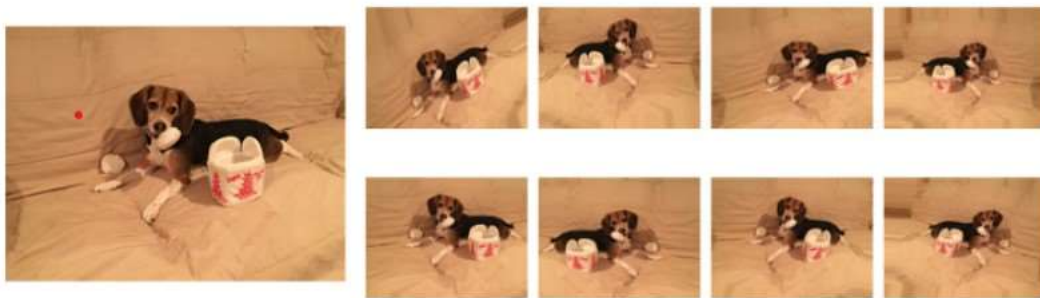


Gambar 2.9 Visualisasi Model VGG16

VGG16 meraih juara pertama dan kedua dalam ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014, dengan 1000 kategori dan 1000 poin data untuk setiap kategori.

2.4 Data Augmentation

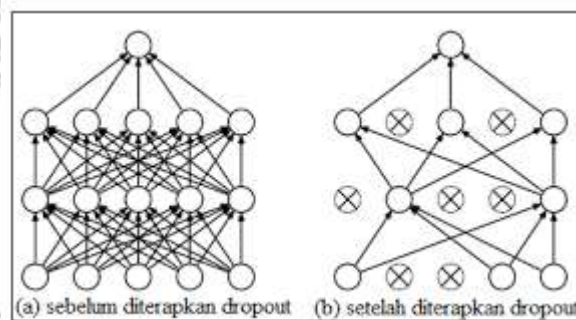
Augmentasi data adalah metode untuk meningkatkan ukuran kumpulan data. Untuk mencapai hasil yang optimal, pembelajaran mendalam membutuhkan kumpulan data yang besar. Untuk mencapai tujuan ini, augmentasi data sangat membantu karena meningkatkan ukuran kumpulan data dan mengurangi risiko overfitting karena keragaman yang diperoleh melalui teknik augmentasi data [11].



Gambar 2.10 Visualisasi Peningkatan Data

2.5 Dropout

Dropout merupakan teknik dimana sejumlah neuron tidak digunakan oleh teknik augmentasi data. secara acak pada tingkat probabilitas p . Tujuan menghilangkan neuron pada suatu lapisan adalah untuk memastikan bahwa neuron berikutnya tidak bergantung pada neuron tertentu. Hal ini membantu meminimalkan over-fitting.



Gambar 2.11 Visualisasi Layer Sebelum dan Sesudah Dropout

2.6 Tensorflow

Tensorflow adalah platform machine learning sumber terbuka yang dikembangkan oleh Google pada tahun 2015. Tensorflow menyediakan berbagai API berbeda dalam bahasa pemrograman berbeda yang menyederhanakan proses pembuatan model AI [12].



TensorFlow

Gambar 2.12 Logo Tensorflow

