

BAB II

LANDASAN TEORI

1.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan tinjauan Pustaka yang dapat mendukung penelitian, berikut ini merupakan tinjauan Pustaka yang diambil :

1. (Suartika, 2016) Ia melakukan penelitian berjudul "Klasifikasi gambar menggunakan jaringan saraf konvolusional (CNN) di Caltech 101." Dalam penelitiannya, ia membahas pembangunan model jaringan saraf tiruan konvolusional yang dapat mengklasifikasikan objek dalam gambar untuk menyederhanakan dan mempercepat proses klasifikasi gambar objek dan mengurangi tingkat kesalahan dalam klasifikasi yang menghasilkan hasil diperoleh lebih akurat daripada proses manual. Temuan Hal ini menghasilkan nilai presisi yang lebih baik, metode klasifikasi menggunakan CNN relatif reliabel dengan perubahan parameter itu sudah selesai. Waktu pelaksanaan yang diperlukan untuk sebuah gambar dapat ditentukan sebagai label kelas yang sesuai antara 3 hingga 5 jam.
2. (Zufar & Setiyono, 2016) Ia melakukan penelitian berjudul "Jaringan saraf konvolusional untuk pengenalan wajah secara real-time". Hasil penelitian ini menunjukkan bahwa pembuat model CNN sebagai pembangun jaringan mampu mengklasifikasikan citra wajah dengan rata-rata akurasi diatas 87%.
3. (Nurhikmat, 2018) Ia melakukan penelitian dengan judul "Implementasi deep learning untuk klasifikasi citra menggunakan algoritma

convolutional neural network (CNN) pada citra Wayang Golek”. Gangguan dari penelitian ini adalah model CNN pada penelitian ini menggunakan bentuk input 64x64, faktor pembelajaran 0,001, ukuran filter 3x3, 20 Epochs, 240 data latih, dan 60 data uji. Ini memberikan tingkat pelatihan dan pengujian akurasi dalam mengkategorikan citra Wayang. golek untuk 95% latihan dan 90% ujian.

4. (Dewi, 2018) Melakukan Penelitian yang berjudul “*Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network*”. Hasil dari pendeteksian klasifikasi meja dan kursi pada suatu citra digital menggunakan *Convolutional Neural Network* dapat dinilai bekerja dengan baik, Tingkat akurasi model yang didapatkan dari hasil pendeteksian klasifikasi citra meja dan kursi motif ukiran Jepara pada suatu citra digital menggunakan *Convolutional Neural Network* berkisar antara 70% hingga 99%.
5. (Pujoseno, 2018) Ia melakukan penelitian berjudul "Implementasi pembelajaran mendalam menggunakan jaringan saraf konvolusional untuk mengklasifikasikan alat tulis." Implementasi metode Convolutional Neural Network (CNN) untuk klasifikasi instrumen perekaman dilakukan dengan menggunakan paket Keras pada software RStudio versi 1.1.383. Jumlah layer weave yang digunakan sebanyak empat layer weave, tingkat akurasi data uji yang diperoleh dari model yang dibuat adalah 95% pada melakukan klasifikasi alat tulis..

Beberapa tinjauan pustaka diatas merupakan referensi bagi perkembangan penelitian tersebut. Berdasarkan hasil analisis dari berbagai penelitian yang telah

dilakukan sebelumnya, akan dilakukan penelitian tentang penerapan pengolahan citra hingga pendeteksian bola dengan metode Deep Learning yaitu Convolution Neural Network. Metode deep learning ini digunakan untuk menyelesaikan masalah deteksi bola karena dalam penggunaannya memerlukan proses komputasi yang relatif cepat, sehingga diharapkan penelitian ini dapat mendeteksi bola secara akurat.

1.2 Object Detection

Deteksi objek menentukan keberadaan objek dan ruang lingkungannya serta lokasi pada gambar. Ini bisa diambil sebagai pengantar objek kelas kedua, satu kelas mewakili kelas objek dan kelas lainnya mewakili kelas non-subjek. Deteksi objek dapat dibagi lagi menjadi lunak dan keras deteksi. Deteksi lembut hanya mendeteksi keberadaan objek selama deteksi sulit mendeteksi keberadaan objek dan lokasi objek (Jalied, 2016).

1.3 Citra (*Image*)

Citra (*image*) adalah citra pada bidang dua dimensi dan terdiri dari banyak pixel yang membentuk bagian terkecil dari gambar. Gambar keseluruhan dibentuk dari kotak biasa sehingga jarak horizontal dan vertikal antar pixel sama di seluruh gambar. Gambar sebagai keluaran sistem pencatatan data dapat berupa :

- a. Optik berupa foto,
- b. Analog berupa sinyal video seperti gambar pada monitor televisi,
- c. Digital yang dapat langsung disimpan pada media penyimpanan

Gambar tersebut terbagi menjadi dua bagian yaitu gambar diam. dan gambar bergerak. Gambar diam ditampilkan secara berurutan (berurutan), memberikan kesan gambar bergerak pada mata. Setiap foto dalam urutan tersebut disebut

bingkai. Gambar yang muncul dilayar lebar dalam sebuah film atau TV pada dasarnya terdiri dari ratusan hingga ribuan frame(Sitorus & Syahriol, 2006).

1.3.1 Citra Digital

Gambar digital adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi kontinu ke gambar diskrit dalam proses pengambilan sampel. Proses mengubah gambar menjadi gambar digital disebut digitalisasi. Mendigitalkan adalah proses mengubah gambar, teks, atau suara objek 15 dapat dilihat dalam data elektronik, dan disimpan serta diproses kebutuhan lainnya. Dalam konteks yang lebih luas, pengolahan citra digital lebih dari itu mengacu pada pemrosesan data dua dimensi apa pun. Ada pemrosesan gambar digital disiplin yang mempelajari teknik pemrosesan gambar. Gambar yang dimaksud di sini adalah gambar diam (foto) atau film (video). Sementara itu, digitalisasi memiliki tujuan penting di sini Pengolahan citra / image dilakukan secara digital dengan menggunakan komputer. Ada larik dalam gambar digital yang berisi nilai nyata dan kompleks yang diwakili oleh bit tertentu. (Kusumanto & Tomponu, 2011).

Di komputer, gambar digital dipetakan ke kisi dan elemen pixel dalam bentuk matriks dua dimensi. Masing-masing pixel ini memiliki ekstensi angka yang mewakili saluran warna. Sejumlah dicatat untuk setiap pixel secara berurutan oleh komputer dan sering kali dibatasi untuk digunakan kompresi atau pemrosesan tertentu. Sebuah gambar digital bisa mewakili matriks yang terdiri dari M kolom setelah baris N dengan persimpangan kolom dan baris disebut pixel. (pixel = elemen gambar) yang merupakan elemen terkecil gambar. Pixel memiliki dua parameter yaitu koordinat dan intensitas atau warna. Nilai dalam koordinat (x, y) adalah $f(x,$

y), yang besar intensitas atau warna pixel pada saat itu. Oleh karena itu, gambar dapat disimpan ke matriks :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Berdasarkan rumus diatas, suatu citra $f(x,y)$ dapat dituliskan kedalam fungsi matematis seperti berikut ini :

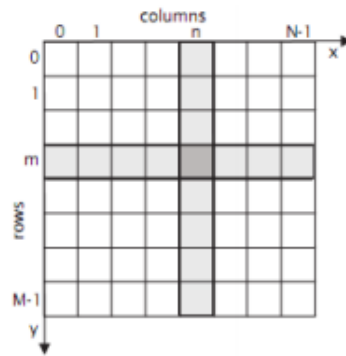
$$\begin{aligned} 0 \leq x \leq M-1 \\ 0 \leq y \leq N-1 \\ 0 \leq f(x,y) \leq G-1 \end{aligned}$$

Dimana :

M = jumlah pixel baris pada *array* citra
 N = jumlah pixel kolom pada *array* citra
 G = nilai skala keabuan (*grayscale*)

Besarnya nilai M, N, dan G biasanya merupakan perpangkatan dari dua seperti yang terlihat pada persamaan berikut : $M = 2^m ; N = 2^n ; G = 2^k$.

Dimana nilai *m*, *n*, dan *k* merupakan bilaingan positif. Interval (0,G) disebut dengan (*grayscale*). Besarnya nilai G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan)



Gambar 2.1 Representasi Citra Digital dalam 2 Dimensi

1.3.2 Pengolahan Citra

Pengolahan citra adalah proses pengolahan citra menggunakan komputer menjadi kualitas gambar yang lebih baik. Objektif Pemrosesan gambar ini dimaksudkan untuk meningkatkan kualitas gambar agar memungkinkan mudah diartikan oleh manusia atau mesin (komputer) (Riadi, 2016).

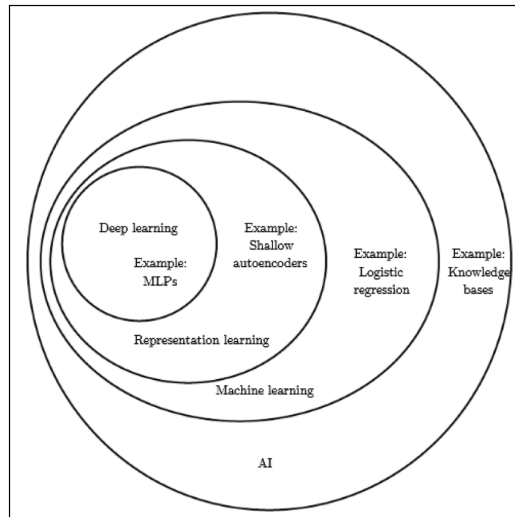
1.4 *Deep Learning*

Deep Learning adalah salah satu bidang pembelajaran mesin menggunakan jaringan saraf tiruan untuk memecahkan masalah kumpulan data besar. Teknik Deep Learning memberikan arsitektur yang sangat kuat untuk pembelajaran yang diawasi. Kemudian tambahkan lebih banyak lapisan Model pembelajaran ini dapat merepresentasikan data gambar yang diberi tag dengan lebih baik. Dalam pembelajaran mesin, ada teknik untuk menggunakan ekstraksi fungsi dari data pelatihan dan algoritma pembelajaran khusus untuk klasifikasi gambar dan mengenali suara. Namun metode ini masih memiliki sedikit kecepatan dan keakuratannya kurang.

Aplikasi konseptual yang dalam (multi-layer) dari jaringan saraf tiruan dapat dibuat menanggulangi algoritma pembelajaran mesin yang ada ke komputer

sekarang dapat belajar dengan kecepatan, keakuratan, dan skala. Aturan ini terus berkembang hingga Deep Learning semakin banyak digunakan di masyarakat penelitian dan industri untuk membantu memecahkan banyak masalah yang berkaitan dengan data besar seperti Visi komputer, pengenalan ucapan dan pemrosesan bahasa alami. Dijelaskan Teknik adalah salah satu fitur utama Deep Learning pola data yang berguna untuk membantu membedakan model kelas. Rekayasa Fitur juga merupakan teknik yang paling penting mencapai hasil yang baik dalam tugas prediksi. Namun, sulit untuk dipelajari dan menguasai karena mereka membutuhkan kumpulan data dan tipe data yang berbeda berbagai pendekatan teknik.

Algoritma yang digunakan dalam rekayasa fungsi dapat menemukan pola umum yang penting dalam membedakan antar kelas. Dalam pembelajaran yang mendalam, metode CNN atau Convolutional Neural Network sangat baik dalam menemukan fitur-fitur yang baik pada suatu citra hingga lapisan selanjutnya untuk membuat hipotesis nonlinier yang dapat meningkatkan kompleksitas model. Model yang kompleks tentunya akan membutuhkan waktu lama untuk dilatih, sehingga dalam dunia deep learning, penggunaan GPU sangatlah umum.(Danukusumo & Pudi, 2017)



Gambar 2.2 Diagram Venn Deep Learning

1.5 *Neural Network*

Menurut (Yunita, 2015) Neural Network adalah Jaringan saraf yang mensimulasikan jaringan saraf biologis manusia menjadi arsitektur komputer baru dan arsitektur algoritme untuk komputer konvensional. Ini memungkinkan Anda untuk menggunakan operasi komputasi yang sangat sederhana (penjumlahan, pengurangan, dan logika dasar) untuk menyelesaikan masalah yang kompleks, tidak jelas secara matematis, non-linier, atau stokastik..

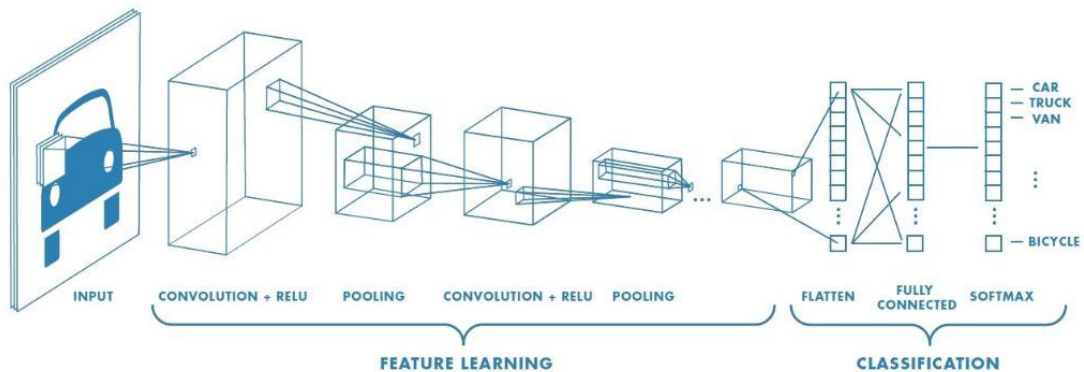
Ada beberapa karakteristik kemampuan otak manusia :

1. Mengingat
2. Menghitung
3. Menggeneralisasi
4. Adaptasi
5. Konsumsi energi yang rendah

NN berusaha meniru struktur atau arsitektur dan cara kerja otak manusia sehingga mampu menggantikan beberapa pekerjaan manusia.

1.6 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap pixel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik (P, et al., 2016). Berikut adalah jaringan arsitektur Convolutional Neural Network :



Gambar 2.3 Arsitektur Convolutional Neural Network

Berdasarkan gambar diatas, 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*.

1.6.1 Convolutional Layer

Convolutional layer adalah bagian dari langkah dalam arsitektur CNN. Pada tahap ini, operasi konvolusional dilakukan di pintu keluar lapisan sebelumnya. Lapisan ini adalah proses utama yang mendasari arsitektur CNN. Konvolusi adalah istilah matematika di mana satu fungsi diterapkan berulang kali ke hasil fungsi lainnya. Operasi konvolusi adalah operasi pada dua fungsi argumen bernilai nyata. Operasi ini menggunakan fungsi keluaran sebagai peta fungsi dari citra masukan. Input dan output ini dapat dilihat sebagai dua argumen nilai nyata. Operasi konvolusi dapat ditulis sebagai berikut :

$$s(t) = (x * t)(t) = \sum_{\alpha = -\infty}^{\infty} x(\alpha) * w(t - \alpha)$$

Keterangan :

S(t) = Fungsi hasil operasi konvolusi
X = Input
W = bobot (kernel)

Fungsi $s(t)$ memberikan *output* tunggal berupa *feature Map*. Argumen pertama adalah *input* yang merupakan x dan argumen kedua w sebagai kernel atau filter. Apabila dilihat *input* sebagai citra dua dimensi, maka bisa dikatakan t sebagai pixel dan menggantinya dengan i dan j . Maka dari itu, operasi untuk konvolusi ke *input* dengan lebih dari satu dimensi dapat menulis sebagai berikut :

$$s(i,j) = (K * I)(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} I(i - m, j - n) K(m, n)$$
$$s(i,j) = (K * I)(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} I(i + m, j + n) K(m, n)$$

Berdasarkan kedua persamaan diatas merupakan perhitungan dasar dalam operasi konvolusi, dengan i dan j adalah sebuah pixel dari citra. Perhitungan

tersebut bersifat komulatif dan muncul saat K sebagai kernel, kemudian I sebagai *input* dan kernel yang dapat dibalik relatif terhadap *input*. Sebagai alternatif operasi konvolusi dapat dilihat sebagai perkalian perkalian matriks antara citra masukan dan kernel dimana keluarannya dihitung dengan *dot product*. Selain itu, penentuan volume *output* juga dapat ditentukan dari masing-masing lapisan dengan *hyperparameters*. *Hyperparameter* yang digunakan pada persamaan di bawah ini digunakan untuk menghitung banyaknya neuron aktivasi dalam sekali *output*. Perhatikan persamaan berikut :

$$(W - F + 2P)/(S + 1)$$

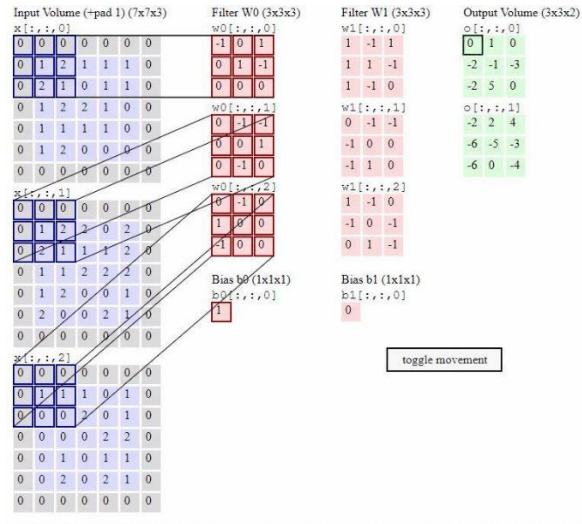
Keterangan :

- W = Ukuran volume gambar
- F = Ukuran Filter
- P = Nilai *Padding* yang digunakan
- S = Ukuran Pergeseran (*Stride*)

Berdasarkan persamaan di atas, dapat dihitung ukuran spasial dari volum *output* dimana *hyperparameter* yang dipakai adalah ukuran volume (W), *filter* (F), *Stride* yang diterapkan (S) dan jumlah *padding* nol yang digunakan (P). *Stride* merupakan nilai yang digunakan untuk menggeser *filter* melalui *input* citra dan 34 *Zero Padding* adalah nilai untuk mendapatkan angka nol di sekitar *border* citra. Berikut adalah operasi.

Convolutional Layer terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Sebagai contoh, layer pertama pada *feature extraction layer* biasanya adalah conv. Layers dengan ukuran 5x5x3. Panjang 5 *pixels*, tinggi 5 *pixels* dan tebal/jumlah 3 buah sesuai dengan *channel* dari *image* tersebut. Ketiga filter ini akan digeser keseluruhan bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara input

dan nilai dari filter tersebut sehingga menghasilkan sebuah output atau biasa disebut sebagai *activation map* atau *feature map*. Perhatikan ilustrasi berikut :

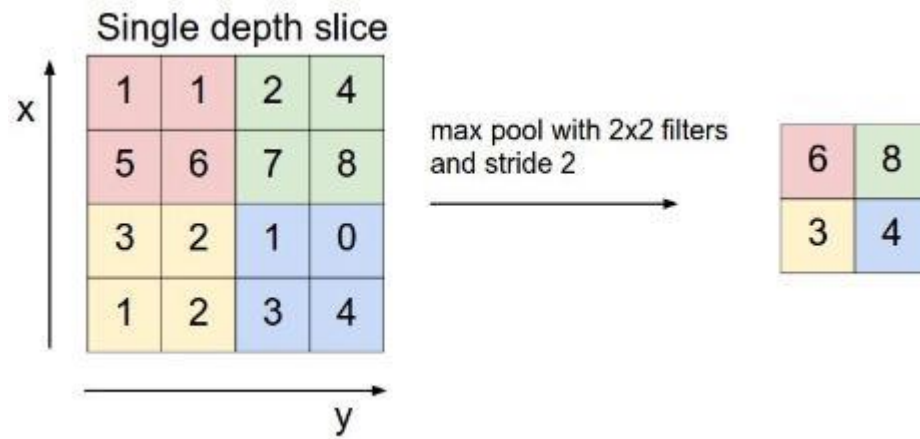


Gambar 2.4 Convolutional Layer

1.6.2 Pooling Layer

Pooling layer adalah merupakan pengurangan ukuran matriks menggunakan operasi gabungan. Lapisan pangkalan biasanya muncul setelah konversi. Pada dasarnya layer pool terdiri dari filter dengan ukuran tertentu dan step yang secara bergantian akan memindahkan seluruh area feature map. Ada dua jenis pooling yang umum digunakan pada pooling layer, yaitu pooling rata-rata dan maksimum. Nilai yang diambil untuk rata-rata gabungan adalah nilai rata-rata, sedangkan untuk gabungan maksimum adalah nilai maksimum. Menggabungkan lapisan antara lapisan konvolusional yang berurutan dalam arsitektur model CNN dapat secara bertahap mengurangi ukuran volume keluaran di peta fungsi, sehingga mengurangi jumlah parameter dan komputasi dalam jaringan untuk mengontrol overfitting. Lapisan kolam berjalan pada setiap tumpukan peta objek dan mengurangi ukurannya. Bentuk lapisan pooling biasanya menggunakan filter 2x2

yang diterapkan dalam dua langkah dan berfungsi pada setiap potongan masukan..
Berikut ini adalah contoh gambar operasi *max-pooling*



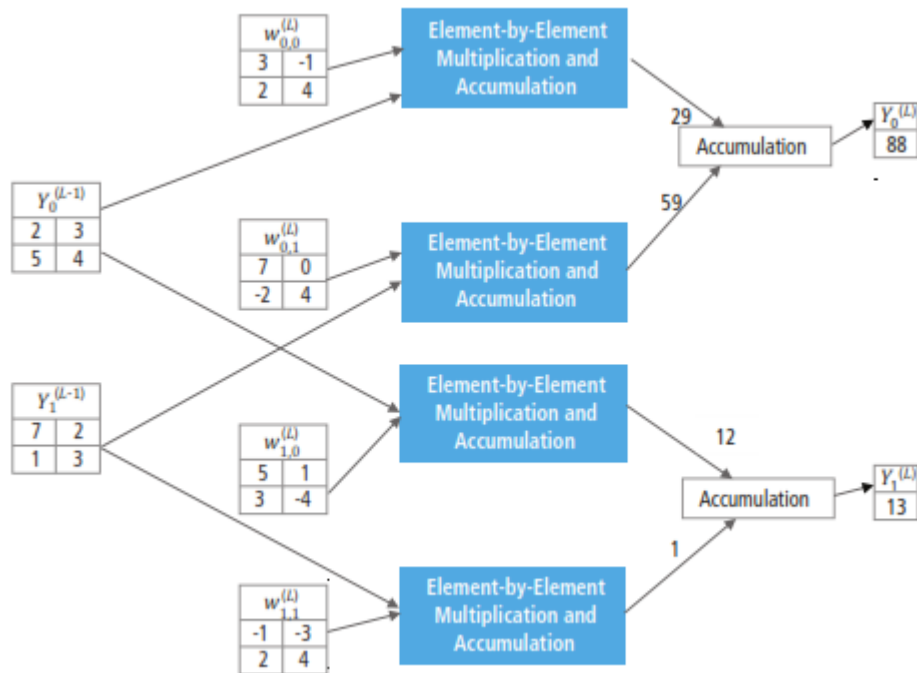
Gambar 2.5 Operasi Max-Pooling

Berdasarkan gambar diatas menunjukkan proses dari *max-pooling*. *Output* dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan pooling diatas akan beroperasi pada setiap irisan kedalaman volume input secara bergantian. Jika dilihat dari gambar diatas operasi *max-pooling* dengan menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4, dari masing-masing 4 angka pada input operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2.

1.6.3 Fully Connected Layer

Fully-Connected Layer Ini adalah lapisan di mana semua neuron yang diaktifkan menghindari lapisan sebelumnya, semuanya terhubung ke neuron di lapisan berikutnya, seperti jaringan saraf normal. Pada dasarnya layer ini biasanya digunakan pada MLP (Multy Layer Perceptron) yang bertujuan untuk mentransformasikan dimensi dari data tersebut sehingga data tersebut dapat diklasifikasikan secara linier. Perbedaan antara lapisan yang sepenuhnya terhubung

dan lapisan pleksus biasa adalah bahwa neuron di lapisan pleksus hanya terhubung ke area tertentu di pintu masuk, sedangkan Lapisan yang sepenuhnya terhubung memiliki neuron yang terhubung sepenuhnya. amunisi, kedua lapisan tersebut masih beroperasi pada perkalian titik, jadi fungsinya tidak jauh berbeda. Berikut ini adalah proses *fully-connected*



Gambar 2.6 Proses Fully Connected Layer

1.7 TensorFlow

TensorFlow adalah *framework machine learning* yang bekerja dalam skala besar dan dalam *environment* yang *heterogeneous*. TensorFlow digunakan untuk bereksperimen dengan model deep learning, melatih model pada set data besar, dan menyesuaikannya dengan produksi. Selain itu, TensorFlow juga mendukung pembelajaran skala besar dan inferensi menggunakan ratusan server yang

menggunakan unit pemrosesan grafis (GPU) untuk pelatihan yang efektif. Penelitian ini menggunakan TensorFlow sebagai struktur backend untuk Keras (Pangestu & Bunyamin, 2018).

1.8 *Google Coral Usb Accelerator*

Coral USB accelerator adalah perangkat USB yang menawarkan kemampuan inferensi Pembelajaran Machine yang kuat untuk sistem Linux. Dipasangkan dengan Edge TPU, ini adalah ASIC kecil yang dirancang oleh Google telah membuat perangkat ini mampu memberikan inferensi pembelajaran mesin dengan performa tinggi namun konsumsi daya tetap rendah melalui antarmuka USB 3.0. Misalnya, perangkat ini bisa bekerja Model visi seluler canggih seperti MobileNet V2 dengan kecepatan tinggi lebih dari 100 FPS. Alat ini memungkinkan Anda untuk menambahkan Inferensi pembelajaran mesin yang cepat pada perangkat dengan AI internal berenergi rendah. Model dikembangkan dengan TensorFlow Lite dan kemudian dikompilasi untuk berjalan pada Coral USB accelerator.

