

BAGIAN 2 ANALISIS DAN PERANCANGAN

Pre Processing

Preprocessing adalah tahap awal pengolahan citra asli sebelum citra tersebut diolah. Dalam tahap ini, umumnya preprocessing mengubah citra masukan menjadi citra abu-abu (grayscale) dan untuk memperbaiki kualitas citra. Selain mengubah citra masukan, preprocessing juga bertujuan untuk memanipulasi citra agar sesuai dengan hasil yang diinginkan.

1.1. Grayscale

Proses ini merupakan proses untuk merubah komposisi warna dari Citra menjadi Hitam Putih, Nilai intensitas paling tinggi pada putih ialah (255) dan Hitam memiliki nilai (0)

1.2. Resize Image

Proses ini merupakan proses untuk merubah ukuran Citra menjadi Citra baru dengan ukuran yang diinginkan, Proses resize sangat di butuhkan untuk mengurangi beban proses pada saat melakukan Train data Citra. Resize Citra yang akan digunakan merupakan fungsi Inter Area pada Open CV yang berfungsi saat akan merubah data Citra menjadi gambar yang lebih kecil.

1.3. Threshold

Penelitian ini menggunakan 2 jenis metode threshold yang akan digabungkan sehingga menghasilkan Citra yang lebih baik untuk di proses.

Binary Threshold

Binary Threshold memisahkan antara Object dengan Backgroundnya yang ditentukan berdasarkan parameter Threshold yang di inputkan, Sehingga jika pixel lebih besar dari threshold yang ditentukan maka bernilai 255 sisanya bernilai 0.

$$dst(x, y) = \begin{cases} maxval \\ 0 \end{cases}$$

$x, y =$ variable untuk representasi Citra

$maxval = 255$ (Batas nilai terbesar)

Jika data lebih dari nilai threshold yang ditentukan maka bernilai 255 jika tidak maka 0

Otsu Threshold

Otsu Threshold menggunakan nilai threshold secara otomatis sehingga dapat memisahkan antara background dan foreground. Tujuan metode ini ialah untuk mencari penyebaran nilai threshold minimum untuk background dan foreground.

Berikut formula untuk menentukan bobot kumulatif dari $\omega(k)$ dari Threshold dari 0 hingga 256

$$\omega(k) = \sum_{i=0}^k i$$

Selanjutnya formula untuk menghitung rerata kumulatif dari $\mu(k)$

$$\mu(k) = \sum_{i=0}^k i \cdot P_i$$

Terakhir, formula untuk penghitungan varians kumulatif dari

$$\sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_T)^2 P_i$$

Selanjutnya setelah mengimplementasikan metode segmentasi binary threshold dan otsu threshold maka citra akan berubah dan menghilangkan noisy yang tidak di perlukan.

1.4.Edge Detection

Canny merupakan salah satu Metode untuk mengimplementasikan deteksi tepi pada Citra, Metode Canny terdiri dari beberapa proses di dalamnya :

1. Penelitian ini menggunakan Fast Non Local Denoising yang dapat di gunakan untuk Citra yang telah di Grayscale maupun yang berwarna hal ini di butuhkan untuk mengurangi Noise pada Citra yang akan di proses.
2. Setelah melakukan denoising dengan menggunakan Non Local Denoising maka akan mendapatkan tepi sehingga gradient pada citra akan dihitung sebagai berikut :

$$|G| = |G_x| + |G_y|$$

1.5. Contours

Contours berfungsi untuk mendeteksi bentuk sebuah object, Contour merupakan sebuah kurva yang menghubungkan semua titik, Proses ini digunakan untuk melakukan segmentasi pada Citra.

Penelitian ini menggunakan beberapa fungsi pada Contour dan juga Metode pada Contour yaitu :

Contours berfungsi untuk mendeteksi bentuk sebuah object, Contour merupakan sebuah kurva yang menghubungkan semua titik, Proses ini digunakan untuk melakukan segmentasi pada Citra.

Penelitian ini menggunakan beberapa fungsi pada Contour dan juga Metode pada Contour yaitu :

1. Retrieval List

Contour memiliki hirarki, Hirarki pada contours dapat di artikan Parent sebagai outer dan Child sebagai Inner



ure 2 Hirarki Contour

Disini terlihat 0,1,2 merupakan hirarki yang sama yang dapat disitilahkan bahwa bentuk tersebut diluar dari bounding dibawahnya. Sedangkan 2, 2a merupakan Parent dan Child yang juga terdapat pada Hirarki yang sama hingga seterusnya.

Fungsi Retrieval List akan menghasilkan contour yang sama antara Parent dan Child seperti contoh 2 dan 2a, keduanya akan menjadi contour untuk area 2.

2. Chain Approx None

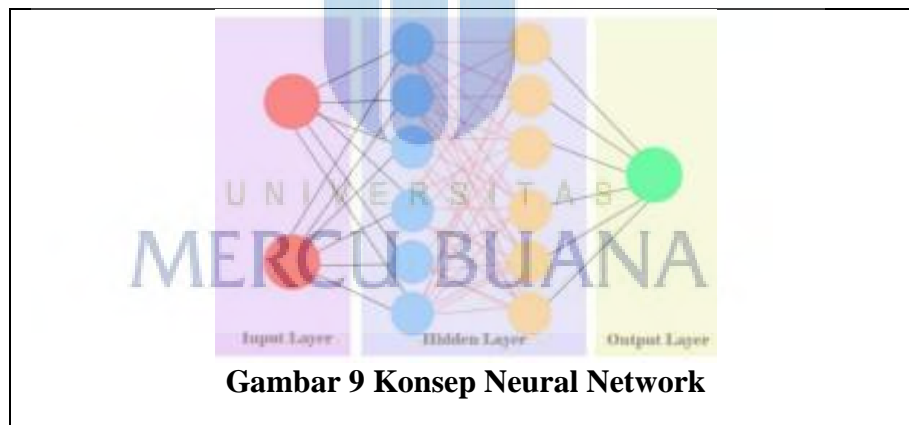
Fungsi ini berfungsi untuk menyimpan seluruh titik contour sehingga, setiap 2 titik berikutnya (x_1, y_1) dan (x_2, y_2) dari kontur akan menjadi *neighborhood* horizontal, vertikal atau diagonal, yaitu maks ($\text{abs}(x_1 - x_2)$, $\text{abs}(y_2 - y_1)$)

Model Convolutional Neural Network

Prinsip Neural Network telah lama dikenal dan di perkenalkan oleh Warren McCulloch dan Walter Pitts (1943) berdasarkan penelitiannya yang berjudul *A Logical Calculus Of The Ideas Immanent In Nervous Activity*, Seiring perkembangan jaman Neural Network semakin berkembang hingga menjadi Multi Layer Neural Network akan tetapi semakin banyaknya Hidden Layer dapat mengurangi akurasi yang semakin buruk.

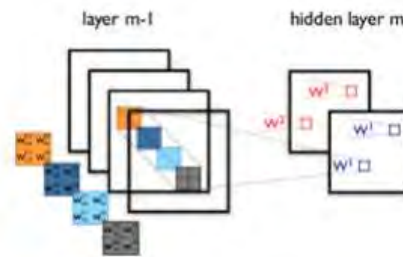
Penelitian terkait Neural Network terus berkembang hingga munculah Deep Belief Network (DBN) Hinton et al. 2006 membuat model Handwritten digit recognition yang paling akurat yang kini dikenal sebagai Istilah Deep Learning.

Convolutional Neural Network (CNN) merupakan pengembangan dari Multilayer Perceptron (MLP). CNN biasanya digunakan untuk mengklasifikasikan data Image 2 dimensi. CNN termasuk kedalam salah satu jenis Deep Learning. Deep Learning memiliki prinsip yang kurang lebih sama dengan Neural Network tetapi Deep Learning memiliki arsitektur yang cukup kompleks dan Hidden Layer yang sangat banyak.



Pada gambar diatas, setiap Input layer memiliki masing masing neuron yang akan terhubung dengan Hidden Layer, Input yang akan digunakan pada penelitian ini merupakan data pixel dari Image yang telah di olah sebelumnya, setiap koneksi antara layer memiliki bobot dan koneksi antara Input akan memiliki nilai Bobot dan Bias. Setiap neuron memiliki 2 operasi yaitu Operasi Linear z_j dan Non Linear/aktivasi

Pada gambar diatas, setiap Input layer memiliki masing masing neuron yang akan terhubung dengan Hidden Layer, Input yang akan digunakan pada penelitian ini merupakan data pixel dari Image yang telah di olah sebelumnya, setiap koneksi antara layer memiliki bobot dan koneksi antara Input akan memiliki nilai Bobot dan Bias. Setiap neuron memiliki 2 operasi yaitu Operasi Linear z_j dan Non Linear/aktivasi



Gambar 10 Konsep CNN

Forward Propagation

Konsep Feedforward merupakan komputasi neural network yang dimulai dari layer input, forward propagation melakukan perhitungan dengan menggunakan operasi Linier z_j dan dilanjutkan dengan melakukan aktivasi yang di simbolkan dengan h_j

$$z_j = \sum_{i=0}^d w_{ij} \cdot x_i$$

$$h_j = \sigma(z_j)$$

(5)

Fig. 124. (6)

Backpropagation

Konsep Backpropagation melakukan sebuah proses perhitungan selisih antara tiap weight dan bias berdasarkan error yang di dapat setelah tahapan feedforward. Proses ini berjalan mundur dimulai dari melakukan perhitungan dari output layer.

CNN Layers

Input Shape

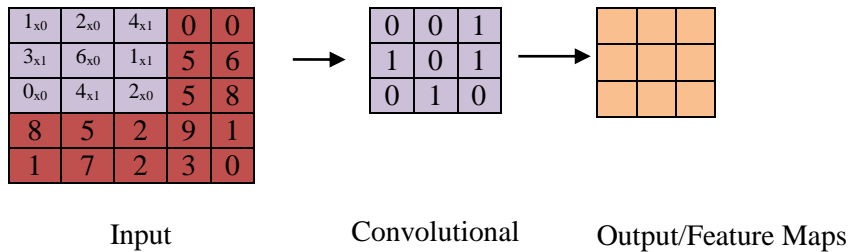
Input Shape merupakan ukuran data Input yang akan di Implementasikan pada model, Ukuran Input shape di sesuaikan dengan ukuran Citra, Citra yang digunakan pada penelitian ini memiliki ukuran 28x28.

Label merupakan data kategori dari aksara, pixel terdiri dari pixel 0 hingga pixel 783 yang merupakan data yang terpecah dari ukuran Citra 28x28.

Convolutional Layers

Convolutional Layer merupakan layer yang akan menghasilkan data ekstraksi dari matrik sebuah citra input, matrik yang terkandung pada citra tersebut (Pixel pada Citra) akan menjadi serangkaian nilai Vektor yang menjadi neuron input layer. Umumnya layer ini akan menghasilkan sebuah feature maps yang dapat bermacam macam dan terdiri dari corners, edges dan endpoints pada citra.

Convolutional Layer akan di jalankan berdasarkan ukuran filter/kernel dari input citra pada proses ini setiap matrik akan dikalikan dengan ukuran filter/kernel yang akan di gunakan. Setiap bagian pada matrix citra akan terus menerus dibagi dan dikalikan hingga menghasilkan sebuah feature maps.



Gambar 11 Proses Convolutional Layer

Activation

Fungsi Activation merupakan fungsi non linier, pada arsitektur CNN di penelitian ini activation function yang akan digunakan ialah ReLU, Fungsi ini akan menyatakan sebuah output 0 jika nilai inputnya adalah negative, jika positive maka nilai tersebut ialah nilai dari hasil aktivasi tersebut.

$$f(x) = \max(0, x) \quad (7)$$

Batch Normalization

Batch Normalization merupakan Teknik untuk meningkatkan kecepatan, kinerja dan stabilitas pada model CNN, Teknik ini akan menormalisasi input dan menskala ulang activation function. Teknik ini dapat mengurangi Overfitting pada model karena teknik ini menambahkan noise pada hidden layer sehingga mengurangi penggunaan *Dropout*.

Dropout

Dropout berfungsi untuk menghilangkan neuron neuron yang akan di hilangkan secara acak diberhentikan sementara dan hal tersebut bertujuan untuk mengurangi Overfitting dan memperbaiki loss function pada model.

Output Layers

Output Layer merupakan layer yang menampung seluruh hasil yang telah di kalkulasi berdasarkan perhitungan output, f merupakan fungsi activation yang akan di jumlahkan dengan nilai weight, input dan bias dari antara satu layer ke layer lanjutan

$$z_j = f \sum_{i=0}^d w_{ij} \cdot x_i + b_j \quad (8)$$

Selanjutnya hasil tersebut akan kembali di aktivasi menggunakan fungsi softmax untuk menghitung probabilitas dari setiap kelas, perhitungan tersebut ialah sebagai berikut :

(9)

$$p(x) = \frac{e^x}{\sum_{k=1}^K e^x}$$

Loss Function

Loss Function akan mengukur nilai loss dengan nilai yang di ekspetasikan dan hasil dari selisihnya akan menjadi tola ukur atas performa pada model. Penelitian ini menggunakan Categorical Crossentropy loss function dengan perhitungan sebagai berikut :

(10)

$$Loss = \sum_{i=1}^{Output\ Size} y_i \cdot \log \check{y}_i$$

CNN Arsitektur

Penelitian ini menggunakan GPU untuk melakukan learning model CNN dengan NVIDIA GEFORGE 930MX, Model CNN yang di implementasikan merupakan library dari Keras dan Tensorflow 2.0.

Table 5 Detail Arsitektur Model CNN

Layer	Size	Output Shape
Input	(28, 28, 1)	-
Conv2D + ReLU	32 (3 x 3)	(None, 26, 26, 32)
Batch Normalitzation	(3 x 3)	(None, 26, 26, 32)
Conv2D + ReLU	32 (3 x 3)	(None, 24, 24, 32)
Batch Normalitzation	(3 x 3)	(None, 24, 24, 32)
Conv2D + ReLU	32 (5 x 5)	(None, 12, 12, 32)
Batch Normalitzation	(5 x 5)	(None, 12, 12, 32)
Dropout	0.4	(None, 12, 12, 32)
Conv2D + ReLU	64 (3 x 3)	(None, 10, 10, 64)
Batch Normalitzation	(3 x 3)	(None, 10, 10, 64)

Conv2D + ReLU	64 (3 x 3)	(None, 8, 8, 64)
Batch Normalization	(3 x 3)	(None, 8, 8, 64)
Conv2D + ReLU	64 (5 x 5)	(None, 4, 4, 64)
Batch Normalization	(5 x 5)	(None, 4, 4, 64)
Dropout	0.4	(None, 4, 4, 64)
Conv2D + ReLU	128 (4 x 4)	(None, 1, 1, 128)
Batch Normalization	(4 x 4)	(None, 1, 1, 128)
Dropout	0.4	128
Dense	18	18

