

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Menurut hasil riset yang dilakukan oleh R Jannah, M Walid dan H Hoiriyah pada tahun 2022 dengan judul “ Sitem Pengenalan Citra Dokumen Tanda Tangan Menggunakan Metode *CNN (Convolutional Neural Network)* ”. Pada riset ini terkait bagaimana membangun sistem untuk mengidentifikasi tanda tangan secara akurat, yang disebabkan oleh tingginya kasus pemalsuan tanda tangan dan untuk membantu menjaga keamanan dokumen – dokumen legal menggunakan metode *CNN*. Untuk studi ini citra tanda tangan yang digunakan sebanyak 250 jenis citra, kemudian jenis citra tanda tangan diatas dibagi menjadi 2 folder yaitu data latih (*train*) dan data pengujian (*testing*). Pada studi ini, inputan yang akan digunakan berupa citra tanda tangan yang kemudian dilakukan *scanner*. Setelah itu, langkah selanjutnya adalah membagi citra yang telah dikumpulkan menjadi 2 folder yakni data latih (*train*) dan data uji (*test*). Pada pembagian data citranya ini yaitu 200 citra untuk data *train* dengan latar belakang (*background*) putih dan menggunakan tinta hitam serta bertujuan untuk melatih algoritma atau memberikan informasi tentang tanda tangan yang cocok dengan sistem [4].

Pada penelitian yang dilakukan oleh U. Mawwadah dan H. Armanto pada tahun 2021 dengan judul “ Prediksi Karakteristik Personal Menggunakan Analisis Tanda Tangan Dengan Menggunakan Metode *Convolutional Neural Network (CNN)*. Dalam penelitian ini memiliki tujuan yakni dapat melakukan prediksi karakteristik citra tanda tangan personal dengan menggunakan metode *Convolutional Neural Network* dengan dilatar belakangi sering terjadi kesalahan identifikasi tanda tangan calon karyawan suatu perusahaan terkhusus bagian *HRD (Human Resourche Development)* yang membuat perusahaan rugi baik sisi waktu ataupun biaya yang dikeluarkan oleh perusahaan tersebut. Pada studi ini menggunakan 500 citra tanda tangan untuk datasetnya, dimana datasetnya diperoleh menggunakan alat pemindai (*scanner*) dan *cropping image* . Penelitian ini

juga menggunakan aplikasi software berupa google image search dan juga google image processing yang berfungsi untuk menghasilkan berbagai jenis fitur tanda tangan yang bermacam – macam dengan resolusi yang berbeda. Dengan demikian, 500 tanda tangan frame diekstraksi dan disimpan dalam format *Portable Network Graphic* (PNG) yang memiliki resolusi piksel 2480x3407. Kemudian gambar tanda tangan dibagi menjadi folder yang berbeda menurut kategori tanda tangan. Pada penelitian ini menggunakan *Google Colab* (Colab) karena dianggap baik dalam mendeteksi atau klasifikasi objek dengan menggunakan bahasa pemrograman *Python*. Model tersebut menggunakan *neural network* dimana untuk softwrenya menggunakan *network mobile v2*, dimana fungsi yang pertamanya mendefinisikan *base-model* yang akan digunakan dalam project *include shape* yang didefinisikan dimensi dari gambar yang akan digunakan, dan *Include Top* berfungsi berupa perintah yang menyertakan model yang digunakan akan disertakan pada top layer dan arsitektur *network* [5].

Sesuai hasil yang dipublikasikan ditahun 2023 dengan judul “Klasifikasi Kepemilikan Tanda Tangan Menggunakan *Convolutional Neural Network (CNN)* Dengan Arsitektur *Alexnet*” yang ditulis oleh Krisnaldi Wijaya dan Eka Puji Widiyanto. Topik penelitian ini adalah mengklasifikasi tanda tangan menggunakan metode *cnn* dengan arsitektur *alexnet* yang dilatar belakangi mencegah terjadinya pelanggaran yang merugikan pihak tertentu dengan contoh sederhana seperti mahasiswa seringkali menipiskan absen kepada teman sekelasnya untuk tidak hadir saat pelajaran berlangsung dengan hal ini diperlukan membantu pengajar untuk memvalidasi tanda tangan tersebut. Pada penelitian ini menggunakan *dataset* sebanyak 300 tanda tangan dan 10 tanda tangan eksternal dimana melibatkan 10 responden. Data yang diperoleh untuk studi ini didapatkan dengan proses menggunakan aplikasi *Cam Scanner*. Perbandingan *dataset* untuk penelitian ini yaitu 80% data latih (*train*) dan 20% data uji (*test*) sehingga data yang terbagi menjadi 240 data *training* dan 60 data *testing*. Model yang digunakan pada studi menggunakan arsitektur *Alexnet* dengan *Optimizer Adam, SGD, dan RMSprop* yang dapat mengklasifikasikan dengan baik dan akurat. Untuk penelitian ini hasil akurasi tertinggi didapatkan oleh *Optimizer SGD* dengan nilai *precision* sebesar 90%, *recall* 83.3%, *f1 – score* 80.7%, dan *accuracy* 83.3% [6].

2.2 DASAR TEORI

2.2.1 Kecerdasan Buatan (*Artificial Intellegency*)

Kecerdasan buatan, juga dikenal sebagai *AI* adalah teknologi dibidang ilmu komputer yang memasukan kecerdasan manusia ke dalam mesin (komputer) untuk menyelesaikan berbagai tugas dan masalah dengan cara yang sama atau bahkan lebih baik dari manusia. Kecerdasan buatan bisa dikatakan teknologi berbasis sistem komputer yang memungkinkan memiliki kemampuan untuk melakukan kegiatan manusia yang membutuhkan intelegensi (Healey, 2020). Kegiatan manusia yang dapat dilakukan oleh AI berupa hal-hal yang terkait dengan fungsi kognitif (Simplilearn, 2020). AI terkesan mampu melakukan apapun dan dimanapun, namun AI hanya terkonsentrasi untuk implementasi pragmatis dalam proses intelegensinya (Ertel, 2018). Pada dasarnya, kecerdasan buatan juga dikenal sebagai *artificial intelegency (AI)*, adalah jenis pengetahuan yang memberi komputer kemampuan untuk melakukan tugas – tugas yang dilakukan oleh manusia yang membutuhkan keterampilan manusia [7].

Fungsi *Artificial Intelligence (AI)* untuk mempercepat dan meringankan suatu pekerjaan. Menurut beberapa ilmuwan yang kredibel, termasuk John McCharty-1956, Kecerdasan buatan (*AI*) adalah upaya untuk meniru perilaku manusia dan proses berpikir manusia. Lalu, pendapat dari ilmuwan Herbet Simon-1987, *artificial intelligence* adalah suatu topik yang berkaitan dengan pemrograman komputer dalam melakukan tugas yang dianggap cerdas oleh manusia.

Tujuan dari diciptakanya *artificial intelligence* sebagai berikut ini :

1. Membuat mesin lebih pintar.
2. Memahami kecerdasan.
3. Membuat mesin lebih berguna.
4. Mengurangi waktu kerja sehingga kegiatan bisa lebih efisien.
5. Membuat manusia lebih mudah melakukan tugas dan membuat keputusan dengan kecerdasan buatan.

Artificial Intelligence juga mempunyai kelebihan diantaranya sebagai berikut :

1. Bersifat Permanen.

2. Mudah diduplikasi.
3. Bersifat konsisten.
4. Dapat didokumentasikan.
5. Dapat menyimpan berbagai informasi dan data tanpa adanya batasan.

Sedangkan untuk *Artificial Intelligence* memiliki kekurangan diantaranya sebagai berikut :

1. Rawan terjadi kerusakan
2. Dapat menggantikan manusia
3. Proses pembuatan yang cukup mahal.

2.2.2 Deep Learning

2.2.2.1 Konsep Dasar Deep Learning

Deep Learning atau *deep structured learning/hierarchical learning* merupakan struktur sebuah kecerdasan buatan, dan juga *mechanice learning* adalah bagian pengembangan dari *neural network multiple layer* untuk memberikan ketepatan tugas. Menurut pendapat ilmuwan yang kredibel LeGin, Bengio, & Hinton (LeCun, Bengio, & Hinton, 2015) mendefinisikan *deep learning* adalah metode pembelajaran representasi dengan berbagai tingkat representasi, yang didapatkan melalui menyusun modul dengan sederhana, dimana *nonlinier* yang masing-masing mengubah representasi pada satu tingkat (diawali dari *input mentah*) menjadi representasi pada tingkat yang lebih tinggi dan lebih *abstrak*. Sedangkan menurut pendapat ilmuwan yang diringkas oleh Chollet Francois (Francois, 2018) bahwa *deep learning* sebuah pendekatan baru dalam mempelajari representasi dari data yang menekankan pada pembelajaran lapisan-lapisan representasi yang semakin lama semakin meningkat. Oleh karena itu secara sederhana, *deep learning* adalah pendekatan pembelajaran data yang bertujuan menciptakan representasi data berjenjang (abstraksi) dengan menggunakan berbagai lapisan pengolahan data. Menariknya, representasi data tersebut tidak dibuat langsung oleh manusia, melainkan dihasilkan melalui proses algoritma pembelajaran (LeCun, Bengion, & Hinto 2015) [8].

Beberapa perbedaan mendasar antara *deep learning* dan *mechanice learning* adalah sebagai berikut ini :

1. *Simplicity*

Simplicity memiliki keunggulan yang terletak pada fakta bahwa data dapat digunakan sebagai *input* dalam proses pembelajaran yang tidak memerlukan rekayasa fitur sebelumnya. Dengan demikian, proses pembelajaran model *deep learning* lebih mudah daripada model pembelajaran mesin. Dengan kata lain, pembelajaran model *deep learning* bersifat *end-to-end*, yang membuatnya lebih mudah dilakukan.

2. *Scalability*

Dengan skalabilitas, proses pembelajaran model *deep learning* dapat diimplementasikan secara paralel dengan memanfaatkan kekuatan unit pengolahan grafik (*Graphic Processing Units*) atau Unit Pengolahan Tensor (*Tensor Processing Units*). Selain itu, algoritma pembelajaran yang bersifat iteratif yang menggunakan banyak batch memungkinkan model pembelajaran dari dataset yang sangat besar.

3. *Versatility (adaptability)*

Salah satu keunggulan model *deep learning* adalah kemampuan beradaptasi (*adaptability*). Tidak selalu ada kebutuhan yang memulai proses pembelajaran model tersebut dari nol. Sebaliknya, model *deep learning* yang sudah terlatih dapat dilatih kembali dengan data pelatihan baru. Oleh karena itu, model ini memiliki kemampuan untuk terus belajar secara online atau berkelanjutan.

4. *Reusability*

Dalam transfer learning, sebuah model yang dilatih menggunakan dataset besar dapat digunakan untuk tugas lain dengan dataset yang lebih kecil. Keunggulan ini disebut reusabilitas dan mengurangi waktu yang dibutuhkan untuk pelatihan ulang.

Pada metode *deep learning* menggunakan sejumlah hyperparameter pelatihan untuk mengatur parameter [9]. Penemuan ini didasarkan pada penemuan Lu Yifan pada tahun 2017. Beberapa parameter termasuk :

a. *Hidden layer*

Dalam *deep learning hidden layer* berfungsi untuk menentukan jumlah lapisan tersembunyi dan jumlah *neuron* di setiap lapisan dalam arsitektur pembelajaran mendalam.

b. *Epochs*

Setiap *epoch* menunjukkan bahwa algoritma deep learning telah menyelesaikan satu siklus pembelajaran dari seluruh set data pelatihan, yang berarti bahwa epoch merupakan representasi jumlah iterasi yang diperlukan untuk melatih algoritma deep learning pada seluruh dataset pelatihan.

c. *Learning rate*

Salah satu faktor penting dalam proses pelatihan yang berfungsi untuk mengatur seberapa besar bobot akan diperbarui selama proses pelatihan disebut dengan *learning rate*. Nilai *learning rate* berkisar antara nol (0) hingga satu (1). Semakin tinggi nilai *learning rate*, pelatihan atau *training* akan berlangsung cepat. Namun, nilai *learning rate* yang tinggi juga berarti jaringan akan kehilangan tingkat ketelitian atau akurasi. Sebaliknya, jika nilai *learning rate* rendah, maka tingkat ketelitian jaringan model akan meningkat, tetapi akan mengakibatkan proses pelatihan memakan waktu lebih lama.

d. *Evaluate Performance*

Evaluate Performance atau kinerja model dievaluasi melalui pengaturan parameter *epoch* dan *learning rate*. Hal ini dilakukan untuk menemukan hasil model dengan kinerja terbaik berdasarkan tingkat akurasi, *TPR*, dan *ROC* (kurva karakteristik operasi penerima).

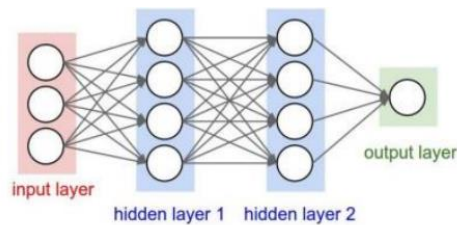
Dalam *Deep Learning* terdapat beberapa jenis algoritmanya sebagai berikut :

1. *Convolutional Neural Network (CNN)*
2. *Recurrent Neural Network (RNN)*
3. *Long Short Term Memory Network (LSTM)*
4. *Self Organizing Maps (SOM)*

2.2.3 Convolutional Neural Network (CNN)

Salah satu jenis *neural network feed forward* (tidak berulang) adalah *Convolutional Neural Network*, yang merupakan evolusi dari *Multilayer*

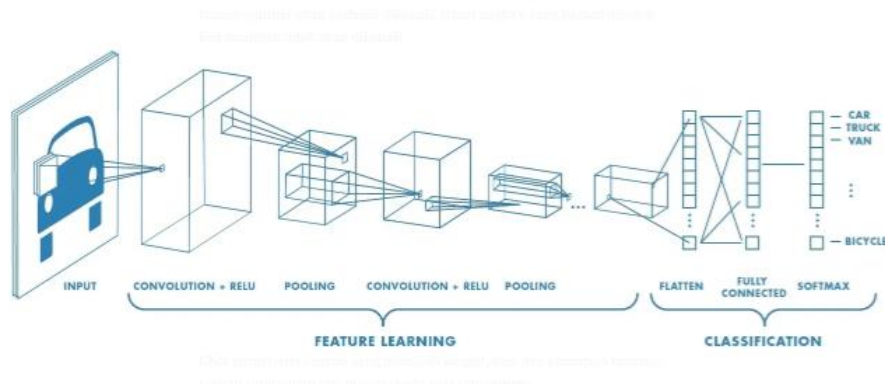
Perceptron (MLP). Karena memiliki banyak lapisan (kedalaman) dan sering digunakan untuk mengolah data citra, dimana dibuat khusus untuk memproses data dua dimensi dan termasuk dalam jenis *Deep Neural Networks*. Untuk menggambarkan jaringan, *CNN* menggunakan analisis gambar visual, deteksi, dan pengenalan objek dalam gambar karena terdiri dari *neuron* yang memiliki fungsi aktivasi, bias, dan bobot dimana *CNN* tidak terlalu berbeda dari *neural network* konvensional [10]. *CNN* dan *MLP* memiliki kesamaan dalam cara kerjanya, tetapi *CNN* setiap *neuron* dalam bentuk dua dimensi sedangkan *MLP* dimana setiap *neuron* hanya berukuran satu dimensi. Gambar 2.1 menunjukkan prinsip kerja arsitektur *MLP*.



Gambar 2.1 Arsitektur *MLP*

Seperti yang ditunjuk pada gambar 2.1, sebuah *MLP* terdiri dari berbagai lapisan atau *layer* i, dimana yang ditunjuk pada kotak merah dan biru, dan setiap lapisan mengandung *neuron* yang ditunjukkan dengan lingkaran putih. Setelah menerima *input* data satu dimensi, *MLP* mengirimkannya ke jaringan untuk menghasilkan *output*. Kualitas mode ditentukan oleh parameter bobot satu dimensi yang dimiliki oleh setiap hubungan *neuron* pada dua lapisan terpisah. Untuk setiap data yang dimasukkan ke dalam lapisan, operasi linier dengan bobotnya dilakukan. Selanjutnya operasi non – linier yang dikenal dengan fungsi aktivasi digunakan untuk mengubah hasil komputasi. Karena data jaringan dua dimensi, operasi linier dan parameter bobot *CNN* berbeda. Sehingga dengan demikian, data *CNN* berbentuk empat dimensi, dimana terdiri dari kumpulan kernel konvolusi, dan operasi linier menggunakan operasi konvolusi. Ini ditunjukkan pada gambar 2.1 [11].

Pada gambar 2.2 akan menjelaskan komponen pada proses *cnn* dimana komponen terbagi dibagian *feature laerning* dan *calssification* [12].



Gambar 2.2 Komponen pada tahapan *CNN*

Pada gambar 2.2 merupakan proses *CNN* dimana terdapat beberapa komponen utama yang sangat penting pada prosesnya [13]. Komponen diantaranya sebagai berikut :

a. *Convolution Layer*

Bagian utama *convolutional neural network* adalah blok *convolutional layer*, yang merupakan operasi matematika untuk menggabungkan dua set data. Untuk membuat *feature map*, *convolutional* diterapkan pada data *input* menggunakan filter *convolutional*.

b. *Pooling Layer*

Max dan *Average Pooling* adalah dua jenis *pooling layer*. *Average pooling* akan menghitung nilai rata – rata dari setiap *feature map* pada setiap pergesseran *window pooling*, dan *Max pooling* akan meringkas data masukan atau *feature map* dengan menghitung nilai atau nilai terbesar dari *feature map* berdasarkan pergesseran *window pooling*. Mengurangi jumlah parameter yang ada pada tensor *input* adalah tujuan utama dari lapisan penyimpanan. Ini dapat membantu mengurangi *overfitting*, mengekstrak fitur representatif dari tensor *input*, mengurangi perhitungan, dan secara keseluruhan meningkatkan efisiensi.

c. *Fully Connected Layer Output*

Setelah fitur diekstraksi dan membentuk *array* multidimensi, *feature map* perlu diubah menjadi vektor agar dapat dijadikan *input* bagi *fully connected layer*. *Fully connected layer* merupakan lapisan dimana setiap *neuron* pada lapisan sebelumnya terhubung dengan semua *neuron* pada lapisan berikutnya, seperti jaringan syaraf tiruan. Aktivitas dari lapisan sebelumnya harus diubah menjadi data satu arah sebelum terhubung dengan semua *neuron* pada lapisan berikutnya.

Lapisan *fully connected* ini sering digunakan dalam pendekatan *MLP* (*Multilayer Perceptron*) untuk memproses data sehingga dapat dikategorikan.

d. *Dropout*

Dalam metode *dropout*, beberapa neuron dipilih secara acak dan tidak digunakan selama pelatihan. *Neuron* yang dibuang atau tidak digunakan akan dihentikan sementara jaringan dan bobot baru tidak diterapkan pada *neuron* saat back propagation. *Dropout* adalah proses yang mempercepat proses *learning* dan mencegah *overfitting*. Menghilangkan suatu *neuron* berarti menghilangkan untuk sementara dari jaringan yang ada. Ini dapat berupa *neuron* yang tersembunyi atau lapisan yang dapat dilihat di dalam jaringan. *Neuron* yang akan dibuang akan dipilih secara acak. Setiap *neuron* akan diberi probabilitas dari 0 hingga 1.

2.2.4 *Google Colab*

Google Colaboratory yang secara luas dikenal sebagai *Google Colab* adalah layanan sumber terbuka yang disediakan oleh *Google* untuk setiap orang yang memiliki akun *Gmail*. *Google Colab* memberikan akses kepada mereka yang kurang memiliki sumber daya atau terbatas dalam kemampuan finansial untuk memiliki *GPU* dalam penelitian mereka. Layanan *Google Colab* menyediakan 12,72 GB *RAM* dan 358,27 GB ruang *harddisk* dalam satu *runtime*. Setiap *runtime* berlangsung selama 12 jam setelah itu *runtime* diatur ulang dan pengguna harus membuat koneksi lagi. Ini untuk memastikan bahwa orang tidak menggunakan *GPU* layanan untuk penambangan mata uang kripto dan ilegal lainnya ilegal lainnya. Setelah pengguna membuka file *Google Colab*, peneliti harus memilih jenis *runtime*. Ada 3 opsi yang tersedia untuk hal yang sama, antara lain :

1. Tidak ada (yang akan menggunakan *CPU* komputer yang digunakan pengguna)
2. *GPU*
3. *TPU* (terutama untuk pemrosesan tensor)

Google Colab pada dasarnya adalah *Notebook Jupyter online* dan memiliki semua fungsi yang ada di dalamnya. Pengguna dapat menghubungkan *Google Colab* ke *Google Drive* mereka. Setelah *Google Drive* dipasang ke *Google Colab*, kemudian

user dapat menggunakannya sebagai *Jupyter Notebook* dan menulis kode kerja ke dalam sel atau memiliki file *python* di *Google Drive* dan menjalankan *file-file* tersebut dari *Google Colab* seperti halnya menjalankannya dari terminal *Linux* tetapi dengan tanda `!` sebelum pernyataan [14].

2.2.5 Model Sequential

Jenis model *sequential* memudahkan untuk pembuatan model di *Keras* dan lapis demi lapis. Bobot setiap lapisan sesuai dengan nilai lapisan berikutnya [15]. Model *sequential* lebih cocok digunakan ketika ingin mengatur lapisan secara berurutan, dimana hanya ada satu sensor *input* yang diteruskan dari satu lapisan ke lapisan lainnya, dan setiap lapisan menghasilkan satu tensor keluaran atau *output*. Meskipun begitu, ada beberapa situasi di mana model *sequential* tidak tepat digunakan, yaitu dalam kasus – kasus berikut :

- a. Input dan output bangun model lebih dari satu.
- b. Pada setiap lapisan memiliki banyak masukan (*input*) atau keluaran (*output*).
- c. Perlu melakukan pembagian lapisan (*layer sharing*).
- d. Ingin menggunakan topologi non-linier (misalnya, koneksi residual atau model multi cabang) [16].

2.2.6 Library Python

Pustaka atau *library* merujuk pada kumpulan modul yang mengandung kode yang bisa dimanfaatkan dalam program. Di *Python*, terdapat berbagai macam pustaka yang dapat digunakan untuk menyerderhanakan program. *Library Python* ini memegang peranan krusial dalam berbagai bidang pembelajaran mesin, ilmu data, visualisasi, manipulasi data, dan bidang lainnya [17].



Gambar 2.3 Library Python

Pada gambar 2.3 merupakan komponen pada *library python* diantaranya sebagai berikut ini :

1. *Numpy*

Paket dasar untuk komputasi ilmiah dengan *python* disebut *numpy*. Ini adalah pustaka atau *library python* yang menyediakan objek *array* multidimensi, berbagai *instance* objek seperti *array* dan matriks bertopeng, serta berbagai rutinitas untuk operasi cepat pada *array* seperti operasi matematika, logika, pengurutan, bentuk, transformasi, pemilihan, I/O, aljabar linier dasar, transformasi *fourier* diskrit, operasi statistik dasar, simulasi stokastik, dll [18].

2. *Tensorflow*

Tensorflow adalah *library* komputasi numerik berbasis *Python* yang *open source* yang mempercepat dan mempermudah penggunaan pembelajaran mesin. *Tensorflow* membantu pengembang membuat grafik aliran data, yaitu struktur yang menjelaskan bagaimana data yang digunakan bergerak melalui berbagai node pemrosesan atau tampilan grafis. Dalam grafik, setiap node mewakili operasi matematika, dan setiap koneksi atau titik akhir antara node tersebut adalah larik data multidimensi, atau tensor [19].

3. *Keras*

Keras adalah paket yang membuat *TensorFlow* lebih mudah digunakan. Namun, *keras* tidak hanya dapat digunakan untuk *TensorFlow*, tetapi juga untuk *Theano* dan *CNTK*. Pada penelitian ini, peneliti menggunakan paket *keras* dengan bahasa pemrograman *python* untuk membantu menjalankan deep learning untuk mengklasifikasi atau mengidentifikasi tanda tangan [20].

4. *Matplotlib*

Matplotlib adalah *library python* yang digunakan untuk membuat visualisasi data lebih menarik dan mudah dipahami. *Matplotlib*, yang diciptakan oleh John Hunter pada tahun 2002, dapat digunakan untuk memvisualisasikan data 2D dan 3D serta menghasilkan gambar berkualitas tinggi yang dapat disimpan dalam berbagai format gambar seperti *JPEG* dan *PNG*. *Matplotlib* tersedia pada *Python* melalui *anaconda*, termasuk *spyder* dan *jupyter notebook* [21].