SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

Perancangan Model *Deep Learning* untuk Penerjemah Bahasa Isyarat SIBI menggunakan *Transfer Learning* MobileNetV2

TUGAS AKHIR

Fikri Pratama Al Fajri
0110220004

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI
AGUSTUS 2024

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

Perancangan Model *Deep Learning* untuk Penerjemah Bahasa Isyarat SIBI menggunakan *Transfer Learning* MobileNetV2

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

Fikri Pratama Al Fajri

0110220004

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : **Fikri Pratama Al Fajri**

NIM : **0110220040**

Depok 23 Juli 2024

Tanda Tangan

Fikri Pratama Al Fajri

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh:

Nama : Fikri Pratama Al Fajri

NIM : 0110220040

Program Studi : Teknik Informatika

Judul Skripsi : Perancangan Model *Deep Learning* untuk Penerjemah Bahasa Isyarat SIBI menggunakan *Transfer Learning* MobileNetV2

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri.

DEWAN PENGUJI

Pembimbing	Penguji
(Ahmad Rio Adriansyah, S.Si., M.Si.)	(Dr. Sirojul Munir S.Si., M.Kom.)

Ditetapkan di : Depok

Tanggal : 23 Juli 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Kepada diri sendiri yang telah menyelesaikan tugas akhir ini dengan semaksimal mungkin.
3. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
4. Bapak Dr. Lukman Rosyidi, ST., MM., MT selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Ibu Tiffany Nabarian, S.Kom, M.T.I selaku Ketua Program Studi Teknik Infomatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Dr. Lukman Rosyidi, ST., MM., MT selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
7. Bapak Ahmad Rio Adriansyah., S.Si., M.Si selaku Dosen Pembimbing Tugas Akhir penulis dalam memberikan arahan, bimbingan serta motivasi dalam penulisan tugas akhir ini.
8. Bapak Dr. Sirojul Munir, S.Si, M.Kom., selaku dosen penguji dalam Tugas Akhir ini.
9. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
10. Seluruh pihak yang terlibat pada Bangkit Academy 2023 yang telah memberikan ilmu yang bermanfaat sehingga saya punya topik dan bisa mengerjakan tugas akhir ini.

11. Untuk Seluruh Teman-Teman yang sudah mendukung saya terutama di grup UIUX, Terimakasih semuanya.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 23 Juli 2024

Fikri Pratama Al Fajri

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama	: Fikri Pratama Al Fajri
NIM	: 0110220040
Program Studi	: Teknik Informatika
Jenis karya	: Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul : **Perancangan Model *Deep Learning* untuk Penerjemah Bahasa Isyarat SIBI menggunakan *Transfer Learning* MobileNetV2.**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal :23 Juli 2024

Yang Menyatakan

(Fikri Pratama Al Fajri)

ABSTRAK

Nama : Fikri Pratama Al Fajri

NIM : 0110220040

Program Studi : Teknik Informatika

Judul : Perancangan Model *Deep Learning* untuk Penerjemah Bahasa Isyarat SIBI menggunakan *TransferLearning* MobileNetV2

Orang-Orang dengan Disabilitas tuna wicara dan tunarungu memiliki kesulitan dalam berkomunikasi karena mereka menggunakan bahasa isyarat dan bahasa ini tidak umum dipelajari. Untuk mengatasi masalah ini diperlukan suatu model *deep learning* yang mampu mendeteksi bahasa isyarat sehingga dapat dibuat aplikasi penerjemah bahasa isyarat yang dapat memudahkan komunikasi antara orang disabilitas dan non disabilitas.

Penelitian ini bertujuan untuk membuat model *deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat SIBI alfabet dengan akurasi yang baik. Model ini menggunakan algoritma CNN dengan arsitektur MobilenetV2 dan metode *transfer learning*.

Hasil dari proses perancangan model ini adalah model memiliki evaluasi akurasi yakni 95,45% yang dapat mendeteksi huruf SIBI dengan baik.

Kata kunci : CNN, *Deep Learning*, Model, MobileNetV2, SIBI

ABSTRACT

Name : Fikri Pratama Al Fajri
NIM : 0110220040
Study Program : *Informatic*
Title : *Designing a Deep Learning Model for SIBI Sign Language Translator using MobileNetV2 Transfer Learning*

People with speech and hearing impairments have difficulty communicating because they use sign language and this language is not commonly learned. To overcome this problem, a deep learning model is needed that is capable of detecting sign language so that a sign language translator application can be created that can facilitate communication between disabled and non-disabled people.

This research aimed to create a deep learning model that could detect SIBI alphabet sign language hand movements with good accuracy. This model used a CNN algorithm with MobilenetV2 architecture and a transfer learning method.

The result of this model design process is that the model has an accuracy evaluation of 95.45% which can detect SIBI letters well.

Keywords: CNN, *Deep Learning*, *Model*, MobileNetV2, SIBI

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Disabilitas merupakan kondisi dimana seseorang memiliki kekurangan atau keterbatasan pada fisik, intelektual, mental dan sensorik. Kondisi tersebut membuat penyandang disabilitas memiliki keterbatasan dalam hal melakukan aktivitasnya. Penyandang disabilitas mencapai 22,97 juta jiwa atau sekitar 8,5% dari jumlah keseluruhan orang Indonesia, dengan jumlah terbanyak adalah orang tua[1]. Jenis dari disabilitas ada beberapa macam yang dimana salah satunya sering terjadi bersamaan yakni adalah TunaWicara dan TunaRungu[2]. TunaWicara adalah suatu kondisi yang dimana seseorang memiliki keterbatasan dalam berkomunikasi atau berbicara. Penyebab dari hal ini biasanya disebabkan oleh kurang atau ada organ yang digunakan untuk berbicara namun tidak berfungsi, organ-organ yang tidak berfungsinya tersebut diataranya seperti rongga mulut, pita suara dan lidah[3]. Berikutnya tunarungu adalah kondisi dimana seseorang tidak dapat mendengar dengan ciri-ciri yang dapat dilihat di intensitas bicara[4]. Menurut data yang bersumber dari Open Data Jabar di Kota Depok dari tahun 2021 sampai dengan 2022 terdapat 658 Jiwa penyandang Tunawicara/Tunarungu.

Berbagai penyebab yang sudah dijelaskan sebelumnya yang membuat seseorang penyandang disabilitas tunarungu serta tunawicara tidak bisa berkomunikasi dengan normal yakni menggunakan mulut dan jika mengalami tunarungu sejak lahir maka tidak dapat mempelajari huruf,kata dan kalimat. Padahal komunikasi merupakan sesuatu yang sangat diperlukan oleh manusia[5] dan jika ada seseorang yang tidak bisa berkomunikasi dengan baik akan menyulitkan dalam melakukan aktivitas. Maka dari itu diperlukan suatu aplikasi penerjemah bahasa isyarat yang mampu mendeteksi gerakan tangan bahasa isyarat yang sudah dibakukan oleh pemerintah. Sebelumnya dalam perancangan aplikasi penerjemah bahasa isyarat ini

penulis dibantu dengan beberapa orang dalam proses pengembangannya menjadi suatu aplikasi nama aplikasi tersebut adalah DIFA *Sign Language Translator For Children*. Ada tiga bagian modul yang di kerjakan yang pertama adalah bagian pengembangan di bagian *frontend* untuk tampilan aplikasi ponsel android. Bagian ini juga melibatkan proses pembuatan *User Interface* dan *User Experience* (UI UX) yang dibuat semenarik mungkin karena dengan target pengguna adalah anak-anak. Selanjutnya adalah bagian *backend* melakukan proses API dan menyambungkan dengan *cloud*. Dan terakhir adalah bagian sebagai pengembang model *machine learning* yang digunakan untuk fitur utama pada aplikasi yakni membuat mesin dapat mendeteksi bahasa isyarat berjenis Sistem Bahasa Isyarat Indonesia (SIBI). Tugas akhir ini akan berfokus pada pengembangan model *Deep learning* menggunakan tensorflow sebagai *framework.* Algoritma yang digunakan adalah *Convolutional Neural Network* (CNN) dengan menggunakan arsitektur MobileNetV2 dengan metode *transfer learning* untuk membuat model lebih mudah dan komputasi yang ringan[6] namun tetap menghasilkan akurasi yang baik. Dalam penelitian ini akan berfokus pada pengembangan model yang dapat digunakan untuk mendeteksi objek gerakan tangan bahasa isyarat SIBI. Tujuan dari penelitian ini adalah membuat suatu model *deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat SIBI sehingga orang-orang non disabilitas dapat berkomunikasi dengan orang disabilitas dan bisa terciptanya lingkungan yang inklusif.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya maka terdapat beberapa rumusan masalah yang di antaranya sebagai berikut:

1. Bagaimana proses membangun model menggunakan algoritma CNN menggunakan arsitektur MobileNetV2 menggunakan *transfer learning* agar dapat mendeteksi gerakan tangan bahasa isyarat SIBI?

2. Bagaimana hasil dari akurasi model yang sudah dibuat pada proses pembuatan model deteksi gerakan tangan bahasa isyarat SIBI?

1.3 Tujuan dan Manfaat Penelitian

1.3.2 Tujuan Penelitian

1. Membuat suatu model menggunakan algoritma CNN dengan menggunakan *transfer learning* arsitektur MobileNetV2 yang dapat mendeteksi gerakan tangan bahasa isyarat SIBI.
2. Mengetahui hasil dari akurasi model yang sudah dibuat apakah dapat mendeteksi gerakan tangan bahasa isyarat SIBI alphabet.

1.3.2 Manfaat Penelitian

1. Bagi masyarakat adanya penelitian ini perancangan model yang bisa membantu menerjemahkan bahasa isyarat SIBI akan dapat membuat komunikasi lebih mudah antara orang disabilitas tunawicara/tunarungu dengan orang non disabilitas.
2. Selanjutnya untuk institusi pendidikan penelitian ini dapat digunakan sebagai bahan referensi ilmiah yang bisa digunakan kembali pada penelitian selanjutnya yang memiliki topik mengenai pendeteksian objek.

1.4 Batasan Masalah

Berdasarkan dari tujuan serta manfaat yang telah dijelaskan sebelumnya. Maka didapatkan Batasan Masalah dari penelitian ini yang diantaranya sebagai berikut, yakni :

1. Penelitian ini berfokus kepada pengembangan model *deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat SIBI yang menggunakan algoritma *Convolutional Neural Network* (CNN) arsistekur MobilNetV2 dengan metode *transfer learning*.

2. Dalam proses pengembangan model hanya menggunakan data gambar gerakan tangan bahasa isyarat yang bersumber dari kaggle. Isi data yang berada di dalamnya adalah gerakan tangan bahasa isyarat SIBI yang statis yakni hanya huruf alfabet.
3. Pada proses pendeteksian, model hanya dapat mendeteksi alfabet Indonesia selain huruf j dan z dikarenakan kedua huruf tersebut memiliki gerakan tangan yang tidak statis sehingga harus ada gerakan tambahan agar dapat membentuk gerakan bahasa isyarat tersebut.

1.5 Sistematika Penulisan

Pada tugas akhir ini penulisan sistematikanya ditulis sebagai berikut:

1. BAB I PENDAHULUAN

   Pada Bab ini menjelaskan mengenai latar belakang permasalahan yang diangkat dari penelitian yang berikutnya dilanjutkan dengan perumusan masalah, tujuan penelitian dan manfaat penelitian, batasan masalah dan terakhir sistematika penulisan.
2. BAB II KAJIAN LITERATUR

   Pada bagian Bab ini membahas tentang penjelasan penelitian yang sebelumnya dianggap relevan dengan penelitian ini. Berikutnya akan memberikan penjelasan singkat tentang landasan teori yang terkait dengan penelitian ini.
3. BAB III METODOLOGI PENELITIAN

   Bab ini membahas mengenai uraian secara rinci mengenai alat dan bahan yang digunakan dalam penelitian ini. Selain itu diberikan juga penjelasan secara detail tentang langkah-langkah yang harus dilalui untuk mencapai tujuan dan kesimpulan penelitian.

4. BAB IV IMPLEMENTASI HASIL

   Bab ini menjelaskan tentang pemaparan hasil penelitian dan pembahasan tentang bagaimana alur proses pemrosesan data serta menampilkan hasil akurasi model yang sudah dibuat.

5. BAB V KESIMPULAN DAN SARAN

   Bab ini merupakan bab terakhir yang isinya menjelaskan hasil kesimpulan yang telah diperoleh selama proses penelitian, serta beberapa saran dari penulis sehingga tujuan dari penelitian ini dapat terwujud dan bisa menjadi pertimbangan untuk peneliti selanjutnya.

BAB II
KAJIAN LITERATUR

Pada Bab ini, akan disajikan kajian tentang definisi, teori dengan analisis penelitian. Selanjutnya akan ditampilkan juga penelitian terkait yang relevan dengan penelitian ini.

2.1 Bahasa Isyarat

Bahasa isyarat adalah bahasa yang digunakan untuk berkomunikasi oleh penyandang disabilitas terkhusus tunarungu dan tunawicara untuk berbicara atau berkomunikasi dengan masyarakat non disabilitas dan sesamanya. Dalam praktiknya bahasa isyarat menggunakan gerakan tubuh, terutama tangan dan juga mimik wajah yang kemudian membentuk simbol-simbol yang dapat diartikan sebagai huruf atau kata. Dalam perkembangannya di Indonesia ada dua jenis bahasa isyarat yang cukup populer dan sering digunakan yang pertama adalah BISINDO (Bahasa Isyarat Indonesia). BISINDO dibuat dan dikembangkan sendiri oleh masyarakat tunarungu[7]. Selanjutnya ada SIBI (Sistem Isyarat Bahasa Indonesia) bahasa isyarat ini sudah dibakukan oleh KEMENDIKBUD (Kementerian Pendidikan dan Kebudayaan) pada tahun 1994. SIBI biasa dipakai dalam pembelajaran di sekolah luar biasa dan sudah sesuai dengan kaidah bahasa Indonesia yang baik dan benar[8].

2.2 Sistem Isyarat Bahasa Indonesia

SIBI (Sistem Isyarat Bahasa Indonesia) adalah sebuah bahasa isyarat yang sudah diresmikan oleh pemerintah[7] dan digunakan sebagai acuan di dalam pembelajaran Sekolah Luar Biasa (SLB)[9]. SIBI sendiri memiliki kemiripan dengan bahasa isyarat di Amerika yakni *American Sign Language* (ASL) dikarenakan SIBI mengadopsi dari ASL[7].

Dalam penggunaanya SIBI lebih sering digunakan dalam acara formal yang diselenggarakan oleh pemerintah dan juga pada aktivitas pembelajaran di

sekolah. Salah satu penerapan penggunaan SIBI adalah dalam bentuk alfabet dimana setiap gerakan tangan merepresentasikan satu huruf namun hampir semua gerakan tangan bersifat statis atau diam kecuali huruf J dan Z yang bersifat dinamis karena diperlukan gerakan tambahan untuk membentuk huruf itu. Penggambaran SIBI dalam bentuk alfabet dapat dilihat pada gambar 2.1 berikut ini.

Gambar 2. 1 SIBI Alfabet

Sumber : https://daaitv.co.id/

2.3 Machine Learning

Machine learning atau dalam bahasa Indonesia disebut pembelajaran mesin adalah pengembangan teknologi yang dalam penerapannya dapat meniru kecerdasan manusia dengan baik berdasarkan algoritma pembelajaran mesin yang digunakan[7]. *Machine learning* dapat diimplementasikan dalam berbagai bidang untuk menyelesaikan masalah sesuai dengan kebutuhan seperti medis atau kedokteran, lalu lintas, industri, dan teknologi [10].

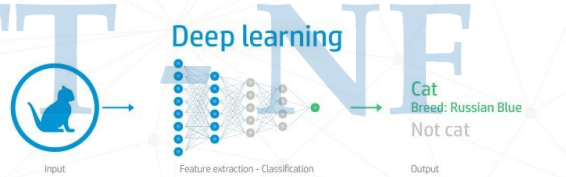
Dalam pembuatannya *machine learning* memerlukan data sampel yang digunakan sebagai pelatihan sesuai kegunaanya. *Machine learning* ini

dikembangkan dengan beberapa disiplin ilmu seperti matematika, statistika dan *data mining*.

Jika dilihat dari teknik yang digunakan ada 2 teknik yakni *supervised learning* dan *unsupervised learning*. *Supervised learning* merupakan salah satu tipe pembelajaran algoritma *machine learning* yang menggunakan dataset yang disebut (*training data*) untuk melakukan prediksi. Selanjutnya adalah *unsupervised learning* yaitu algoritma *machine learning* yang menggunakan data *input* label response dan selanjutnya dapat ditarik kesimpulan dari data tersebut [11].

2.4 Deep Learning

Deep Learning adalah bagian dari pembelajaran mesin dan *artificial intelliegence* yang dimana ini merupakan pengembangan dari *neural network multiple layer*[11]. Pengembangan ini dilakukan agar model dapat diberikan tugas seperti pengenalan suara, dapat mendeteksi objek dan menerjemahkan bahasa dengan tepat. Memiliki perbedaan dengan *machine learning*, *deep learning* secara otomatis dapat mengekstrak fitur dari data mentah berupa video, gambar atau teks tanpa aturan kode atau pengetahuan manusia[12].



Gambar 2. 2 Perbedaan *Machine Learning* dan *Deep Learning*

Sumber : https://dqlab.id/

Deep Learning menggunakan metode *learning* yakni *Artificial Neural Network* (ANN) atau disebut Jaringan Saraf Tiruan yang bentuknya memiliki banyak lapisan (*multilayer*) yang bentuknya dibuat mirip seperti otak pada manusia. Otak manusia terdiri dari banyak neuron-neuron yang terhubung satu dengan yang lainnya dan membentuk jaringan yang rumit dan mirip seperti itulah ANN [11]. Selanjutnya *Deep Learning* memiliki beberapa algoritmanya sendiri yaitu, *Convolutional Neural Networks* (CNN), *Long Short Term Memory Network* (LSTM) *dan Recurrent Neural Network* (RNN) [13]. Contoh dari penerapan deep learning untuk kegiatan sehari hari adalah, pengenalan wajah biometrik, *voice assistant* pada *smartphone* dan mobil dengan *self driving* seperti pada mobil tesla[12].

2.5 Transfer Learning

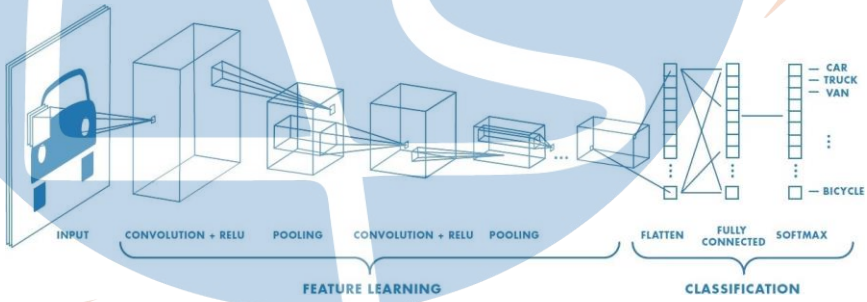
Transfer Learning adalah suatu metode yang menggunakan *network* yang sebelumnya sudah dilatih dan digunakan sebagai awal untuk mempelajari tugas baru[14]. *Transfer learning* mempunyai lapisan arsitektur *pooling* dan *convolutional* yang lebih dalam jika dibandingkan dengan arsitektur CNN sederhana[15]. Melakukan *fine tuning network* menggunakan transfer learning akan jauh lebih cepat dan akan lebih mudah dikarenakan biasanya dalam melakukan *fine tuning* ini akan lebih banyak upaya jika hanya menggunakan ektraksi sederhana. Ini dikarenakan dalam menggunakan *transfer learning* menggunakan jaringan yang sebelumnya di *training* dan mesin sudah mempelajarinya untuk melakukan proses ektraksi serangkaian fitur yang berbeda sehingga jaringan akan lebih akurat. Dalam menggunakan *dataset* yang memiliki jumlah lebih kecil metode ini juga akan tetap bekerja lebih baik, karena jaringan yang baru sudah belajar untuk mempelajari fitur yang baru [14].

2.6 Convolutional Neural Network

Convolutional Neural Networks (CNN) merupakan algoritma pengembangan dari algoritma *Multilayer Perceptron* (MLP) yang

difokuskan dan didesain untuk melakukan pengolahan pada data dua dimensi. Alasan banyak digunakan pada data citra atau data gambar dikarenakan Algoritma CNN ini termasuk ke dalam jenis *Deep Neural Network* yang memiliki kedalaman jaringan yang tinggi. *NeoCognitron* merupakan nama sebelum CNN yang pertama kali dikembangkan oleh peneliti dari NHK *Broadcasting Science Research laboratories* bernama Kunihiko Fukushima yang berasal dari Tokyo, Jepang [16].

Di tahun 2010 peneliti yang bernama Yaan LeCun, menyatakan pendapatnya mengenai CNN yang merupakan pengembangan dari MLP yang didasari oleh tiga ide yakni *local receive field*, *weight sharing* dan terakhir *spatial subsampling*. Ketiga hal tersebut dijelaskan ke dalam dua jenis layer yakni *convolution layer* dan *pooling layer* yang disusun pada *featur map*. Arsitektur CNN terbagi menjadi dua bagian yakni *feature extraction layer* dan *fully connected layer* (MLP) dan kontribusi CNN yang ada pada *feature extraction layer* adalah *convolution* dan *pooling layer* [17].

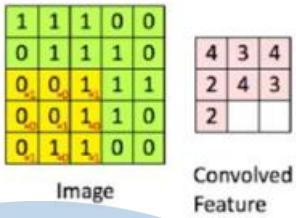


Gambar 2. 3 Ilustrasi Arsitektur CNN

Sumber gambar : www.mathworks.com

Convolutional layer merupakan bagian utama dari pengembangan model CNN yang sebagian besar dari prosesnya adalah komputasi yang menggunakan proses operasi matematika linear aljabar dengan mengalikan matriks dari filter pada gambar. Hal ini dilakukan dengan tujuan konvolusi pada gambar dapat mengekstraksi fitur dari gambar input. Bobot yang ada

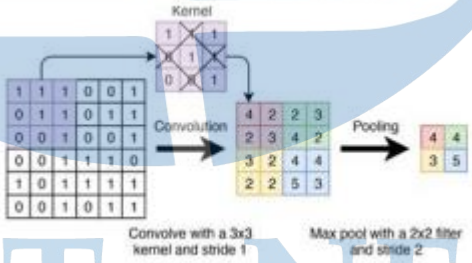
pada layer menyesuaikan kernel konvolusi yang digunakan sehingga kernel konvolusi dapat dilatih sesuai dengan input pada CNN.



Gambar 2. 4 Proses Konvolusi *Layer* [17]

Sumber gambar : jurnal "Sistemasi: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network"

Selanjutnya adalah *Pooling Layer* yang dimana dalam prosesnya adalah mereduksi ukuran dari sebuah data gambar yang bertujuan untuk meningkatkan variasi posisi dari fitur dan juga membagi *output* dari *convolutional layer* menjadi beberapa bagian grid kecil dengan nilai maksimal setiap dari *grid* yang digunakan menyusun matriks gambar yang direduksi menggunakan operasi *max pooling*. Dalam proses itu dapat memastikan fitur yang didapatkan akan sama walaupun objek gambar mengalami pergeseran.



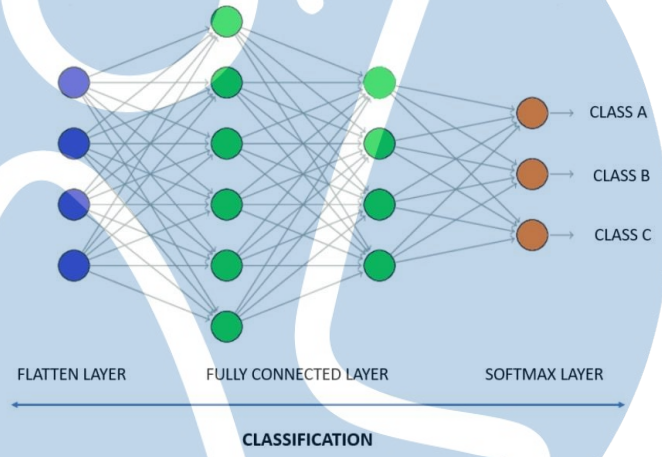
Gambar 2. 5 Operasi *Max Pooling* [17]

Sumber gambar : jurnal "Sistemasi: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network"

Secara keseluruhan pada umumnya *pooling layer* akan mengikuti *layer* konvolusi, digunakan untuk mengurangi dimensi pada *feature map* dan

mempercepat dalam komputasi karena parameter yang ada harus dirubah menjadi semakin sedikit. Secara prinsipnya *pooling layer* adalah filter dengan *stride* dan ukuran yang akan bergeser di seluruh area pada *feature map.* Cara ini umumnya digunakan pada layer *Max pooling* dan *Average pooling* [17].

Pada pembuatan arsitektur CNN terdapat *layer* diletakan di akhir setiap pembuatan model. Dalam *layer* ini, setiap neuron terhubung ke semua neuron pada *layer* sebelumnya, proses ini dinamakan *Fully Connected. Layer* yang digunakan untuk pengklasifikasian pada arsitektur CNN.



Gambar 2. 6 *Fully Connected Layer*

Sumber : https://indiantechwarrior.com/

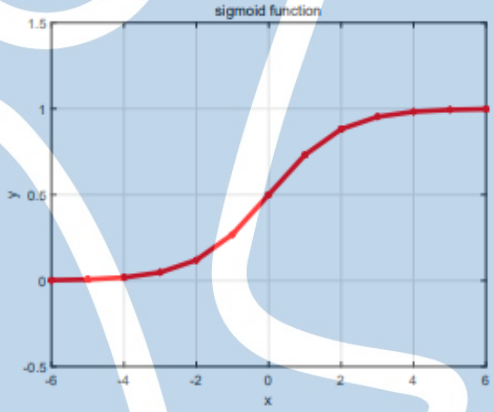
Masukan *fully connected layer* ini berasal dari *pooling layer* terakhir atau dari lapisan konvolusional. *Output* dari lapisan ini akan mewakili luaran akhir dari CNN .

Selanjutnya adalah fungsi aktivasi, fungsi aktivasi atau *activation function* yang mengacu pada fitur neuron yang dapat diaktifkan, dipertahankan dan dipetakan menggunakan fungsi non linear, yang dapat digunakan untuk menyelesaikan masalah non linear. Fungsi aktivasi digunakan untuk meningkatkan kemampuan pada model yang dibuat. Terdapat 3 jenis fungsi

aktivasi yang pertama adalah *sigmoid*, tanh dan relu. *Sigmoid* memiliki formula sebagai berikut :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.\ 1)$$

Sigmoid memiliki output antara 0 dan 1 karena nilai outputnya terikat pada terikat antara 0 dan 1. Penggunaaan sigmoid cocok untuk pembuatan model prediksi probabilitas sebagai *output* karena nilai dari probabilitas adalah antara 0 dan 1. Untuk grafik *sigmoid* dapat dilihat pada gambar 2.5



Gambar 2. 7 Grafik fungsi *sigmoid*[18]

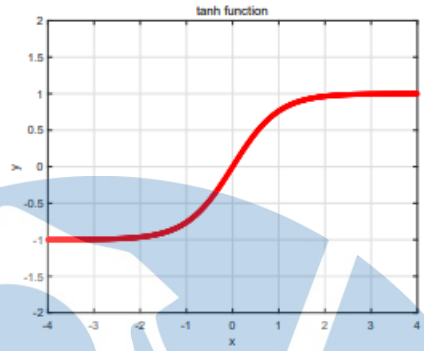
Sumber gambar : Jurnal "The influence of the activation function in a convolution neural network model of facial expression recognition"

Selanjutnya adalah fungsi *tanh*. Fungsi *tanh* merupakan versi terbaru dari *sigmoid* yang memiliki formula sebagai berikut :

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.\ 2)$$

Memiliki fungsi yang relatif sama dengan *sigmoid* namun memiliki kelebihan yakni tingkat konvergensinya lebih tinggi dibandingkan dengan

fungsi *sigmoid* namun memiliki masalah dengan *gradient diffusion,* grafik kurva fungsi *tanh* dapat dilihat pada gambar 2.6



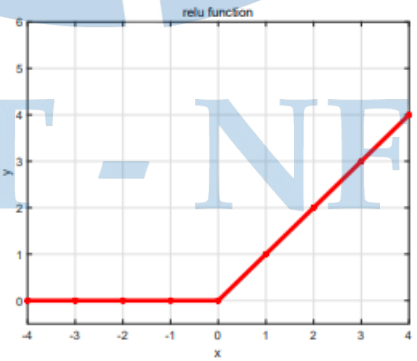
Gambar 2. 8 Kurva Grafik fungsi *tanh*[18]

Sumber : Jurnal "The influence of the activation function in a convolution neural network model of facial expression recognition"

Selanjutnya adalah *ReLu (Rectified Linear Unit*) yang merupakan *activation function* yang sangat populer pada saat ini pada pembuatan model *deep learning*. *ReLu* memiliki formula sebagai berikut :

$$f(x) = Max\ (0, x) \tag{2. 3}$$

Gambar grafik kurva *ReLu* dapat dilihat pada gambar 2.7



Gambar 2. 9 Kurva grafik fungsi *ReLu*[18]

Sumber : Jurnal “The influence of the activation function in a convolution neural network model of facial expression recognition”

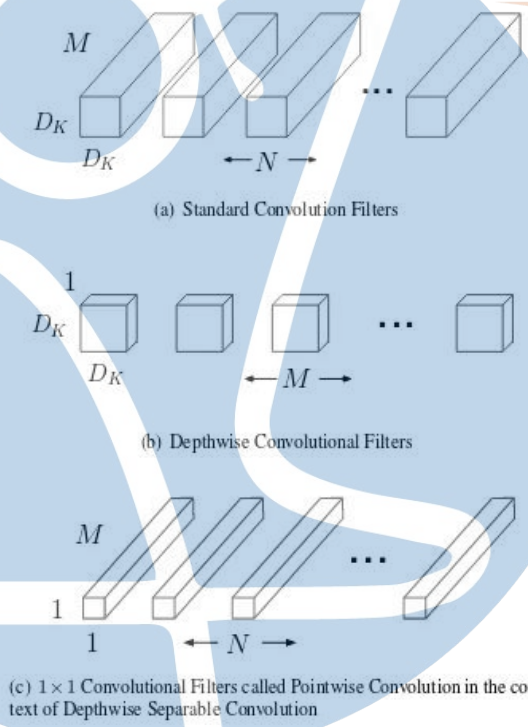
Pada gambar 2.7 dapat dilihat bahwa fungsi ini akan memaksa keluaran menjadi nol jika nilai masukan kurang dari atau sama dengan nol. Jika tidak itu akan berhasil nilai *output* sama dengan nilai *input.* Jika dibandingkan dengan *sigmoid* dan juga tanh, fungsi *ReLu* memiliki kecepatan komputasi yang jauh lebih cepat[18].

CNN memiliki banyak arsitektur, arsitektur model ini merupakan faktor yang cukup penting dalam meningkatkan kinerja dari berbagai macam aplikasi. Banyak modifikasi yang sudah dilakukan pada arsitektur CNN ini yang dimulai pada tahun 1989 dan sampai saat ini. Modifikasi ini meliputi reformulasi struktural, regularisasi, optimalisasi parameter dan lainnya. Sebaliknya,perlu diingat bahwa peningkatan utama pada kinerja CNN sebagian besar ada karena reorganisasi unit pemrosesan serta pengembangan blok baru. Secara khusus perkembangan yang paling baru pada arsitektur CNN adalah dilakukan pada *depth network* [19]. Banyak model dalam arsitektur CNN yang diantaranya adalah AlexNet yang berhasil memenangkan kompetisi ImageNet *Competition* pada tahun 2012. Berikutnya bermunculan arsitektur CNN yang lain yakni GoogleNet, ResNet, Inception [20], dan MobileNet [21].

Menggunakan CNN memiliki manfaat dan keuntungannya tersendiri dibandingkan dengan menggunakan *neural network* yang lainnya di dalam lingkungan *computer vision* yang pertama adalah mempertimbangkan CNN adalah fitur pembagian bobot, yang mengurangi jumlah pada parameter jaringan yang bisa dilatih sesuai gilirannya hal ini membantu jaringan meningkatkan generalisasi dan menghindari *overfitting.* dan berikutnya adalah dalam implementasi pada suatu jaringan berskala besar CNN akan jauh lebih mudah digunakan dibandingkan dengan *neural network* yang lainnya [19].

2.7 MobileNetV2

MobileNet adalah salah satu arsitektur dari Algoritma CNN yang merupakan suatu model kecil yang memiliki latensi rendah daya yang diukur untuk memenuhi batasan sumber daya untuk kasus penggunaan yang berbeda[22]. Perbedaan mendasar antara arsitektur MobilNet dengan CNN adalah di bagian lapisan konvolusi, jika CNN menggunakan ketebalan filter yang sesuai dengan ketebalan pada input gambar, pada MobileNet konvolusi dibagi menjadi *depthwise convolution* dan *pointwise convolution.*



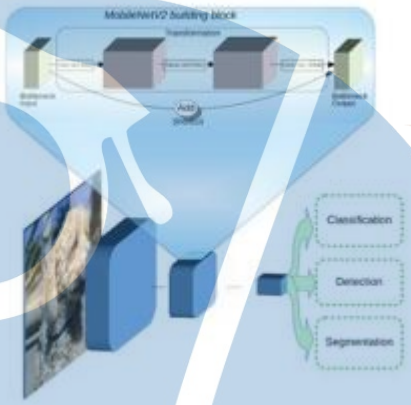
Gambar 2. 10 (a) Konvolusi standar pada CNN, (b) dibagi menjadi dua lapisan: *depthwise convolution dan pointwise convolution*,(c) untuk membuat filter terpisah secara mendalam (*depthwise*)

Sumber : https://medium.com/nodeflux/

MobileNet V2 memiliki peningkatan kinerja model yang memungkinkan membangun rangkaian model seluler yang lebih efisien[23]. Penggunaan kata *mobile* pada arsitektur ini di tunjukan kepada penggunaanya yang dapat digunakan untuk ponsel atau mobile[24] dan MobileNet V2 juga merupakan

ekstraktor fitur yang efektif untuk melakukan deteksi dan juga segmentasi pada objek.

Masih sama seperti mobileNetV1, MobileNetV2 menggunakan *depthwise* dan *pointwise convolution*. Dengan menambahkan 2 fitur baru pada Mobile Net V2 yakni *linear bottleneck* dan *shortcut connections* antar *bottlenecks* yang struktur dasar pada arsitektur ini dapat dilihat pada gambar 2.11.



Gambar 2. 11 *bottleneck* yang terdapat dalam arsitektur MobileNet V2[22]

Sumber Gambar : Jurnal "Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360° Ternormalisasi"

Pada bagian *bottleneck* dapat dilihat terdapat input dan juga output antar model sedangkan pada *layer* bagian dalam mengenkapsulasi kemampuan dari model untuk mengubah input dari konsep tingkat yang rendah ke tingkat yang lebih tinggi.

Perbedaan pada model MobileNetV2 dan MobileNetV1 adalah dari arsitektur dari setiap model, pada MobileNetV1 hanya memiliki satu *layer* dan MobileNetV2 memiliki dua *layer* sehingga pada prosesnya membuat MobileNetV2 lebih cepat jika dibandingkan dengan MobileNetV1. MobileNetV2 juga menggunakan operasi yang dua kali lebih sedikit, memiliki akurasi yang lebih tinggi dan juga membutuhkan 30 persen parameter yang lebih sedikit serta 30 sampai 40 persen lebih cepat[22].

2.8 Optimizer

Optimizer merupakan suatu fungsi matematika yang bergantung pada variabel bebas dari model yakni bias dan nilai bobot. *Optimizer* juga merupakan salah satu metode yang digunakan untuk meminimalisasi nilai *output* dari *cost function.* Cara kerja dari *optimizer* adalah dengan mempelajari bagaimana cara mengubah bias dan nilai bobot pada neural network untuk mengurangi *error.* Terdapat beberapa jenis *optimizer* yang dikembangkan dan biasa digunakan dalam pengembangan model *Deep learning* diantaranya adalah sebagai berikut:

1. SGD (*Stochastic Gradient Descent*)

   SGD merupakan *optimizer* yang biasa digunakan dalam proses training model machine learning. Pada prosesnya SGD, nilai bobot diubah pada setiap satu data *training* selesai dievaluasi. Metode ini memiliki kelebihan yakni prosesnya lebih cepat jika dibandingkan dengan optimizer BGD (*Batch Gradient Descent*), pada perkembangan model juga lebih detail dikarenakan proses yang dimana perubahan dilakukan pada saat satu data selesai di *training*. Dibalik kelebihan yang dimiliki optimizer SGD juga memiliki kekurangan yakni daya komputasinya menjadi tinggi dan banyaknya *noise* yang muncul pada *gradient* dikarenakan proses perubahan yang menjadi lebih sering terjadi dan menyebabkan tingkat *error* melonjak dan tidak berkurang secara bertahap[25].

2. RMSProp (*Root Mean Square Propagation*)

   RMSProp adalah optimasi yang menggunakan besarnya gradient terbaru untuk menormalkan *gradient*, fungsi optimasi RMSprop dapat menjaga pergerakan rata-rata diatas *gradient root mean square*, sehingga diberi nama RMS. Optimasi ini dapat mempertahankan rata-rata dari kuadrat *gradient* pada setiap bobot. RMSProp memiliki kesamaan dengan AdaProp (*Adaptive Realtime Perceptron Optimizer*), yang merupakan perbaikan dari AdaGard (*Adaptive Gradient Algorithm*)[26].

3. Adam (*Adaptive Moment Estimation*)

   Adam adalah optimizer yang penggunaanya populer pada pengembangan model *deep learning* dan merupakan salah satu algoritma optimizer yang bisa menggantikan prosedur SGD klasik yang dapat memperbarui *weight network* berdasarkan data *training* secara berulang. Dapat digambarkan adam merupakan penggabungan keuntungan dari dua ekstensi dari SGD yakni AdaGard dan RMS. Dari penggambaran tersebut adam dapat memberikan pengoptimalan suatu algoritma yang dapat mengatasi sparse gradient pada *noisy problem*[26]

   .

2.9 Loss Function

Loss function adalah proses yang mengukur margin kesalahan antara prediksi model dan nilai target. *Lost function* juga merupakan komponen yang penting dalam pembuatan model machine learning yang dapat mengukur perbedaan antara keluaran yang diprediksi oleh algoritma *machine learning* dan nilai target yang sebenarnya. *Loss function* terkadang disamakan dengan *performance matriks* dikarenakan keduanya digunakan untuk melakukan evaluasi performa model pada *deep learning* namun keduanya tetap memiliki tujuan yang berbeda. *Loss function* digunakan selama training untuk mengoptimalkan parameter model.

Selanjutnya *loss function* dapat digunakan untuk pembuatan model tugas deep learning yang umum, salah satunya adalah klasifikasi. Klasifikasi termasuk kedalam *supervised machine learning* yang dimana model dilatih untuk melakukan prediksi kelas atau kategori *input* data tertentu. Klasifikasi bertujuan untuk mempelajari pemetaan dari *input* kelas ataupun kategori tertentu. ada beberapa jenis klasifikasi yakni *classification binary* yang dimana model dilatih hanya untuk mendeteksi satu dari dua kelas. Selanjutnya adalah *multi class classification* yang digunakan untuk memprediksi beberapa kelas. terakhir *multi label classification* yang

digunakan untuk memprediksi beberapa label untuk satu titik data. Berikut adalah *loss function* yang dapat digunakan pada pembuatan model machine learning atau deep learning untuk klasifikasi diantaranya adalah :

1. BCE (*Binary Cross Entropy*)

   BCE (*Binary Cross Entropy*) atau dikenal juga *log loss*, adalah salah satu loss function yang biasa digunakan pada *classification binary*. Pada dasarnya BCE mengukur suatu ketidaksamaan pada probabilitas prediksi suatu kelas dan pada label sebenarnya. Pada klasifikasi biner, kelas sebenarnya akan diwakili oleh *vector* yang dikodekan *one-hot* yang memiliki nilai 1, dan kelas yang lain memiliki nilai 0. Rumus pada BCE dapat didefinisikan sebagai berikut:

$$L\,(y,p) = -(y\log(p) + (1-y)\log(1-p)) \tag{2. 4}$$

   Penggunaan BCE memiliki keuntungan yakni mudah dihitung dan memberikan suatu interpretasi probabilistik pada keluaran model. Hal ini juga yang menghasilkan permukaan pengoptimalan yang halus dan kurang sensitif pada *outlier* dibandingkan pada *loss function* yang lain.

2. CCE (*Categorical Cross Entropy*)

   CCE (*Categorical Cross Entropy*) adalah suatu *loss function* yang biasa digunakan untuk pembuatan model klasifikasi untuk banyak kelas atau *multi class*. CCE mengukur perbedaan pada distribusi probabilitas yang diprediksi dengan distribusi sebenarnya. Mengingat pada distribusi probabilitas yang diprediksi, didefinisikan sebagai rata-rata kemungkinan pada log negatif dari kelas sebenarnya maka rumus CCE dapat didefinisikan sebagai berikut:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} y_{i,j}\log\left(P_{i,j}\right) \tag{2. 5}$$

ada loss function ini loss dihitung pada setiap sampel dan dirata-ratakan untuk seluruh kumpulan data.

3. *Focal Loss*

Focal loss mulai diperkenalkan oleh Tsung-Yi dan rekan-rekannya yang merupakan suatu variasi dari CCE yang dapat mengatasi masalah pada ketidakseimbangan kelas, yang terjadi saat jumlah sampel positif jauh lebih kecil dari pada jumlah sampel negatif. Karena pada kasus sample positif lebih kecil dari jumlah sampel negatif, model akan cenderung berfokus ke sampel negatif dan akan mengabaikan sampel positif, sehingga dapat menghasilkan performa yang buruk. *Focal loss* dapat mengatasi masalah ini dengan cara menurunkan bobot sampel negatif mudah dan menaikkan bobot pada sampel positif. Rumus *focal loss* dinyatakan sebagai berikut :

$$FL(P_T) = -a_t(1 - P_t)\log(P_t) \quad (2.6)$$

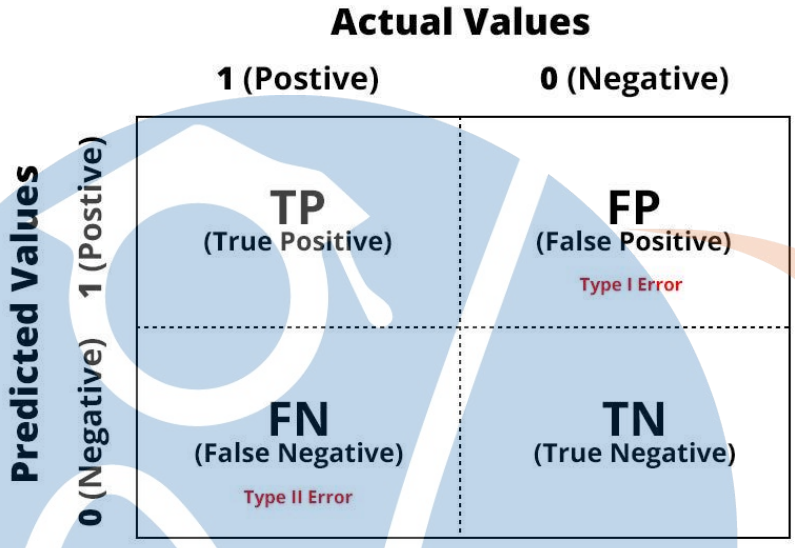
Pada sumber makalah aslinya penggunaan focal loss dikombinasikan dengan fungsi aktivasi *sigmoid* untuk *binary classification* dan *multi class classification*. Hal ini dilakukan untuk meningkatkan kinerja model deteksi objek dan segmentasi semantik[27].

2.10 Confusion Matrix

Confusion matrix atau dalam bahasa Indonesia matriks kebingungan adalah matriks yang berisi data pada target prediksi yang selanjutnya dibandingkan dengan data pada target terbaru. Data aktual merupakan nila yang kita miliki sedangkan data prediksi merupakan nilai yang diperoleh dari hasil pemodelan *Deep learning*. Simbol untuk data prediksi adalah $\hat{y}$ dan untuk data terbaru simbolnya adalah *y*.

Dalam proses pembuatan model *Deep learning confusion matrix* dibutuhkan agar dapat diketahui apakah model yang dibuat bisa bekerja

sesuai yang diinginkan. Pada *confusion matrix* didalam terdapat berbagai performa yang bisa diukur seperti *precision, accuracy, precision, recall, specificity* dan terakhir *F1 Score*. pada gambar 2.6 dapat dilihat struktur dari *confusion matrix* sebagai berikut:



Gambar 2. 12 Tabel *confusion matrix*

Sumber : ksnugroho.medium.com

Salah satu ukuran performa yang terdapat dalam *confusion matrix* adalah *accuracy*. *Accuracy* mengacu kepada prediksi seberapa akurat model *Deep learning* yang sudah dibuat dapat memprediksi nilai dengan benar sehingga *accuracy* dapat didefinisikan sebagai rasio prediksi benar dengan keseluruhan data. Untuk formula dari *accuracy* adalah[28]

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \qquad (2.\ 7)$$

Selanjutnya setelah *accuracy* adalah *precision* (*Positive Predictive Value*) pada *precision* mengacu kepada keakuratan dari model yang dibuat yang digambarkan pada tingkat keakuratan antara data aktual dengan hasil prediksi. Secara sederhanya *precision* memberikan kita informasi berapa banyak prediksi yang benar-benar positif dari total prediksi positif. Untuk formula dari *precision* adalah sebagai berikut:

$$Precision = \frac{TP}{TP + FP} \quad (2.\ 8)$$

Berikutnya adalah *recall* yang pada prosesnya *recal* atau *sensitivity* menggambarkan keberhasilan suatu model yang dibuat dalam menemukan kembali sebuah informasi, yang biasa digunakan dalam pembuatan model information *retrieval*. Secara jelas *recall* adalah suatu rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar. Rumusnya sebagai berikut:

$$Recall = \frac{TP}{TP + FN} \quad (2.\ 9)$$

Terakhir adalah *F1 Score* adalah perbandingan rata-rata *precision* dan recall yang dibobotkan. Rumusnya adalah sebagai berikut:

$$F1\ Score = 2x\frac{Recall\ x\ Precission}{Recall +\ Precision} \quad (2.\ 10)$$

2.11 Tensorflow

Tensorflow adalah suatu *framework open source* yang banyak digunakan dalam proses pembuatan model Deep learning yang dibuat dan dikembangkan oleh Google. Tensorflow memiliki *tools* dan *library* dan sumber daya komunitas yang besar yang memungkinkan para tim peneliti dan tim pengembang dapat dengan mudah membangun dan mengembangkan suatu model dan aplikasi *Deep learning*. Arstiketur dari Tensorflow sendiri fleksibel dan memungkinkan melakukan komputasi dengan mudah di berbagai *platform* seperti (CPU, GPU dan TPU) dan dari perangkat seperti *cloud, desktop* dan terakhir perangkat seluler[29].

2.12 Streamlit

Streamlit adalah suatu kerangka kerja web yang dibuat untuk menampilkan kinerja model dan visualisasi menggunakan bahasa pemrograman python dengan mudah. Memiliki kelebihan yakni cepat dan minimalis namun juga

mempunyai tampilan yang ramah pengguna. Pada streamlit pengguna dapat melakukan beberapa hal seperti dapat, memasukan teks, melakukan upload gambar dan menggunakan elemen yang ada di HTML (*Hypertext Markup Language*). Streamlit dapat digunakan secara gratis dan dijalankan menggunakan *code editor* seperti visual studio code[30].

Saat menjalankan streamlit terdapat dua cara yakni menjalankan langsung di jaringan *localhost* atau melakukan *deploy* saat dirasa sudah siap digunakan. Streamlit tidak memerlukan bantuan penyedia *web server localhost*, karena saat menjalankan streamlit akan langsung diarahkan menuju *localhost* yang sudah disediakan langsung oleh streamlit.

Tabel 2. 1 Penelitian terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	M. Bahrul Subkhi, Mochamad Yuda Trinurais, Ridho Kuncoro Adji Wibowo, Bryan Rizqi Prakosa[31],2024	Deteksi Bahasa Isyarat Berdasarkan SIBI (Sistem Bahasa Isyarat Indonesia) menggunakan *Transfer Learning*	Deteksi Bahasa Isyarat	Masyarakat Disabilitas	Model *Machine Learning*
2	Sherrly Sugiono Sindarto, Dian Eka Ratnawati, Issa Arwani[32], 2022	Klasifikasi Citra Sistem Bahasa Isyarat Bahasa Indonesia (SIBI) dengan Metode *Convolutional Neural Network* pada Perangkat Lunak berbasis Android	Aplikasi Penerjemah bahasa isyarat	Masyarakat Disabilitas	Aplikasi penerjemah Bahasa Isyarat

3	Ihsan Mudzakir, Toni Arifin[21], 2022	Klasifikasi Penggunaan Masker dengan *Convolutional Neural Network* Menggunakan Arsitektur MobileNetV2	Penggunaan Arsitektur MobileNetv 2	Masyarakat Umum	Model *Machine Learning*
4	Imam Fauzi Annur, Jumhurul Umami, Moch Nasheh Annafii, Niken Trisnaningru, Oddy Virgantara Putra[33], 2023	Klasifikasi Tingkat Keparahan Penyakit *Leafblast* Tanaman Padi Menggunakan MobileNet V2	Klasifikasi Tingkat Keparangan Penyakit Tanaman padi	Petani	Model *Machine Learning*

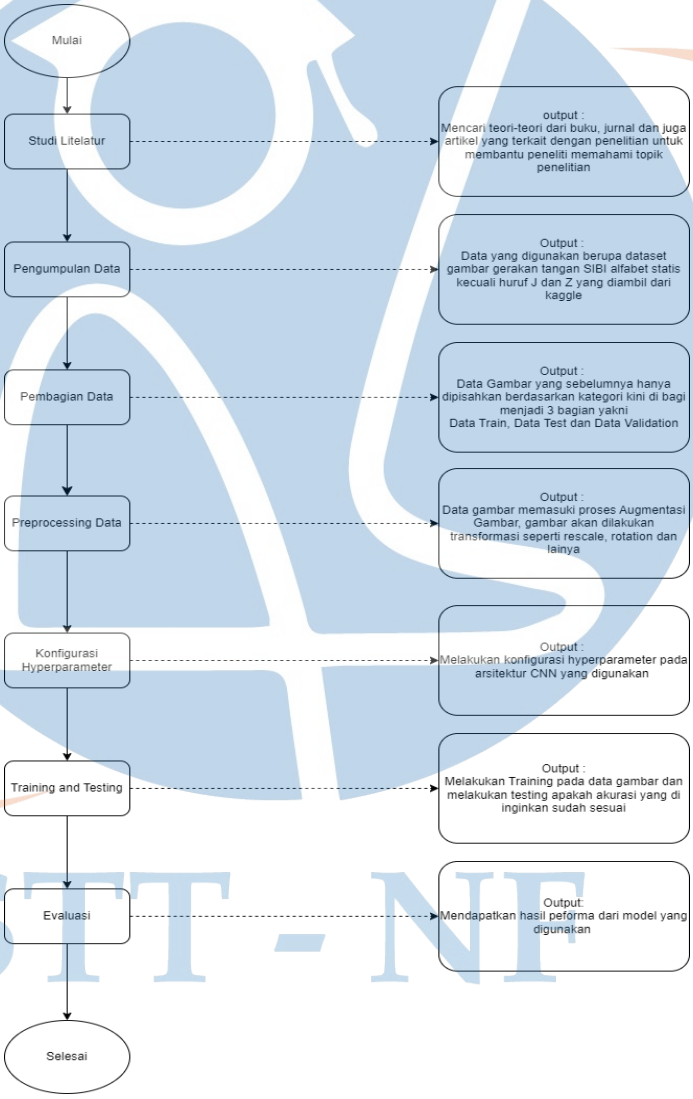
1. Pada penelitian yang berjudul "Model Penerjemah Bahasa Isyarat Indonesia (BISINDO) menggunakan pendekatan *Transfer Learning*" yang dibuat pada tahun 2022 menyimpulkan bahwa penggunaan MobileNetV1 dalam proses pembuatan model yang mampu mendeteksi bahasa isyarat memiliki akurasi 98% akan tetapi memiliki waktu yang relatif lebih lama dalam mengelolanya jika dibandingkan dengan model VGG16. Hasil tersebut didapat saat sudah melakukan proses augmentasi gambar *resize*[31].
2. Penelitian yang berjudul "Klasifikasi Citra Sistem Bahasa Isyarat Bahasa Indonesia (SIBI) dengan Metode *Convolutional Neural Network* pada Perangkat Lunak berbasis Android" yang di lakukan pada tahun 2022 menyatakan dalam penelitiannya bahwa dalam proses pengembangan model yang dapat mendeteksi bahasa isyarat SIBI mendapatkan akurasi 88% untuk 15 kelas menggunakan arsitektur *EfficenDet-Lite4* dengan dasar CNN [32].

3. Hasil kesimpulan yang dijelaskan pada penelitian yang berjudul "Penggunaan Masker dengan *Convolutional Neural Network* Menggunakan Arsitektur MobileNetv2" adalah penggunaan model CNN dengan arsitektur Mobile NetV2 berhasil melakukan deteksi objek dan mendapatkan hasil yang baik. Dan hasil dari proses *training* tidak menunjukkan bahwa model yang dibuat *overfitting* [34].
4. Penelitian yang dilakukan pada tahun 2023 yang berjudul "Klasifikasi Tingkat Keparahan Penyakit *Leafblast* Tanaman Padi Menggunakan MobileNetV2" memberikan kesimpulan bahwa menggunakan MobileNetV2 untuk melakukan klasifikasi pada tingkat keparahan penyakit *leafblast* pada tanaman Padi memiliki akurasi yang cukup baik yakni sebesar 78% [33].
5. Dari penelitian yang dijelaskan sebelumnya posisi penelitian yang saat ini sedang dilakukan memiliki kemiripan dalam penggunaan arsitektur model dengan penelitian pertama, ketiga dan keempat dikarenakan memiliki kesamaan arsitektur yakni MobileNet dan MobileNetV2 terutama dengan penelitian pertama karena berkaitan dengan pendeteksian bahasa isyarat SIBI. Namun tetap memiliki perbedaan yakni dimana jika pada penelitian pertama menggunakan MobileNetV1 pada proses pembuatan model pendeteksian gerakan bahasa isyarat maka pada penelitian ini menggunakan arsitektur MobileNetV2. Perbedaan lain dengan penelitian terkait yakni mengenai objek pendeteksian yang berbeda namun penggunaan arsitektur MobileNetV2 dan menggunakan algoritma CNN.

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian secara umum dapat dilihat pada alur penelitian yang akan dilakukan oleh peneliti pada gambar 3.1 sebagai berikut.



Gambar 3. 1 Alur Penelitian

Berikut ini penjelasan dari alur penelitian yang akan dilakukan oleh peneliti :

1. Studi Literatur

   Yang dilakukan oleh peneliti pada tahap ini adalah, mengumpulkan sumber-sumber berupa jurnal, buku dan artikel terkait dengan penelitian yang sedang dibuat. Dari hal tersebut didapatkan teori-teori yang digunakan untuk membantu memahami topik yang sedang diteliti.

2. Pengumpulan Data

   Pada tahap ini peneliti mengumpulkan data yang akan digunakan dalam penelitian. Data yang digunakan adalah data gambar gerakan tangan bahasa isyarat berjenis SIBI alfabet statis namun terkecuali dua huruf yakni J dan Z karena tidak bersifat statis yang didapat dari *website* Kaggle.

3. Pembagian Data

   Data yang akan digunakan sebelumnya harus di *split* yakni dibagi menjadi tiga bagian yakni data *training* yang digunakan untuk melatih model agar model dapat mempelajari pola dari data training yang disediakan, selanjutnya data *testing* yang digunakan untuk mengevaluasi atau mengetes kinerja pada model dengan menggunakan data yang benar-benar baru dan terakhir data *validation* yang digunakan juga selama proses training yang digunakan untuk mengoptimalkan model selama proses training. Hal tersebut perlu dilakukan dikarenakan data akan diproses ke dalam *deep learning*.

4. Preprocessing Data

   Pada tahap ini data gambar memasuki proses Augmentasi gambar untuk menciptakan variasi baru dari data gambar dengan transformasi sederhana hal yang dilakukan dalam proses ini seperti rotasi, pergeseran, perbesaran, membalik gambar, memotong gambar. Hal ini dilakukan untuk membantu dalam mengurangi terjadinya *overfitting* pada model.

5. Konfigurasi *Hyperparameter*

   Pada proses ini konfigurasi *hyperparameter* adalah menentukan nilai yang optimal untuk parameter yang mempengaruhi kinerja pada model *Deep learning*, seperti menentukan nilai *batch size*, jumlah *epoch*, menentukan *activation dan* juga *optimizer*.

6. Proses *Training* dan *Testing*

   Pada tahap ini peneliti menggunakan algoritma CNN model arsitektur MobileNet V2 dengan metode *transfer learning*, agar proses *training* dapat dilakukan dengan cepat dan ringan dibandingkan menggunakan CNN klasik. Berikutnya tahap *testing* peneliti menguji menggunakan data baru dan mencoba melakukan *testing* apakah model dapat mendeteksi data gambar baru yang di *input* dan melihat apakah hasil deteksinya sudah sesuai atau masih salah dalam mendeteksi gambar.

7. Evaluasi

   Pada tahap ini peneliti akan melakukan evaluasi pada model yang sudah dibuat, dengan menggunakan matriks evaluasi agar model tersebut bisa dinyatakan akurat dan tervalidasi. Matriks yang digunakan adalah *Accuracy*.

3.2 Rancangan Penelitian

3.2.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini adalah *research and development* dikarenakan dalam prosesnya penelitian ini mengembangkan suatu model *Deep learning* yang mampu melakukan pendeteksian bahasa isyarat SIBI menggunakan algoritma yang sudah ada dan bisa langsung digunakan yakni CNN dengan arsitektur MobileNet V2 dengan metode

transfer learning. *Output* dari penelitian ini adalah model *Deep learning* dengan *accuracy* dan mampu mendeteksi bahasa isyarat dengan baik.

3.2.2 Metode Analisis Data

Metode yang digunakan dalam penelitian ini adalah kuantitatif yang dimana data yang digunakan yakni data gambar gerakan tangan bahasa isyarat SIBI di proses secara matematis menggunakan algoritma CNN dengan arsitektur MobileNetV2 dan metode *transfer learning*. Data gambar yang digunakan merupakan data angka jika dilakukan pengkodean gambar karena setiap piksel yang ada di dalam gambar terdapat angka yang merepresentasikan kecerahan dan juga warna pada kombinasi nilai 3 warna yakni *red*/merah, *green*/hijau dan *blue*/biru dengan angka intensitasnya berada diantara 0-255. Gambar tersebut dimasukan kedalam proses *convolutional layer* yang didalamnya terdiri dari jaringan neuron yang disusun sehingga terbentuk filter dengan tinggi dan panjang. Namun proses pengolahan data ini juga menggunakan arsitektur dari algoritma CNN yakni MobileNetV2. Dengan menggunakan arsitekur yang sebelumnya sudah dilatih dengan jutaan gambar dan juga metode *transfer learning* maka proses menghasilkan model yang mampu mendeteksi bisa lebih cepat.

3.2.3 Metode Pengumpulan Data

Metode pengumpulan data pada penelitian ini menggunakan 2 cara yang diantaranya sebagai berikut:

1. Studi Literatur

   Studi Literatur merupakan yang akan dilakukan oleh peneliti dalam penelitian ini adalah mencari dan mengumpulkan sumber dari jurnal, buku dan juga artikel di internet yang sesuai dengan topik penelitian. Hasil dari pencarian dan pengumpulan studi literatur ini akan membantu peneliti menemukan informasi

mengenai teori dan metode yang dapat membantu dalam proses pengembangan serta penulisan pada penelitian ini.

2. Data Sekunder

Data sekunder yang digunakan merupakan data yang pengumpulan datanya dilakukan oleh pihak lain. Pada penelitian ini peneliti menggunakan data gambar gerakan tangan bahasa isyarat SIBI alfabet dengan masing-masing huruf berjumlah 220 gambar yang bersumber dari situs kaggle. Sumber data tersebut dapat diakses melalui link berikut ini:

https://www.kaggle.com/datasets/alvinbintang/sibi-dataset

Gambar 3. 2 Kaggle *Dataset*

Sumber : https://www.kaggle.com/datasets/alvinbintang/sibi-dataset

3.2.4 Metode Pengujian

Dalam penelitian ini metode pengujian yang akan digunakan adalah *statistical testing* dengan melihat tingkat persen keberhasilan model mendeteksi objek gambar gerakan tangan bahasa isyarat SIBI. Dalam prosesnya menggunakan matriks akurasi dengan menghitung jumlah prediksi yang benar lalu dibagi dengan jumlah keseluruhan data dengan persamaan sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

Prosesnya menghitung akurasi ini akan dilakukan setelah model selesai di *training*, dengan *data validation*, menggunakan *confusion matrix* dengan bantuan heatmap lalu setelahnya dihitung menggunakan rumus *accuracy*

3.2.5 Metode Implementasi dan Evaluasi

Dalam proses ini dilakukan tahapan implementasi pembuatan model *Deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat SIBI. Dataset yang digunakan dalam penelitian ini diambil dari https://www.kaggle.com/datasets/alvinbintang/sibi-dataset yang berasal dari website kaggle, yang selanjutnya akan dilakukan *training* menggunakan algoritma CNN dengan arsitektur MobileNet V2 dan *transfer learning*. Dalam tahapan ini dilakukan beberapa tahapan sebelumnya seperti *splitting dataset*, *augmentation image*, konfigurasi *hyperparameter* dan terakhir proses training dan testing model. Pada tahapan evaluasi, model yang sudah dibuat diuji menggunakan matriks *accuracy* dan jika model masih kurang baik maka dilakukan lagi proses konfigurasi namun jika model sudah baik maka akan dilanjutkan ke tahap pembuatan kesimpulan.

3.2.6 Lingkungan Pengembangan

Dalam penelitian ini peneliti menggunakan beberapa *tools* yang terdiri dari dua bagian yakni perangkat keras dan perangkat lunak yakni sebagai berikut:

1. Perangkat Keras
    a. Model : MSI Modern 14
    b. CPU : Core(TM) i5-1155G7 11th Gen Intel(R) @ 2.50GHz
    c. Ram : 8192 MB
    d. GPU : Intel(R) Iris(R) Xe Graphics
    e. Memory : SSD 512 GB

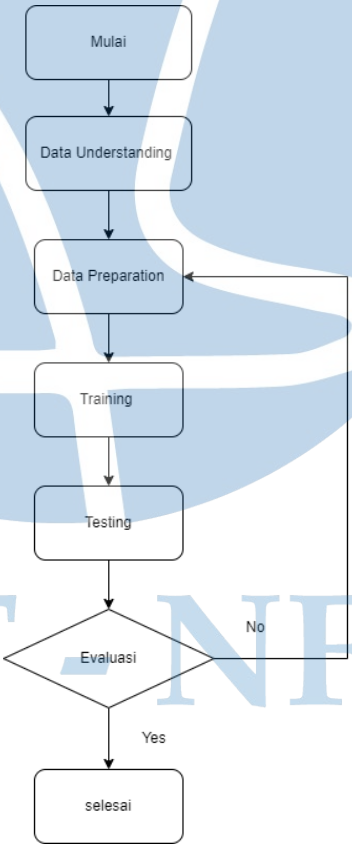
2. Perangkat Lunak
    a. Google Collab
    b. Visual Studio Code

BAB IV
IMPLEMENTASI DAN EVALUASI

Bagian ini akan berisikan pembahasan serta penjelasan mengenai implementasi dan evaluasi. Perancangan dalam pembangunan model *Deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat SIBI akan dijelaskan disini.

4.1 Analisis dan Perancangan Model *Deep Learning*

Pada bagian ini akan ditampilkan dan dijelaskan alur perancangan dalam pembuatan model *Deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat menggunakan CNN arsitektur MobileNetV2 dengan metode *transfer learning*. Berikut adalah alur perancangan model:

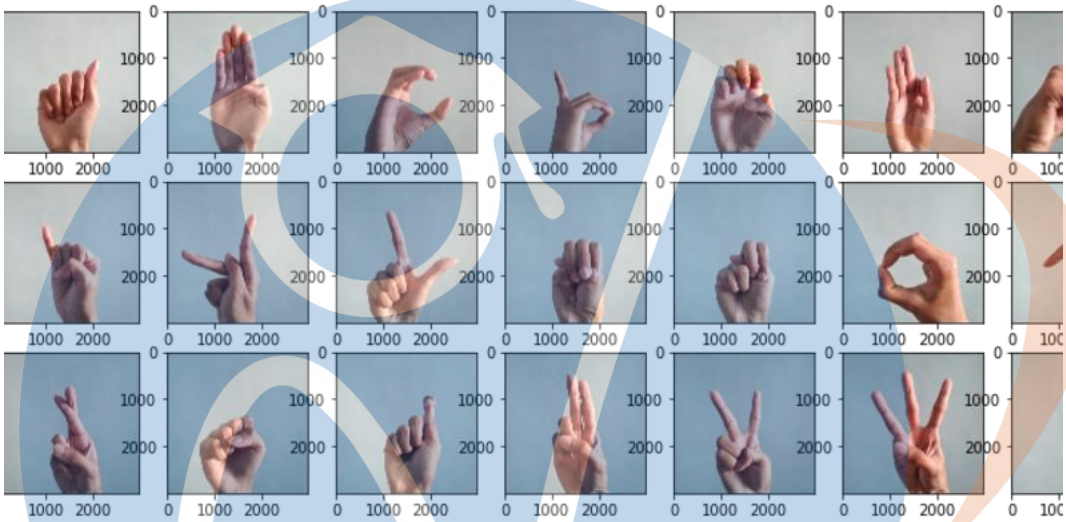


Gambar 4. 1 Alur Perancangan Model

1. Tahap yang pertama dilakukan adalah Data *Understanding* pada tahap ini hal yang dilakukan adalah untuk melihat dan mengetahui *dataset* yang sudah dikumpulkan sebelum *dataset* tersebut digunakan. *Dataset* yang digunakan dalam penelitian adalah data gambar gerakan tangan bahasa isyarat SIBI yang didapat dari situs penyedia *dataset* kaggle.
2. Lalu pada tahap selanjutnya proses yang dilakukan adalah *data preparation*. Sebelum data masuk ke proses pembuatan model data perlu dilakukan *preparation* agar *dataset* yang digunakan siap untuk digunakan. Pada proses ini akan dilakukan beberapa proses pada data yang digunakan seperti *split dataset* dan *augmented image*.
3. Selanjutnya adalah proses *training*, pada proses ini *dataset* memasuki proses pelatihan yang dimana algoritma yang digunakan adalah CNN berarsitektur MobileNetV2 menggunakan *transfer learning* agar proses dalam *training*, data bisa lebih cepat dari pada menggunakan CNN secara langsung.
4. Proses selanjutnya adalah *testing*. Pada proses ini model yang sudah dibuat dilakukan proses pengetesan dengan cara memasukan sebuah gambar untuk melakukan uji coba apakah gambar dapat dideteksi oleh model tepat atau salah.
5. Setelah melakukan proses testing hal yang dilakukan selanjutnya adalah melakukan evaluasi. Pada tahap ini model yang sudah jadi dilakukan uji validasi menggunakan *confusion matrix accuracy*, untuk menentukan apakah model dapat digunakan atau tidak, proses ini juga berkaitan dengan *testing*. Proses ini memiliki dua luaran jalur yakni "*No*" dan "*Yes*" sehingga jika pada hasil *confusion matrix* nilai akurasi rendah atau pada *testing* model tidak dapat mendeteksi dengan tepat harus kembali lagi ketahap *preparation "No"*. Namun jika hasil *testing* dan juga nilai pada matriks akurasi sudah bagus maka dapat proses dapat dilanjutkan hingga selesai "*Yes*".

4.2 Implementasi dan Evaluasi Model *Deep Learning*

4.2.1 Data *Understanding*



Gambar 4. 2 Cover *Dataset* SIBI

Sumber : https://kaggle.com

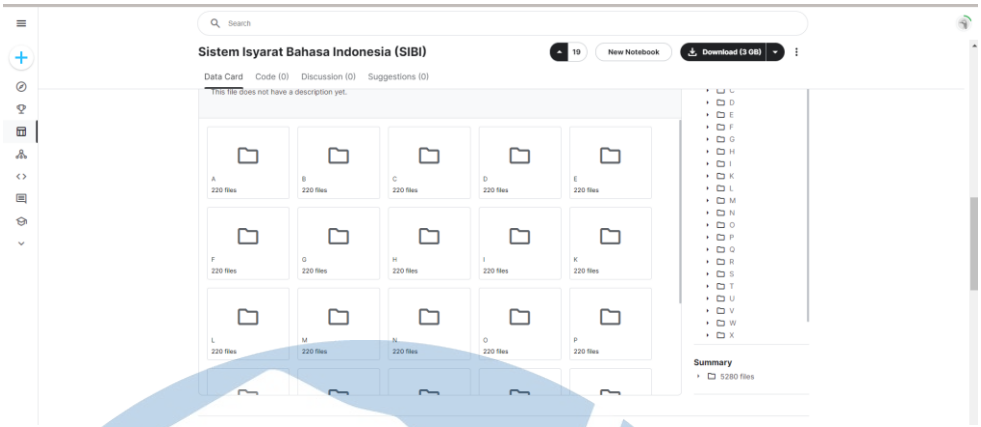
Pada penelitian ini, data yang dikumpulkan merupakan data gambar gerakan tangan bahasa isyarat SIBI. Data gambar ini didapatkan dari kaggle. Data tersebut berisikan gambar gerakan tangan bahasa isyarat seluruh huruf alphabet kecuali J dan Z dikarenakan gerakan tangan bahasa isyarat huruf tersebut tidak statis.

Jumlah data yang digunakan dalam penelitian ini dapat dilihat pada tabel sebagai berikut:

Tabel 4. 1 Jumlah Data

Dataset Huruf Alfabet	Jumlah
A	220
B	220

C	220
D	220
F	220
G	220
H	220
I	220
K	220
L	220
M	220
N	220
O	220
P	220
Q	220
R	220
S	220
T	220
U	220
V	220
W	220
X	220
Y	220



Gambar 4. 3 Dataset gambar SIBI

Sumber : https://kaggle.com

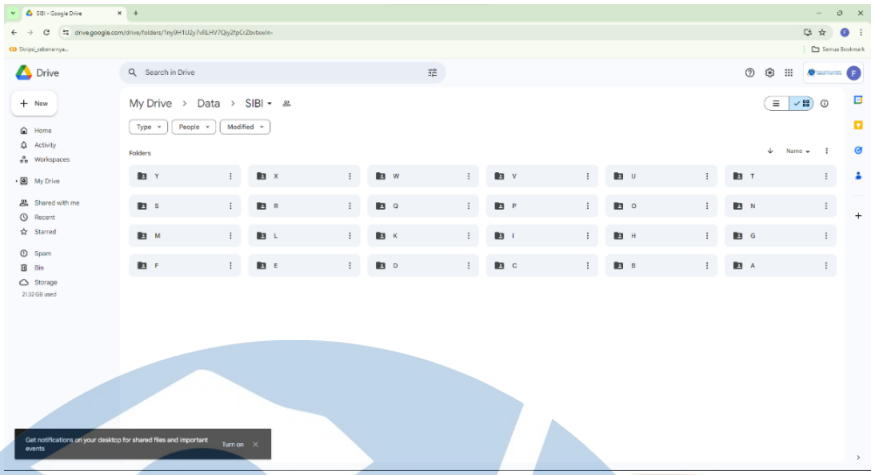
Pada setiap folder tersebut berisikan gambar dengan warna yang berbeda - berbeda. Ada gambar dengan *grayscale* dan ada gambar yang memiliki warna contohnya dapat dilihat pada gambar 4.1

Gambar 4. 4 contoh gambar SIBI huruf B berwarna dan grayscale

4.2.2 Data *Preparation*

4.2.2.1 *Splitting* Data

Sebelum data memasuki tahap modeling data gambar yang sudah di unduh masuk ke dalam proses selanjutnya yakni data *preparation*, namun sebelum dilakukan proses data *preparation*, data tersebut di upload terlebih dahulu kedalam google drive.



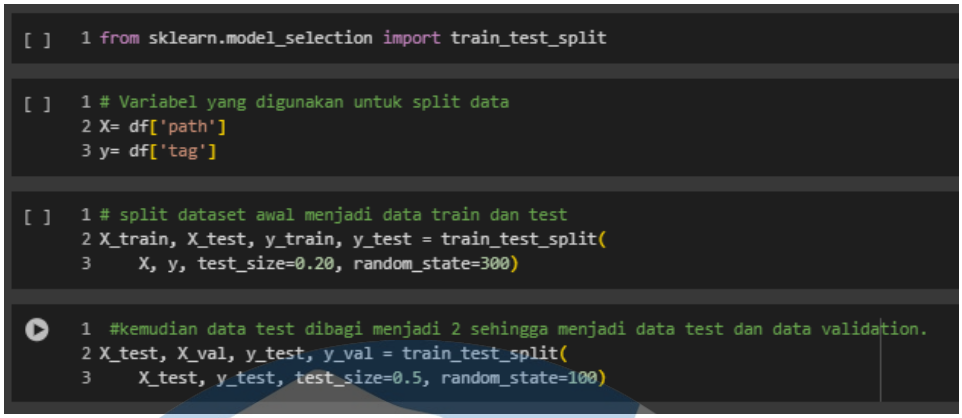
Gambar 4. 5 Folder *dataset* di google drive

Proses ini dilakukan agar mempermudah setiap proses pengerjaan, dikarenakan dalam penelitian ini menggunakan *tools* google collab yang bisa dengan mudah terintegrasi produk google yang lain termasuk google drive.

Tabel 4. 2 Menghubungkan google colab dengan google drive

```
drive.mount('/content/gdrive')
```

Selanjutnya masuk ke tahap berikutnya yakni pembagian *dataset*. *Dataset* harus dibagi terlebih dahulu menjadi 3 bagian yakni data *training*, data *testing* dan terakhir data *validation*. Proses pembagian data ini dilakukan menggunakan bahasa pemrograman python dan juga *library* sklearn.



```
[ ]  1 from sklearn.model_selection import train_test_split

[ ]  1 # Variabel yang digunakan untuk split data
     2 X= df['path']
     3 y= df['tag']

[ ]  1 # split dataset awal menjadi data train dan test
     2 X_train, X_test, y_train, y_test = train_test_split(
     3     X, y, test_size=0.20, random_state=300)

     1  #kemudian data test dibagi menjadi 2 sehingga menjadi data test dan data validation.
     2 X_test, X_val, y_test, y_val = train_test_split(
     3     X_test, y_test, test_size=0.5, random_state=100)
```

Gambar 4. 6 Proses Pembagian Data : *training,test,validation*

Pada gambar 4.6 *Dataset* gambar di bagi menjadi 3 bagian dengan 2 tahapan. Tahapan pertama yakni dataset dibagi menjadi 2 bagian terlebih dahulu yakni data *training* dan data *test* dengan persentase *training* sebesar 80% dan untuk data *test* sebesar 20%. Berikutnya tahap kedua untuk membuat data *validation* menggunakan data yang berasal dari data *testing* yang dibagi menjadi 2 bagian yakni data *testing* dan data *validation* dengan persentasemasing-masing yakni 50% dengan total persentase10%, berikut tabel proporsi dari pembagian data:

Tabel 4. 3 Proporsi Pembagian Data

No.	Jenis pembagian data	Presentase
1	Data *Training*	80%
2	Data *Testing*	10%
3	Data *Validation*	10%

.

Dari tabel diatas dapat diketahui bahwa pembagian untuk data *training* lebih besar dibanding dengan data *testing* dan *validation* yakni sebesar 80%. Hal ini dilakukan agar model dapat melakukan

pelatihan dengan data yang banyak sehingga memberikan hasil yang baik.

4.2.2.2 Data *Augmentation*

Data *Augmentation* merupakan proses yang dimana dengan menggunakan *dataset* yang ada, dapat membuat data baru dengan beberapa bentuk variasi agar pada proses *training* model dapat mengetahui lebih banyak keragaman data dan menghasilkan akurasi yang lebih tinggi. Pada proses augmentasi ini menggunakan metode augmentasi data di tempat atau *on the fly*. Hal tersebut dikarenakan menggunakan *library* keras *ImageDataGenerator*.

Tabel 4. 4 *Import ImageDataGenerator*

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Augmentasi yang dilakukan menggunakan *library ImageDataGenerator* tidak mengembalikan data yang asli dengan data yang diubah dan dengan penamaan *on the fly* karena augmentasi dilakukan pada waktu pelatihan. Proses augmentasi yang dilakukan pada data *training* dalam penelitian meliputi melakukan *rescale*, proses ini gambar akan di normalisasi nilai pikselnya dari rentang [0,255] menjadi [0,1]. Selanjutnya pada *shear*, proses ini menerapkan transformasi geser pada gambar dengan sudut hingga 20%. Berikutnya *rotation range* adalah memutar gambar dengan acak dalam rentang 0 sampai 25 derajat. Selanjutnya adalah *width shift* yang dimana prosesnya adalah menggeser gambar secara horizontal secara acak hingga 20% dari tinggi gambar. Untuk *height shift* memiliki kesamaan dengan *widh shift* namun pada *height shift*

gambar digeser secara vertikal. Selanjutnya proses zoom yakni melakukan zoom secara acak pada gambar dalam rentang 80% sampai 120% dari ukuran sebenarnya. Terakhir horizontal flip adalah proses membalik gambar secara horizontal secara acak. Untuk data *test* dan data *validation* hanya dilakukan *rescale* yakni untuk menormalisasi nilai piksel gambar. Code dalam proses augmentasi dapat dilihat pada gambar 4.7.

```
12
13 train_datagen = ImageDataGenerator(rescale=1. / 255,
14                                    shear_range=0.2,
15                                    rotation_range = 25,
16                                    width_shift_range = 0.2,
17                                    height_shift_range = 0.2,
18                                    zoom_range=0.2,
19                                    horizontal_flip=True)
20
21
22
23 validation_datagen = ImageDataGenerator(rescale=1. / 255)
24 test_datagen = ImageDataGenerator(rescale=1. / 255)
25
```

Gambar 4. 7 Proses Augmentasi

4.2.2.3 *Loading Data* dengan tensorflow

Pada tahap ini data yang sudah dilakukan proses augmentasi data, selanjutnya masuk ke dalam proses pemuatan data pada setiap masing-masing bagian data yang akan digunakan untuk klasifikasi gambar menggunakan tensorflow. Pada proses ini perlu mengatur beberapa *hyperparameter* seperti jumlah *batch size* yakni sebanyak 256, *image size* 224 x 224, jumlah kelas yang sesuai dengan banyak kelas yang ada pada data yakni sebanyak 24. Terakhir *class mode* yang di isi menggunakan *categorical* dikarenakan jumlah kelas yang akan diklasifikasi lebih dari dua.

```
# Loading data with tensorflow
train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

validation_generator = validation_datagen.flow_from_directory(
    validation_data_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)

Found 4224 images belonging to 24 classes.
Found 528 images belonging to 24 classes.
Found 528 images belonging to 24 classes.
```

Gambar 4. 8 Loading Data dengan tensorflow

Dapat dilihat ada proses yang ada pada gambar 4.8 dapat diketahui bahwa jumlah data pada setiap pembagian data adalah sebagai berikut:

Tabel 4. 5 Proporsi Data

Jenis Pembagian Data	Jumlah
Data *Training*	4224
Data *Testing*	528
Data *Validation*	528

4.2.3 *Training* Model

Pada tahap ini data yang dilakukan proses *preparation* selanjutnya masuk ke dalam proses *training* model. Pada penelitian ini algoritma yang digunakan dalam pembuatan model adalah CNN dengan arsitektur MobileNetV2 menggunakan metode *Transfer learning*. Dengan menggunakan model klasifikasi pra-terlatih atau yang sebelumnya sudah

dilatih menggunakan data yang sangat besar bersumber ImageNet. Model tersebut dapat di *import* dari keras API tensorflow dengan menggunakan potongan code berikut :

Tabel 4. 6 *Import* MobileNetV2

```
from tensorflow.keras.applications import MobileNetV2
```

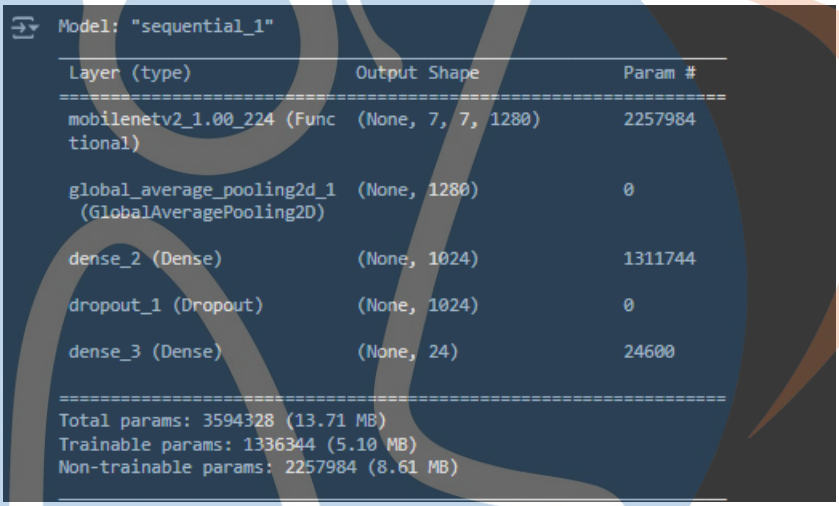
Setelah melakukan import model pra-latih MobileNet V2 selanjutnya adalah, membangung model langkah pertama dalam proses membangun model ini yang pertama yakni dengan menggunakan *include_top = false* yang berarti arsitektur yang digunakan tidak akan menyertakan bagian *fully connected*, karena bagian tersebut akan dibuat sendiri sesuai dengan kebutuhan penelitian ini. Selanjutnya adalah parameter *weight=imagenet* yang berarti model yang digunakan akan diinisialisasi dengan bobot yang sudah dilatih sebelumnya pada *dataset* yang sangat besar yakni imageNet. Berikutnya menentukan *input shape* = 224,224,3 yang berarti model yang dibuat akan menerima input dalam gambar 224x224 piksel dan angka 3 yang berarti dengan gambar berwarna. Terakhir model ini perlu dilakukan dibekukan agar tidak terjadi proses *training* kembali pada model yang sudah dilatih. Potongan code dalam membangun *base model* MobileNetV2 dapat dilihat pada gambar 4.9.

```
6 base_model = MobileNetV2(include_top=False,
7                          weights='imagenet',
8                          input_shape=(224, 224, 3))
9
10 for layer in base_model.layers:
11     layer.trainable = False
```

Gambar 4. 9 membuat *basemodel* MobileNetV2

Proses selanjutnya adalah membuat *dense layer* menggunakan *sequential* yang disediakan oleh tensorflow. Dalam *sequential* ini yang pertama di input adalah base model yang didalamnya sudah terdapat MobileNetV2, berikutnya *GlobalaveragePooling*2D yang digunakan untuk melakukan *pooling* rata-rata *global* pada *output* dari MobileNetV2. Selanjutnya

lapisan *dense* yang pertama setelah *base* model adalah 1024 dengan activation ReLu ini merupakan lapisan *fully connected layer* pertama setelah *base model*. Lalu berikutnya *dropout* dengan nilai 0.5 atau 50% hal ini dilakukan untuk mencegah *overfitting* pada model. Terakhir adalah lapisan *dense* terakhir yakni dengan nilai 24 sesuai dengan jumlah kelas yang ada pada *dataset* dan *activation softmax yang* digunakan sebagai klasifikasi terakhir. Kedua lapisan dense yang dibuat ini merupakan *fully conected layer* yang digunakan sebagai luaran akhir model. Berikut *summary* dari model yang sudah dibuat :



Gambar 4. 10 *Summary* model secara keseluruhan

Selanjutnya adalah melakukan konfigurasi dalam prosesnya dinamakan *model compile* dengan menggunakan fungsi *compile* yang ada pada *library* Keras, pada konfigurasi ini menentukan 3 parameter yakni *loss function*, *optimizer* dan terakhir adalah *metrics*. Pada penelitian ini Optimizer yang digunakan adalah Adam *(Adaptive Moment Estimation*). Penggunaan adam sebagai *optimizer* dikarenakan efisien secara komputasi dan dapat mencapai hasil yang baik dengan cepat. Selanjutnya *loss function* yang digunakan adalah *categorical crossentropy*, penggunaan *loss function* ini adalah dikarenakan *output* dari model adalah *single label*, dengan *multiple classes* dengan *final activation* adalah *softmax*. Terakhir matriks yang digunakan adalah akurasi.

Jumlah *epoch* yang ditentukan dalam proses *training* pada penelitian ini adalah sebanyak 50 *epoch* dengan total *batch*nya adalah 16.

4.2.4 *Testing Model*

Pada tahap ini model yang sudah dilakukan *training* masuk kedalam proses pengetesan. Proses ini menggunakan file h.5 yang selanjutnya menggunakan *framework* streamlit untuk membuat app browser sederhana untuk melakukan menampilkan hasil kinerja model pada gerakan tangan bahasa isyarat SIBI. Menggunakan jaringan *local*, tampilan streamlit dapat dilihat pada gambar 4.11

Gambar 4. 11 Streamlit *app browser* deteksi objek gerakan tangan bahasa isyarat SIBI

Pengetesan akan dilakukan sebanyak 3 kali dengan bentuk tangan yang berbeda dan gambar yang berbeda. Sumber gambar yang didapat dari 2 jenis yang pertama adalah sumber dari google dan yang kedua sumber dari kamera ponsel. Pertama akan menggunakan gambar dari google yang bersumber https://pmpk.kemdikbud.go.id/sibi/kosakata.

1. Pengetesan huruf sumber google

   Pada proses pengetesan model ini menggunakan gambar yang bersumber dari https://pmpk.kemdikbud.go.id/sibi/kosakata, pada pengetesan pertama, model dapat memprediksi bahwa gambar yang

di input adalah huruf B dengan tingkat keyakinan bahwa itu adalah huruf B sebesar 99,98%. Berikutnya untuk *input* kedua yakni berupa gambar gerakan tangan huruf L, model juga mampu mendeteksi bahwa inputan adalah huruf L dengan persentase keyakinan adalah 63,29%. Terakhir untuk *input* gambar SIBI huruf W, model dapat mendeteksi inputan yang masuk adalah huruf W dengan persentase keyakinan sebesar 99,17%. Hasil dari pengetesan dapat dilihat pada gambar 4.12.

Gambar 4. 12 Hasil *testing* deteksi pada gambar sumber kamus sibi

2. Pengetesan huruf SIBI Sumber ponsel

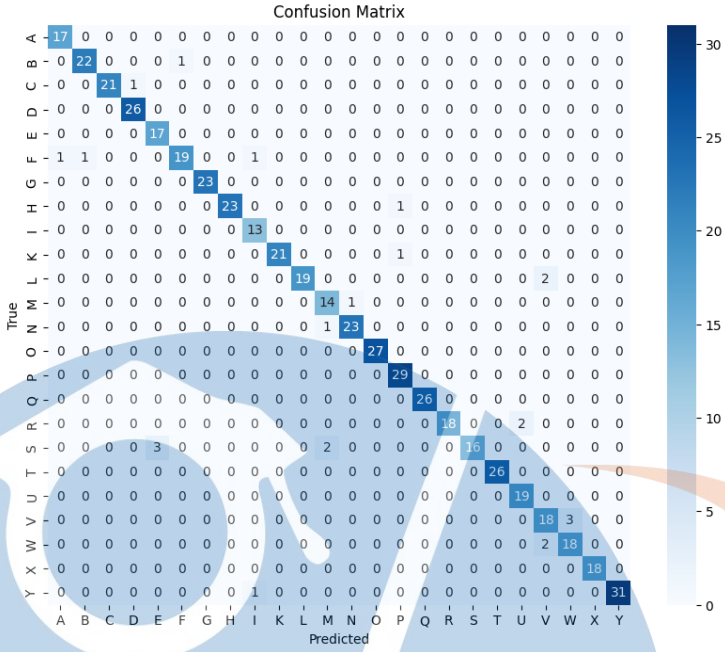
   Selanjutnya pada proses pengetesan menggunakan sumber dari ponsel. Untuk inputan gambar SIBI huruf B model dapat mendeteksi dengan persentase keyakinan sebesar 99,62%. Selanjutnya untuk inputan gambar huruf L, model dapat mendeteksi huruf dengan tingkat persentase sebesar 89,64%. Terakhir pada inputan gambar huruf W, model mampu mendeteksi dengan benar bahwa huruf yang diinput adalah huruf W dengan persentase keyakinan sebesar 99,91%. Untuk hasil lengkap bisa dilihat pada gambar 4.17

Gambar 4. 13 *Testing* model sumber gambar ponsel

4.2.5 Evaluasi Model

Pada tahap ini model yang sudah melalui proses pengetesan masuk ke dalam proses evaluasi. Proses ini merupakan tahapan yang dimana model diukur untuk menganalisis performanya dengan menggunakan matriks. Dalam penelitian ini menggunakan *confusion matrix* dengan metode matriks *accuracy*. B

erikut *confusion matrix* dari model yang sudah melalui proses *training* yang divisualisasikan menggunakan heatmap dapat dilihat pada gambar 4.14.



Gambar 4. 14 *Heatmap Confusion Matrix* Model Penelitian

Pada gambar diatas terdapat heatmap yang menunjukan TP (*True Positive*) dan TN (*True Negative*) pada bagian kolom yang membentuk diagonal dengan jumlah keseluruhan adalah 504, selanjutnya FP (*False* Positif) dan FN (*False Negative*) yang ditampilkan pada kolom selain pada bagian diagonal dengan jumlah keseluruhan sama yakni 24. Menggunakan metode perhitungan akurasi maka hasil *confusion matrix* yang sudah didapat akan dihitung untuk menentukan seberapa besar akurasi yang dimiliki dari model yang sudah dilatih. Berikut perhitungan matriks akurasi :

$$Accuracy = \frac{\text{Jumlah Prediksi Benar}}{TP + TN + FP + FN} \qquad (4.\ 1)$$

$$= \frac{504}{528} = 0.9545$$

Maka akurasi yang di dapat adalah 0,9545 atau jika dipersentasekan maka nilainya menjadi 95,45%.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pembuatan aplikasi yang mampu mendeteksi gerakan tangan bahasa isyarat yang dapat menerjemahkan bahasa isyarat adalah suatu yang diperlukan bagi orang-orang disabilitas. Penggunaan aplikasi tersebut dapat membantu orang non disabilitas berkomunikasi dengan baik saat dengan orang disabilitas. Dalam pembuatan aplikasi pendeteksian ini diperlukan pembuatan model *deep learning* yang mampu mendeteksi bahasa isyarat SIBI. Penelitian ini yang berfokus pada pembuatan model *deep learning* yang mampu mendeteksi gerakan tangan Bahasa Isyarat SIBI mendapatkan hasil kesimpulan sebagai berikut :

1. Proses pembuatan model pendeteksi gerakan tangan bahasa Isyarat SIBI menggunakan algoritma CNN arsitektur MobilenetV2 dengan metode transfer learning dilakukan dengan beberapa tahapan yakni, mengunduh dataset, melakukan data understanding, data preparation, melakukan pembagian dataset menjadi training, testing dana validation, menambah variasi dengan augmentasi, memuat data kedalam tensorflow. Selanjutnya proses training dilakukan dan setelah selesai model di testing menggunakan streamlit untuk mendeteksi data baru dan dilakukan evaluasi dengan matriks akurasi.
2. Hasil evaluasi dari matriks akurasi yang digunakan dalam pembuatan model pendeteksi gerakan tangan bahasa isyarat SIBI ini didapatkan 95,45% yang merupakan akurasi yang sudah cukup tinggi.

5.2 Saran

Adapun saran yang dapat diberikan oleh peneliti untuk pengembangan pada penelitian selanjutnya adalah sebagai berikut:

1. Pada penelitian ini masih diperlukan penambahan data gerakan tangan lagi, agar bisa lebih bervariasi terutama agar dapat mendeteksi gerakan tangan dalam setiap kondisi, karena cahaya saat pengambilan gambar mempengaruhi hasil dari testing.
2. Jika ingin melakukan pengembangan dari penelitian ini dengan dilakukannya pembuatan aplikasi mobile maka disarankan menggunakan TFlite untuk mengkonversi model yang berformat .h5 menjadi file dengan format .tflite lalu selanjutnya melakukan implementasi model pada aplikasi android.

DAFTAR PUSTAKA

[1] "Pemerintah Penuhi Hak Penyandang Disabilitas Di Indonesia | Kementerian Koordinator Bidang Pembangunan Manusia Dan Kebudayaan." Accessed: Feb. 29, 2024. [Online]. Available: Https://Www.Kemenkopmk.Go.Id/Pemerintah-Penuhi-Hak-Penyandang-Disabilitas-Di-Indonesia

[2] N. Sman, "Pemerolehan Bahasa Penderita Tuna Rungu Dan Tuna Wicara (Kajian Pragmatik Pada Kosakata Dan Fonetis)," Vol. 1, No. 1, 2021.

[3] N. Wiranda And A. E. Putro, "Model Identifikasi Kata Ucapan Tuna Wicara," *IJEIS (Indonesian Journal Of Electronics And Instrumentation Systems)*, Vol. 9, No. 2, P. 131, Oct. 2019, Doi: 10.22146/Ijeis.47609.

[4] F. Nur Fauziah Kumala, A. Kamalia, S. Khorriyatul Khotimah, And U. Islam Negeri Sunan Ampel, "Gambaran Dukungan Sosial Keluarga Yang Memiliki Anak Tuna Rungu," Vol. 13, No. 1, Pp. 1–10, 2022.

[5] T. Mutiah And I. Albar, "Etika Komunikasi Dalam Menggunakan Media Sosial," *Global Komunika*, Vol. 1, No. 1, 2019.

[6] M. F. Supriadi, E. Rachmawati, And A. Arifianto, "Pembangunan Aplikasi Mobile Pengenalan Objek Untuk Pendidikan Anak Usia Dini," Vol. 8, No. 2, Pp. 357–364, 2021, Doi: 10.25126/Jtiik.202184363.

[7] S. Apendi And M. W. Paryasto, "Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector," *E-Proceeding Of Engineering*, Vol. 10, No. 1, P. 249, 2023.

[8] "Kamus SIBI." Accessed: Mar. 13, 2024. [Online]. Available: Https://Pmpk.Kemdikbud.Go.Id/Sibi/Profil

[9] R. Fatmawati, R. Asmara, Y. R. Prayogi, And R. Y. Hakkun, "Aplikasi Pembelajaran Sistem Isyarat Bahasa Indonesia (SIBI) Berbasis

Voice Menggunakan Opensibi," *Technomedia Journal (TMJ*, Vol. 7, No. 1, Pp. 22–39, 2022, Doi: 10.33050/Tmj.V7i1%20Juni.1690.

[10] J. Homepage, A. Roihan, P. Abas Sunarya, And A. S. Rafika, "IJCIT (Indonesian Journal On Computer And Information Technology) Pemanfaatan Machine Learning Dalam Berbagai Bidang: Review Paper," *IJCIT (Indonesian Journal On Computer And Information Technology)*, Vol. 5, No. 1, Pp. 75–82, 2019.

[11] R. Pradono Iswara, T. Informatika, F. Sains Dan Teknologi, U. Syarif Hidayatullah Jakarta, And S. Gotong Royong Jakarta, "Pengembangan Algoritma Unsupervised Learning Technique Pada Big Data Analysis Di Media Sosial Sebagai Media Promosi Online Bagi Masyarakat," *Jurnal Teknik Informatika*, Vol. 12, No. 1, 2019.

[12] E. Dwi Handayani, "Penerapan Metode Convolutional Neural Network Pada Face Recognition Untuk Mengetahui Ikan Dapat Di Konsumsi Atau Tidak," *JMI Jurnal Manajemen Informatika*, Vol. 15, No. 2, 2023.

[13] A. Raup, W. Ridwan, Y. Khoeriyah, Q. Yuliati Zaqiah, And U. Islam Negeri Sunan Gunung Djati Bandung, "Deep Learning Dan Penerapannya Dalam Pembelajaran," *Jurnal Ilmiah Ilmu Pendidikan*, Vol. 5, No. 9, 2022, [Online]. Available: Http://Jiip.Stkipyapisdompu.Ac.Id

[14] D. M. Wonohadidjojo, "Perbandingan Convolutional Neural Network Pada Transfer Learning Method Untuk Mengklasifikasikan Sel Darah Putih," *Ultimatics : Jurnal Teknik Informatika*, Vol. 13, No. 1, P. 51, 2021.

[15] A. Solihin, D. Iskandar Mulyana, And M. B. Yel, "Amat Solihin, Dadang Iskandar Mulyana, Mesra Betty Yel Klasifikasi Alat Musik Tradisional Papua Menggunakan Metode Transfer Learning Dan Data Augmentasi," *Jurnal Sistem Komputer & Kecerdasan Buatan* , Vol. 5, No. 2, 2022, Doi: Https://Doi.Org/10.47970/Siskom-Kb.V5i2.279.

[16] F. Nurona Cahya, N. Hardi, D. Riana, And S. Hadianti, "SISTEMASI: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," *SISTEMASI: Jurnal Sistem Informasi*, Vol. 10, No. 3, 2021, [Online]. Available: Http://Sistemasi.Ftik.Unisi.Ac.Id

[17] A. S. Riyadi, I. P. Wardhani, D. S. Widayati, And K. Kunci, "Klasifikasi Citra Anjing Dan Kucing Menggunakan Metode Convolutional Neural Network (Cnn)," *Universitas Gunadarma Jl. Margonda Raya No*, Vol. 5, No. 1, P. 12140, 2021.

[18] Y. Wang, Y. Li, Y. Song, And X. Rong, "The Influence Of The Activation Function In A Convolution Neural Network Model Of Facial Expression Recognition," *Applied Sciences (Switzerland)*, Vol. 10, No. 5, Mar. 2020, Doi: 10.3390/App10051897.

[19] L. Alzubaidi *Et Al.*, "Review Of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *J Big Data*, Vol. 8, No. 1, Dec. 2021, Doi: 10.1186/S40537-021-00444-8.

[20] W. Setiawan, "Perbandingan Arsitektur Convolutional Neural Network Untuk Klasifikasi Fundus," Vol. 7, No. 2, 2019.

[21] I. Mudzakir And T. Arifin, "Klasifikasi Penggunaan Masker Dengan Convolutional Neural Network Menggunakan Arsitektur Mobilenetv2," *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, Vol. 12, No. 1, P. 76, Jun. 2022, Doi: 10.36448/Expert.V12i1.2466.

[22] N. Nufus *Et Al.*, "Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD Mobilenet V2 Dengan Menggunakan Gambar 360° Ternormalisasi," *Prosiding Seminar Nasional Sains Teknologi Dan Inovasi Indonesia (SENASTINDO)*, Vol. 3, Pp. 123–134, Dec. 2021, Doi: 10.54706/Senastindo.V3.2021.123.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, And L.-C. Chen, “Mobilenetv2: Inverted Residuals And Linear Bottlenecks,” Jan. 2019, [Online]. Available: Http://Arxiv.Org/Abs/1801.04381

[24] T. Bayu Sasongko And A. Amrullah, “Analisis Efek Augmentasi Dataset Dan Fine Tune Pada Algoritma Pre-Trained Convolutional Neural Network (Cnn),” Vol. 10, No. 4, Pp. 763–768, 2023, Doi: 10.25126/Jtiik.2023106583.

[25] H. Suryalim, K. R. W. Retno, And H. Heryanto, “Analisis Optimizer Pada Convolutional Neural Network Untuk Meningkatkan Akurasi Pengenalan Wajah,” *Jurnal Edukasi & Penelitian Informatika*, Vol. 9, No. 2, 2023, Doi: Http://Dx.Doi.Org/10.26418/Jp.V9i2.60432.

[26] Karlina Surya Witanto, Ngurah Agus Sanjaya ER, AAIN Eka Karyawati, I Gusti Agung Gede Arya Kadyanana, I Ketut Gede Suhartana, And Luh Gede Astutia, “Implementasi LSTM Pada Analisis Sentimen Review Film,” *Jurnal Elektronik Ilmu Komputer Udayana*, Vol. 10, No. 4, 22AD.

[27] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, And E. A. Chavez-Urbiola, “Loss Functions And Metrics In Deep Learning,” Jul. 2023, [Online]. Available: Http://Arxiv.Org/Abs/2307.02694

[28] I. "Kristiyanti, A. D. "Saputra, *Machine Learning Untuk Pemula*. Bandung: INFORMATIKA, 2022.

[29] J. Arfianto And I. Muhimmah, “Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learningdengan Tensorflow,” *AUTOMATA*, Vol. 2, No. 2, 2021.

[30] W. Hastomo, N. Aini, A. Satyo, B. Karno, And L. M. R. Rere, “Metode Pembelajaran Mesin Untuk Memprediksi Emisi Manure Management,” *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi* |, Vol. 11, No. 2, 2022.

[31] M. B. Subkhi, M. Yuda Trinurais, R. Kuncoro, A. Wibowo, And B. R. Prakosa, "Deteksi Bahasa Isyarat Berdasarkan Sibi (Sistem Bahasa Isyarat Indonesia) Menggunakan Transfer Learning," *Prosiding Seminar Nasional Teknologi Dan Sains Tahun 2024*, Vol. 3, 2024.

[32] S. S. Sindarto, D. E. Ratnawati, And I. Arwani, "Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI) Dengan Metode Convolutional Neural Network Pada Perangkat Lunak Berbasis Android," 2022. [Online]. Available: Http://J-Ptiik.Ub.Ac.Id

[33] I. F. Annur, J. Umami, Moch. N. Annafii, N. Trisnaningrum, And O. V. Putra, "Klasifikasi Tingkat Keparahan Penyakit Leafblast Tanaman Padi Menggunakan Mobilenetv2," *Fountain Of Informatics Journal*, Vol. 8, No. 1, Pp. 7–14, May 2023, Doi: 10.21111/Fij.V8i1.9419.

[34] M. Farid Naufal And S. Ferdiana Kusuma, "Pendeteksi Citra Masker Wajah Menggunakan Cnn Dan Transfer Learning," Vol. 8, No. 6, Pp. 1293–1300, 2021, Doi: 10.25126/Jtiik.202185201.