BAB 2
LANDASAN TEORI

2.1 Aksara Jepang

Bahasa Jepang adalah suatu bahasa resmi yang digunakan oleh Negara Jepang dengan jumlah penutur 128 juta di seluruh dunia [9]. Dalam penulisan bahasa Jepang digunakan 3 jenis aksara yaitu *hiragana*, *katakana* dan *kanji*. *Hiragana* dan *katakana* merupakan simbol fonetik yang memiliki satu bunyi suku kata setiap aksaranya [10].

2.1.1 Hiragana

Hiragana merupakan satuan fonetik, yang memiliki satu bunyi suku kata setiap aksaranya [10]. Ciri khas dari *hiragana* adalah bentuk aksara yang banyak lekukannya [3]. Menurut Zakiyyah [3] *hiragana* memiliki beberapa fungsi, yaitu:

1. Untuk menuliskan kosakata asli bahasa Jepang.
2. Untuk menuliskan partikel yang di dalam kalimat.
3. Untuk menuliskan *furigana* (cara membaca kanji).
4. Untuk menuliskan *okurigana* (karakter untuk mengiringi kanji).
5. Untuk menuliskan honorifik (ungkapan penghormatan kepada orang).

Huruf *hiragana* dasar memiliki 46 huruf. Aksara *hiragana* dijelaskan pada Gambar 2.1.

UNIVERSITAS MULTIMEDIA NUSANTARA

Gambar 2.1. Hiragana dasar

sumber: [11]

Selain dari 46 hiragana dasar, terdapat 2 tanda baca tambahan yang bernama dakuten dan handakuten untuk melengkapi keseluruhan bunyi yang ada pada bahasa Jepang.

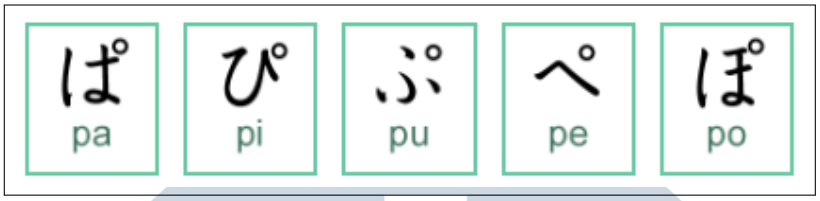
A Dakuon

Dakuon merupakan huruf hiragana yang memiliki tambahan dakuten di bagian kanan atas huruf. Dakuten merupakan dua buah garis yang berbentuk seperti 2 titik atau tanda petik dua (゛) [12]. Dakuten digunakan pada konsonan k, s, t, dan h untuk membentuk konsonan g, z, d, dan b. Semua huruf *hiragana* dakuon memiliki 20 huruf. Aksara *hiragana* dakuon dijelaskan pada Gambar 2.2.

Gambar 2.2. Hiragana dakuon
sumber: [11]

B Handakuon

Handakuon merupakan huruf hiragana yang memiliki tambahan handakuten pada bagian kanan atas huruf. Handakuten merupakan sebuah tanda baca tambahan yang berbentuk seperti lingkaran kecil (゜) [12]. Handakuten hanya digunakan pada konsonan h untuk membentuk konsonan p. Semua huruf *hiragana* handakuon memiliki 5 huruf. Aksara *hiragana* handakuon dijelaskan pada Gambar 2.3.



Gambar 2.3. Hiragana handakuon
sumber: [11]

2.2 Machine Learning

Machine learning merupakan suatu mesin yang telah dikembangkan untuk dapat belajar sendiri tanpa adanya arahan dari usernya [13]. *Machine learning* merupakan algoritma yang mempelajari sendiri berdasarkan data yang telah disediakan. Semakin banyak data dan sampel untuk *training*, maka algoritma akan semakin beradaptasi untuk meningkatkan kinerjanya. Dalam *machine learning*, data disebut juga sebagai *training set*, ataupun dataset. Tahapan proses *machine learning* dibagi menjadi 5 bagian [14], yaitu

1. Data collection

   Pengumpulan data dilaksanakan pada berbagai macam sumber. Jenis data dapat berbagai macam format, seperti csv, json, dan lain-lain.

2. Data wrangling and cleaning

   Data yang telah di kumpul akan dicek kembali, data yang tidak diperlukan akan di hapus / tidak digunakan. Selain itu data juga perlu di koreksi, dan menangani data jika ada data yang hilang.

3. Model building

   Ada 2 cara untuk model building, yaitu proses *training* dan testing. *Training* dilakukan untuk mesin dapat mengenali pola dari data. Testing dilakukan untuk melihat seberapa baik mesin dapat mengenali / memprediksi jawaban berdasarkan trainingnya.

4. Model evaluation

   Setelah model selesai dibuat maka perlu mengevaluasi model, contohnya seperti melihat akurasinya, atau menggunakan *confusion matrix*.

5. Model deployment

   Model deployment merupakan penerapan model yang telah dibuat ke suatu tempat lain, seperti dalam aplikasi mobile, atau website.

Dalam machine learning, umumnya menggunakan 2 jenis skenario pembelajaran yaitu supervised learning dan unsupervised learning.

2.2.1 Supervised learning

Supervised learning merupakan suatu pendekatan dalam pembuatan *machine learning*. *Supervised* sendiri berarti diawasi atau dipimpin. *Supervised learning* berarti suatu algoritma *machine learning* yang proses pembelajarannya diawasi oleh suatu supervisor atau guru. *Supervised learning* ini memerlukan data yang telah berlabel untuk membangun suatu model.

2.2.2 Unsupervised learning

Unsupervised learning merupakan suatu proses pembelajaran yang bebas karena tidak ada pengawasan. Algoritma *unsupervised learning* akan lebih bebas saat proses pembelajarannya karena data tidak memiliki label.

2.3 Artificial Neural Network

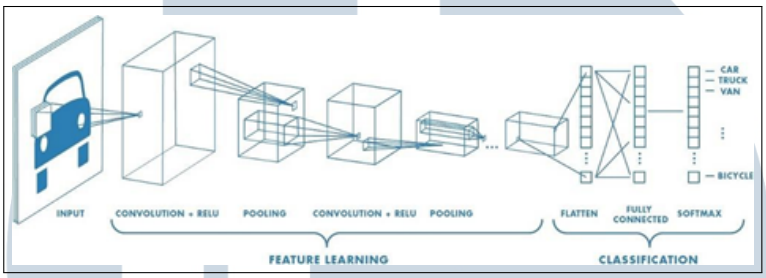
Artificial Neural Network atau sering disebut jaringan syaraf tiruan adalah sebuah jaringan dari sekelompok unit proses kecil yang dibuat berdasarkan perilaku jaringan saraf manusia. *Artificial Neural Network* terdiri dari sejumlah besar node komputasi yang saling berhubungan (disebut sebagai neuron), yang bekerja terjalin secara terdistribusi untuk secara kolektif learning dari input untuk mengoptimalkan output akhirnya [15].

2.4 Convolutional Neural Network

Convolutional Neural Network merupakan pengembangan dari *Artificial Neural Network*. *Convolutional Neural Network* adalah salah satu algoritma dari *Deep Learning* yang menerima input dengan bentuk gambar, menentukan objek dalam gambar yang bisa digunakan oleh mesin untuk dapat belajar mengenali suatu gambar, dan dapat membedakan suatu gambar dengan gambar yang lainnya.

Kelebihan dari *Convolutional Neural Network* dibandingkan oleh ANN yaitu memiliki kemampuan untuk menghasilkan fitur yang lebih relevan sesuai dengan input gambar yang telah diberikan [16].

Arsitektur dari Convolutional Neural Network dilihat pada Gambar 2.4.



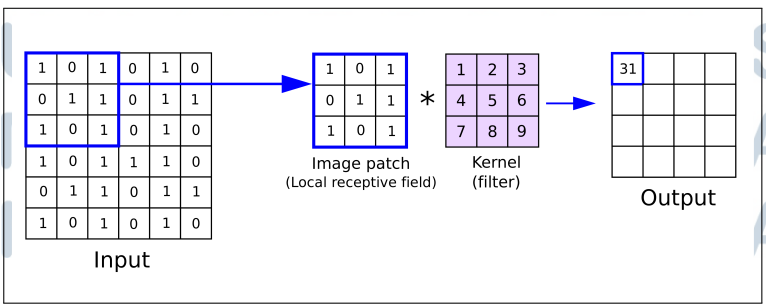
Gambar 2.4. Arsitektur CNN
sumber: [17]

2.4.1 Input Layer

Input layer merupakan layer yang berfungsi untuk menerima input citra. Data yang telah di input akan ditentukan nilai tinggi, lebar, dan jumlah channelnya. Biasanya jumlah channel adalah 3, untuk menampilkan nilai RGB setiap pikselnya. Citra yang diterima di input layer ini akan dilanjutkan ke layer berikutnya.

2.4.2 Convolutional Layer

Tahap ini merupakan inti dari CNN. Pada bagian ini, terjadi konvolusi antara citra yang telah di input dengan sebuah filter. Setiap gambar yang ada, di konversikan dalam bentuk matriks, lalu dilakukan perkalian matriks pada setiap kombinasinya. Proses ini dapat dilihat pada gambar 2.5.



Gambar 2.5. Convolutional layer
sumber: [18]

Convolutional layer memiliki beberapa komponen utama [19],

1. Filter

   Filter merupakan suatu parameter untuk *convolutional layer* dapat mengonfigurasi kumpulan filter lapisan. Filter adalah fungsi yang memiliki tinggi dan lebar lebih kecil dari tinggi dan lebar inputan. Filter akan diterapkan melintasi tinggi dan lebar suatu volume input dengan cara sliding window. Filter juga akan diterapkan untuk setiap kedalaman volume input.

2. Activation maps

   Activation adalah hasil numerik jika suatu neuron memutuskan untuk membiarkan informasi lewat. *Activation maps* merupakan suatu pemetaan / mapping yang sesuai dengan activation pada berbagai bagian gambar. Filter akan melintasi dimensi spasial (lebar, tinggi) volume input selama penerusan informasi melalui CNN. Hal ini menghasilkan output 2 dimensi yang disebut *activation maps* untuk filter tertentu.

3. Parameter sharing

   Parameter sharing digunakan untuk mengontrol jumlah dari parameter total. Hal ini membuat waktu training lebih sedikit, karena lebih sedikit sumber daya untuk mempelajari dataset training.

4. Layer-specific hyperparameters

   Berikut ini merupakan *hyperparameter* yang menentukan pengaturan spasial dan ukuran volume output, yaitu

   (a) Filter size

   Setiap filter memiliki daerah spasial sesuai dengan lebar dan tinggi ukuran filter.

   (b) Output depth

   Output depth akan mengontrol jumlah neuran di convolutional layer yang terkoneksi ke wilayah yang sama dari input volume.

   (c) Stride

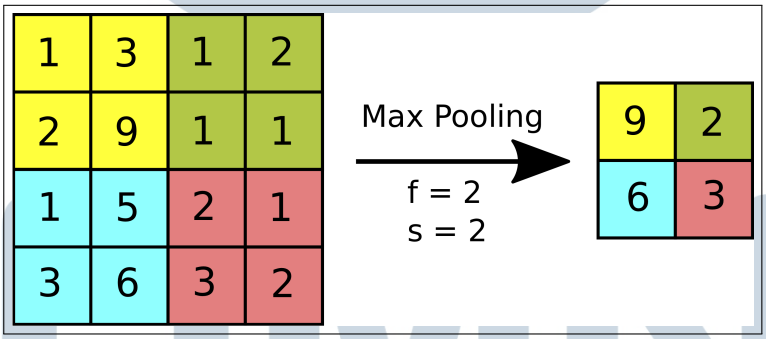
   Stride akan mengontrol seberapa jauh filter window digeser setiap pengaplikasian fungsi filter.

(d) Zero padding

Zero-padding akan mengontrol ukuran spasial volume output. Hal ini dilaksanakan saat ingin mempertahankan ukuran spasial dari volume input maupun output.

2.4.3 Pooling Layer

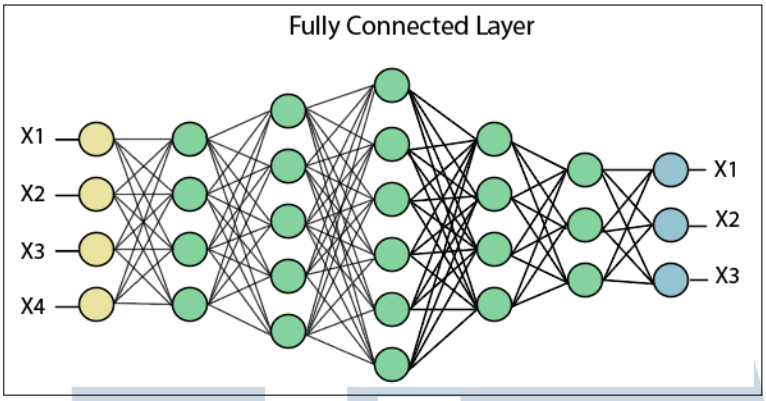
Tahap ini melakukan ekstraksi hal-hal penting dalam suatu citra. Fungsi dari *pooling layer* ini adalah untuk mereduksi input / mengurangi jumlah parameter dengan metode *down-sampling* [20]. Metode dalam *pooling* ada 2 jenis, yaitu *max pooling* dan *average pooling*. Metode dalam *pooling layer* yang sering digunakan adalah *max pooling*. Metode *max pooling* merupakan suatu metode yang mengambil nilai tertinggi dari sebuah matrix yang dipilih / dicakup oleh suatu filter. *Average pooling* merupakan suatu metode yang mengambil nilai rata-rata dari sebuah matrix yang dipilih / dicakup oleh suatu filter. Gambar 2.6 merupakan contoh gambaran cara kerja *Max pooling layer*.



Gambar 2.6. Max pooling layer
sumber: [18]

2.4.4 Fully Connected Layer

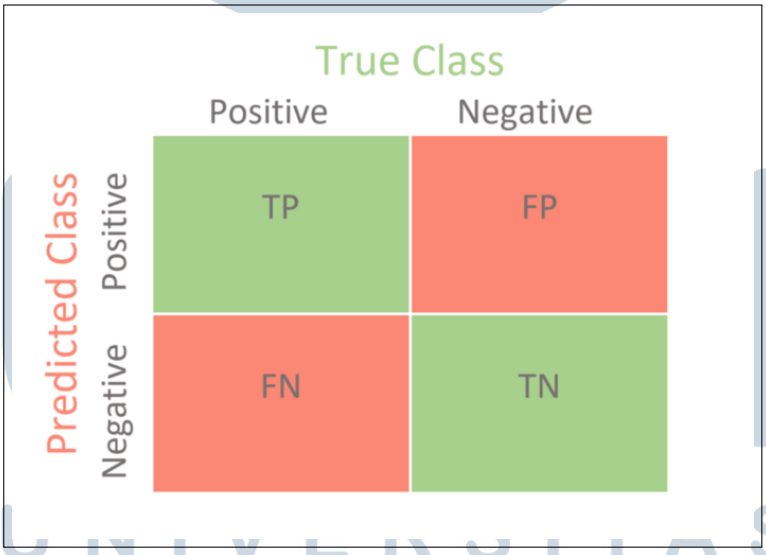
Fully connected layer merupakan layer akhir dari CNN. Layer ini memiliki hubungan / koneksi antara semua *neuron* pada layer sebelumnya [19]. Setelah mengumpulkan semua hasil nilai dari layer-layer sebelumnya, layer ini akan melakukan komputasi dan menghasilkan nilai setiap kelasnya. Berdasarkan nilai yang telah didapatkan layer ini akan mengklasifikasikan citra yang telah di input ke beberapa kelas yang ditentukan. Gambar 2.7 merupakan gambaran dari *fully connected layer*.



Gambar 2.7. Fully connected layer
sumber: [21]

2.5 Confusion Matrix

Confusion matrix yaitu suatu cara tabular yang digunakan untuk melakukan evaluasi dan visualisasi dari model yang ada [22] [23]. Gambar 2.8 merupakan gambaran dari confusion matrix.



Gambar 2.8. Confusion matrix
sumber: [22]

Berdasarkan Gambar 2.8 terdapat 4 istilah hasil klasifikasi pada confusion matrix [22] [23], yaitu:

1. TP, merupakan singkatan dari true positif. True positif merupakan jumlah prediksi dari kelas positif dan hasil prediksi benar.

2. FP, merupakan singkatan dari false positif. False positif merupakan jumlah prediksi dari kelas positif dan hasil prediksi salah.

3. FN, merupakan singkatan dari false negatif. False negatif merupakan jumlah prediksi dari kelas negatif dan hasil prediksi salah.

4. TN merupakan singkatan dari true negatif. True negatif merupakan jumlah prediksi dari kelas negatif dan hasil prediksi benar.

Hasil data dari confusion matrix dapat digunakan untuk menghitung akurasi, precision, recall, f1-score dari suatu model.

1. Akurasi

   Akurasi mengambarkan seberapa akurat model secara keseluruhan, yang berarti fraksi dari total sampel yang diklasifikasikan dengan benar oleh pengklasifikasi [22]. Nilai akurasi dapat dihitung dengan menggunakan rumus berikut:

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2. Precision

   Precision merupakan rasio dari kelas positif yang benar-benar positif [22]. Nilai precision dapat dihitung dengan menggunakan rumus berikut:

$$precision = \frac{TP}{TP + FP} \tag{2.2}$$

3. Recall

   Recall merupakan rasio dari semua sampel positif yang diprediksi dengan benar sebagai positif [22]. Nilai recall dapat dihitung dengan menggunakan rumus berikut:

$$recall = \frac{TP}{TP + FN} \tag{2.3}$$

4. F1-score

F1-score merupakan penggabungan dari presisi dan recall, yaitu rata-rata harmonik dari precision dan recall [23]. Nilai f1-score dapat dihitung dengan menggunakan rumus berikut:

$$f1-score = 2 * \frac{precision * recall}{precision + recall} \tag{2.4}$$

F1-score dapat digabungkan dari setiap kelas untuk memiliki satu nilai untuk keseluruhan model. Beberapa cara untuk menggabungkan yaitu:

(a) Micro F1

Micro averaged f1 score dihitung dengan mempertimbangkan total TP, total FP dan total FN model [22]. Micro averaged tidak mempertimbangkan setiap kelas secara individual, melainkan menghitung metrik secara global.

(b) Macro F1

Macro averaged f1-score dihitung menggunakan rata-rata aritmatika dari semua skor f1 per kelas [22].

(c) Weighted F1

Weighted f1-score dihitung dengan mengambil rata-rata dari skor f1 per kelas sambil mepertimbangkan support dari masing-masing kelas [22].

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA