

BAB 2

LANDASAN TEORI

Terdapat beberapa teori yang digunakan dalam penelitian, yaitu Bahasa Korea, *Deep Learning*, Convolutional Neural Network, Fungsi Aktivasi, Fungsi Loss, *Vanishing Gradient*, dan *Confusion Matrix*.

2.1 Bahasa Korea

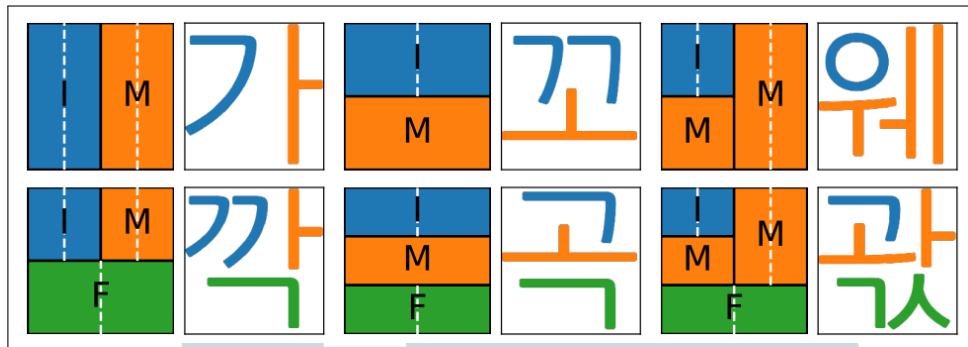
Orang Korea, baik di Korea Selatan maupun Korea Utara, menggunakan bahasa Korea sebagai bahasa utama. Alfabet Korea disebut 'Hangeul' atau 'Hangul' [3]. Sebelum Hangul digunakan, bahasa Cina klasik digunakan sebagai bahasa tertulis resmi di Korea. Pada abad ke-15 sekitar tahun 1443, Raja Sejong menciptakan Hangeul. Sistem penulisan Korea adalah satu-satunya tata cara penulisan yang sengaja dirancang untuk mempermudah pembacaan bagi masyarakat umum [27].

Suku kata dalam bahasa Korea terdiri dari kombinasi tiga jenis huruf, yaitu Cho-sung (konsonan awal), Jung-sung (vokal), dan Jong-sung (konsonan akhir). Terdapat 19 Cho-sung, 21 Jung-sung, dan 28 Jong-sung, sementara Jong-sung menggunakan semua 19 huruf Cho-sung [16]. Gambar 2.1 berikut menunjukkan konsonan awal, vokal, dan konsonan akhir dalam Bahasa Korea.

Initial	ㄱ ㅋ ㆁ ㄷ ㄸ ㄴ ㄹ ㄺ ㄻ ㄼ ㄽ ㆁ ㄷ ㅌ ㅍ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ
Medial	ㅏ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ
Final	ㄱ ㅋ ㆁ ㄷ ㄸ ㄴ ㄹ ㄺ ㄻ ㄼ ㄽ ㆁ ㄷ ㅌ ㅍ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ

Gambar 2.1. Konsonan Awal, Vokal, dan Konsonan Akhir Bahasa Korea [28]

Sistem penulisan Hangul Korea memisahkan kata-kata ke dalam blok suku kata yang diatur secara alfabetis-silabis, dan di antara blok suku kata terdapat jarak fisik yang memisahkannya [29]. Suku kata tersebut terdiri dari 2-4 huruf vokal dan konsonan dengan struktur konsonan-vokal (CV), konsonan-vokal-konsonan (CVC), atau konsonan-vokal-konsonan-konsonan (CVCC). Kombinasi struktur vokal dan konsonan Huruf Hangul ditunjukkan pada Gambar 2.2.



Gambar 2.2. Kombinasi Struktur Vokal/Konsonan [28]

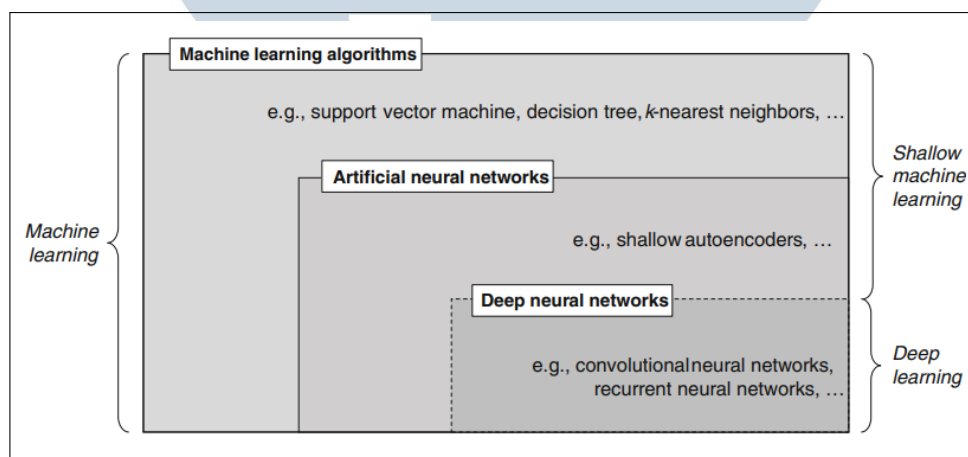
2.2 Handwritten Character Recognition

Handwritten Character Recognition (HCR) adalah kemampuan *software* atau perangkat untuk mengenali dan menganalisis tulisan tangan manusia dalam berbagai bahasa. HCR digunakan secara luas dalam berbagai aplikasi seperti membaca alamat pos, terjemahan bahasa, formulir bank dan jumlah cek, perpustakaan digital, deteksi kata kunci, dan deteksi tanda-tanda lalu lintas. Proses utama dalam sistem HCR meliputi akuisisi gambar, praproses, segmentasi, ekstraksi fitur, dan klasifikasi. Akuisisi gambar melibatkan mendapatkan gambar karakter tulisan tangan, yang kemudian diproses untuk menghilangkan distorsi dan dikonversi menjadi gambar biner. Pada tahap praproses, setiap karakter dibagi menjadi sub-gambar untuk analisis lebih lanjut. Ekstraksi fitur berfokus pada ekstraksi ciri-ciri khas dari gambar karakter tersebut. Akhirnya, metode klasifikasi seperti Convolutional Neural Network (CNN), Support Vector Machines (SVM), Recurrent Neural Networks (RNN), dan lainnya digunakan untuk mengklasifikasikan dan mengenali karakter tulisan tangan dengan akurasi tinggi. HCR memiliki peran penting dalam mendigitalkan dan menginterpretasikan informasi tulisan tangan, memungkinkan otomatisasi dan efisiensi dalam berbagai bidang [19].

2.3 Deep Learning

Deep learning adalah konsep *machine learning* yang menggunakan *artificial neural network* untuk membangun model analitis secara otomatis. *Machine learning* merupakan konsep yang menggambarkan kemampuan sistem untuk belajar dari data pelatihan yang spesifik guna otomatisasi pembangunan model

analitis dan penyelesaian tugas terkait. *Machine learning* telah menghasilkan berbagai kemajuan yang luar biasa dalam algoritma pembelajaran yang canggih dan teknik praproses yang efisien. Salah satu kemajuan tersebut adalah evolusi Artificial Neural Network (ANN) menjadi arsitektur jaringan saraf yang semakin dalam dengan kemampuan pembelajaran yang lebih baik yang dikenal sebagai *deep learning*. Model *deep learning* memiliki kinerja yang lebih baik daripada model *machine learning* dangkal dan pendekatan analisis data tradisional. Hal ini memungkinkan sistem cerdas untuk belajar secara mandiri dari data pelatihan yang spesifik dan mengatasi tugas yang kompleks. *Deep learning* memiliki manfaat dalam domain dengan data yang besar dan berdimensi tinggi, sehingga *neural network* dalam *deep learning* mengungguli algoritma *machine learning* dangkal dalam sebagian besar aplikasi yang membutuhkan pemrosesan data teks, gambar, video, ucapan, dan audio [30]. Gambar 2.3 menunjukkan gambaran konsep dari machine learning.

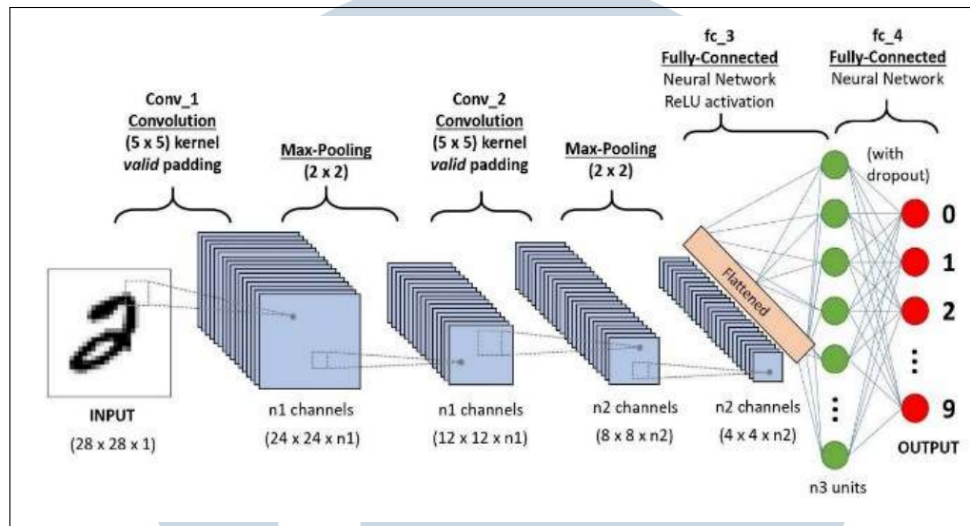


Gambar 2.3. Konsep *Machine Learning* [30]

2.4 Convolutional Neural Network

CNN merupakan jaringan saraf yang menggunakan ekstraktor fitur, terdiri dari *convolution layer* dan *down-sampling layer*. Setiap neuron pada lapisan tersebut hanya terhubung dengan bagian neuron pada *upper layer* yang disebut sebagai bidang reseptif lokal. *Convolution Layer* pada umumnya terdiri dari beberapa *feature map*, dimana setiap *feature map* memiliki sejumlah neuron, dan bobot neuron dibagi di antara *feature map* yang sama untuk mengurangi koneksi antara lapisan jaringan dan mencegah *overfitting* [22]. Arsitektur CNN

memiliki terbagi menjadi tiga bagian fundamental, yaitu *convolution*, *pooling*, *fully-connected neural network*, ditunjukkan pada Gambar 2.4.

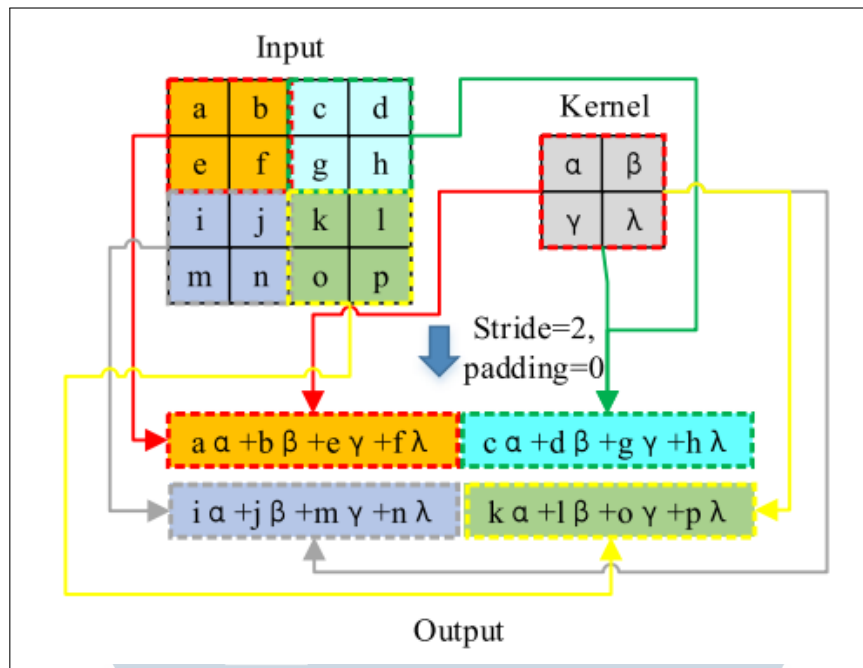


Gambar 2.4. Arsitektur Convolutional Neural Network [23]

2.4.1 Convolution

Convolution adalah proses matematis yang banyak digunakan dalam pemrosesan gambar, seperti menghaluskan, meningkatkan fitur, dan mempertajam gambar. Ini melibatkan penggunaan matriks angka, atau filter, untuk mengubah gambar. Dalam *convolution*, setiap nilai *pixel* pada gambar dikalikan dengan nilai *pixel* yang sesuai pada filter, kemudian dijumlahkan untuk menghasilkan nilai *pixel* baru pada gambar yang difilter. Proses ini digunakan untuk menangkap fitur dasar pada gambar seperti batas, warna, dan gradien. Pada dasarnya, *convolution* membantu mengubah gambar dengan cara yang mirip dengan cara manusia memahami gambar. Proses ini dapat digunakan untuk mengurangi dimensi fitur pada gambar dan menciptakan pemahaman yang lebih baik terhadap gambar [23]. Ilustrasi proses *convolution* ditunjukkan pada Gambar 2.5.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 2.5. Ilustrasi Proses *Convolutional* [22]

Operasi *convolution* dapat dinyatakan sebagai berikut.

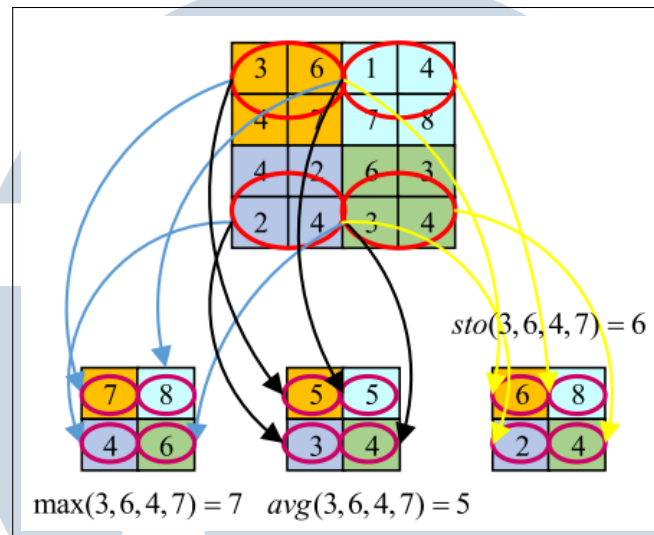
$$f_l^k(p, q) = \sum_c \sum_{x, y} i_c(x, y) \cdot e_l^k(u, v)$$

Dimana $i_c(x, y)$ adalah elemen dari tensor gambar masukan I_C , yang dikalikan secara elemen dengan indeks $e_{k,l}(u, v)$ dari kernel konvolusi k_l dari *layer* ke-1. Sementara itu, *feature map* keluaran dari operasi konvolusi ke- k dapat dinyatakan sebagai $F_l^k = [f_l^k(1, 1), \dots, f_l^k(p, q), \dots, f_l^k(P, Q)]$ [20].

2.4.2 Pooling

Pooling bertujuan untuk mengurangi kompleksitas komputasi di dalam jaringan dengan meningkatkan skala spasial deskripsi gambar secara perlahan-lahan. Terdapat beberapa jenis *pooling* yang dapat digunakan, termasuk *maximum pool*, *minimum pool*, *average pool*, dan *adaptive pool*. *Max-pooling* adalah jenis yang paling umum digunakan dan membantu menjaga fitur yang dominan dengan mempertahankan rotasi dan translasi yang sama dalam pelatihan model yang efisien. Sementara itu, *average pooling* dapat memberikan hasil yang lebih baik dalam mengurangi *noise* di gambar. Keduanya dapat membantu dalam pengurangan

dimensi gambar dan mengoptimalkan kinerja CNN [23]. Ilustrasi proses *pooling* ditunjukkan pada Gambar 2.6.



Gambar 2.6. Ilustrasi Proses *Pooling*
[22]

Operasi *pooling* dapat dinyatakan sebagai berikut.

$$Z_l^k = g_p(F_l^k)$$

Dimana Z_l^k mewakili *feature map* ter-*pooling* dari lapisan ke-1 untuk peta *feature map* ke-k dengan input F_l^k , sedangkan $g_p(\cdot)$ mendefinisikan jenis operasi *pooling* yang digunakan [20].

2.4.3 Fully-Connected Neural Network

Fully-Connected Neural Network menghitung label yang paling cocok untuk mewakili gambar dengan memproses hasil akhir dari operasi *convolution/pooling*. Jaringan ini menggunakan bobot yang dihitung selama pelatihan untuk membuat koneksi antara vektor fitur gambar dan kelas gambar. Dalam operasi ini, *output* dari *convolution/pooling* dikalikan dengan bobot yang sesuai dan kemudian diolah melalui fungsi aktivasi [23].

2.5 Fungsi Aktivasi

Fungsi aktivasi menghitung *input* neuron melalui pemetaan fungsional dari \mathbb{R}^d ke \mathbb{R} , dan menghitung *output* neuron melalui fungsi aktivasi yang merupakan pemetaan fungsional dari \mathbb{R} ke \mathbb{R} . Meskipun ini adalah standar dalam jaringan saraf buatan, ada juga model jaringan saraf yang mengimplementasikan pendekatan yang berbeda. Terdapat juga kelas khusus yang dapat dilatih yang memiliki definisi neuron non standar. Klasifikasi utama didasarkan pada kemampuan modifikasi bentuk fungsi aktivasi selama pelatihan. Klasifikasi utama dalam fungsi aktivasi adalah antara fungsi dengan bentuk tetap (*fixed-shape*) dan fungsi dengan bentuk yang dapat diubah. Fungsi aktivasi dengan bentuk tetap mencakup fungsi-fungsi klasik yang sering digunakan dalam literatur *neural network*, seperti sigmoid, tanh, dan ReLU. Namun, dengan diperkenalkannya fungsi-fungsi terkait *rectifier* (seperti ReLU), kategori ini dapat dibagi lebih lanjut menjadi dua.

1. Fungsi berbasis *rectifier*: termasuk semua fungsi yang terkait dengan keluarga *rectifier*, seperti ReLU, LReLU, dan sebagainya.
2. Fungsi aktivasi klasik: mencakup semua fungsi yang tidak termasuk dalam keluarga *rectifier*, seperti sigmoid, tanh, dan step function [31].

2.6 Fungsi Loss

Fungsi *loss* digunakan untuk mengukur perbedaan antara hasil *output* model dan data sampel yang sebenarnya. Tujuannya adalah membimbing model agar mencapai konvergensi selama proses pelatihan. Dengan meminimalkan nilai *loss*, model dapat disesuaikan dengan data pelatihan dan mengurangi kesalahan saat menguji model. Hal ini memungkinkan model untuk mengklasifikasikan sampel baru dengan tingkat akurasi yang tinggi. SoftMax *loss* merupakan fungsi *loss* yang sangat penting dalam klasifikasi gambar. Fungsi ini mudah dioptimalkan dan memberikan kontraksi cepat. Dalam praktiknya, SoftMax *loss* sering digunakan bersama dengan cross-entropy *loss* untuk memastikan klasifikasi yang akurat dari kategori yang sudah diketahui sebelumnya [32].

2.7 Vanishing Gradient

Salah satu masalah yang umum terjadi dalam *training deep neural network* adalah *vanishing gradient*, yang secara tradisional disebut sebagai pengurangan

panjang gradien. Hal ini disebabkan oleh fungsi aktivasi yang menjenuh dan nilai tunggal yang kecil dari matriks jacobian. Masalah ini menjadi salah satu faktor utama yang menghambat pelatihan jaringan yang sangat dalam pada tahap pengembangan awal jaringan saraf tiruan. Pentingnya mempertahankan intensitas gradien selama *training* dalam penyelesaian masalah *vanishing gradient*. Pilihan fungsi *loss* dapat mempengaruhi intensitas gradien yang terbentuk [33].

2.8 Confusion Matrix

Confusion matrix adalah metode yang umum digunakan untuk menilai kinerja dari algoritma klasifikasi [34]. *Confusion matrix* adalah cara standar untuk menunjukkan jumlah *True Positive* (TP), *False Positive* (FP), *True Negatif* (TN), dan *False Negatif* (FN) agar lebih visual. *True Positive* adalah elemen yang diberi label positif oleh model dan memang benar-benar positif. *False Positive* adalah elemen yang diberi label positif oleh model, tetapi sebenarnya negatif. *True Negative* adalah elemen yang diberi label negatif oleh model dan memang benar-benar negatif. *False Negative* adalah elemen yang diberi label negatif oleh model, tetapi sebenarnya positif. Tabel 4 merupakan tabel *confusion matrix*.

Tabel 2.1. Tabel *Confusion Matrix*

	Classifier says YES	Classifier says NO
In reality YES	True Ppositives	False Negatives
In reality NO	False Positives	True negatives

Sumber: [34]

Confusion matrix memungkinkan untuk menentukan nilai dari *accuracy*, *precision*, *recall*, dan *F1 Score*. *Accuracy* adalah probabilitas bahwa prediksi model itu benar. Dengan kata lain, *accuracy* mengukur sejauh mana model dapat memprediksi dengan benar *class* dari unit yang dipilih secara acak.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision adalah ukuran seberapa akurat model dalam memprediksi unit sebagai positif berdasarkan proporsi unit yang model nyatakan sebagai positif dan benar-benar positif.

$$Precision = \frac{TP}{TP + FP}$$

Recall mengukur akurasi prediksi model terhadap kelas positif dengan cara menilai kemampuan model dalam menemukan semua unit yang memiliki label positif dalam *dataset*.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score merupakan metrik yang mengevaluasi kinerja model klasifikasi dengan menggunakan *confusion matriks*, yang menggabungkan metrik *precision* dan *recall* dalam konsep rata-rata harmonik [35].

$$F1Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

