



MODEL *DEEP LEARNING* UNTUK PENERJEMAH BAHASA ISYARAT SIBI DENGAN ARSITEKTUR *TRANSFER LEARNING*

Fikri Pratama Al Fajri¹, Ahmad Rio Adriansyah², Sirojul Munir³

^{1,2,3}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

Jakarta Selatan, DKI Jakarta, Indonesia 12640

fikripratamaal@gmail.com, arasy@nurulfikri.ac.id, rojulman@nurulfikri.ac.id

Abstract

People with speech and hearing disabilities have difficulties when communicating with non-disabled people because they use sign language which is rarely learned in general. To solve this problem, a deep learning model is needed that can detect sign language hand gestures. A sign language translator application can then be made that facilitates communication between non-disabled people and people with disabilities. This research aims to create a deep learning model that can detect SIBI alphabet-type sign language hand gestures with good accuracy. The CNN algorithm model uses Transfer Learning MobilenetV2 architecture and transfer learning method. The results of this study show that the model evaluation reaches 95.45% and the next model can be applied to the sign translator application, for further development it is expected to use more datasets so that the model gets a lot of variation during the training process.

Keywords: CNN, Deep Learning, MobileNetV2, SIBI, Transfer Learning

Abstrak

Orang-orang dengan disabilitas tuna wicara serta tunarungu memiliki kesulitan ketika berkomunikasi dengan orang non disabilitas dikarenakan mereka menggunakan bahasa isyarat yang jarang dipelajari secara umum. Untuk menyelesaikan permasalahan ini diperlukan suatu model *deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat yang selanjutnya dapat dibuat aplikasi penerjemah bahasa isyarat yang bisa mempermudah komunikasi orang non disabilitas dengan orang disabilitas. Tujuan dari penelitian ini adalah membuat model *deep learning* yang mampu mendeteksi gerakan tangan bahasa isyarat berjenis SIBI alfabet dengan akurasi yang baik. Model algoritma CNN menggunakan *Transfer Learning* arsitektur MobilenetV2 dan metode *transfer learning*. Hasil penelitian ini menunjukkan evaluasi model mencapai 95,45% dan model selanjutnya dapat diterapkan pada aplikasi penerjemah isyarat, untuk pengembangan lanjutan diharapkan penggunaan *dataset* yang lebih banyak agar model mendapatkan banyak variasi saat proses *training*.

Kata kunci: CNN, Deep Learning, MobileNetV2, SIBI, Transfer Learning

1. PENDAHULUAN

Disabilitas adalah kondisi di mana seseorang dengan kekurangan atau keterbatasan fisik, intelektual mental serta sensorik yang membuat penyandangannya memiliki keterbatasan dalam melakukan aktivitasnya. Jenis dari disabilitas ada beberapa macam jenis yang salah satunya biasanya bersamaan yakni tuna rungu yang merupakan gangguan pada pendengaran dan tuna wicara yang merupakan kondisi di mana seseorang tidak dapat berbicara. Kondisi ini dapat terjadi karena ketika seseorang mengalami tuna rungu sejak lahir maka dia tidak akan mengetahui bunyi dari huruf sehingga mereka tidak dapat mengucapkan huruf tersebut[1]. Menurut data yang bersumber dari Open

Data Jabar di Kota Depok dari tahun 2021 sampai 2022 terdapat 658 jiwa penyandang tuna wicara dan tuna rungu.

Orang-orang yang mengalami kondisi ini menggunakan bahasa isyarat sebagai cara utama berkomunikasi, namun bahasa isyarat bukan sesuatu yang umum dipelajari sehingga orang-orang non disabilitas akan kesulitan berkomunikasi dengan orang disabilitas tuna rungu dan tuna wicara.

Untuk mengatasi masalah itu maka diperlukan suatu aplikasi yang didalamnya terdapat model *deep learning* yang dapat mendeteksi gerakan tangan bahasa isyarat, terfokus kepada jenis bahasa isyarat SIBI (Sistem Bahasa

Isyarat Bahasa Indonesia) karena sudah dibakukan oleh pemerintah[2]. Penelitian ini berfokus kepada pembuatan model *deep learning* dengan menggunakan *framework tensorflow*. Menggunakan algoritma CNN (*Convolutional Neural Network*) dengan arsitektur MobileNetV2 dan metode *transfer learning* yang memungkinkan model dibuat dengan lebih cepat dan ringan dalam komputasi[3].

SIBI

SIBI (Sistem Isyarat Bahasa Indonesia) merupakan sebuah bahasa isyarat yang sudah dibakukan oleh pemerintah[4] serta digunakan dalam pembelajaran di Sekolah Luar Biasa (SLB). SIBI lebih sering digunakan dalam acara formal. Pada penerapan SIBI terdapat bentuk alfabet yang di mana setiap gerakan tangan memiliki arti huruf. Hampir semua huruf gerakan statis (hanya membentuk tidak bergerak) namun untuk J dan Z dinamis karena diperlukan untuk membentuk huruf tersebut.

Deep Learning

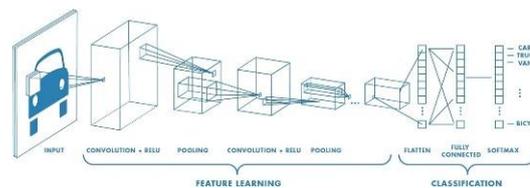
Deep Learning Merupakan bagian dari *machine learning* serta kecerdasan buatan yang di mana *deep learning* pengembangan dari *neural network multiple layer*[5]. Menggunakan metode *Artificial Neural Network (ANN)* yang merupakan model pembelajaran mesin yang *scalable*, kuat dan serbaguna karena algoritma ANN merupakan dasar dan metode sederhana yang digunakan pada pendekatan *deep learning*[6]. ANN yang biasa disebut jaringan syaraf tiruan yang mirip seperti neuron-neuron yang saling terhubung karena ANN memiliki banyak lapisan. *Deep learning* dalam pengembangannya memiliki banyak algoritma yaitu *Long Short Term Memory Network (LSTM)*, *Recurrent Neural Network (RNN)* dan *Convolutional Neural Network (CNN)*[7].

Transfer Learning

Transfer learning merupakan metode yang menggunakan *network* yang sudah dilatih dan siap digunakan untuk mempelajari tugas baru[8]. Penggunaan *transfer learning* memungkinkan model *deep learning* pada saat proses *training* mendapatkan akurasi yang tinggi dengan waktu yang lebih cepat saat proses *training*[9].

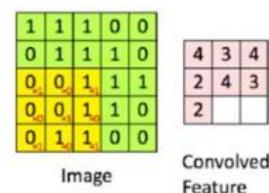
Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) merupakan pengembangan dari algoritma *Multilayer Perceptron (MLP)* yang di desain dapat melakukan pengolahan pada data dua dimensi[10]. Penggunaan algoritma CNN dalam pengembangan model *deep learning* cukup populer yang dimana dapat digunakan untuk pengolahan data dua dimensi gambar atau citra[11]. Ide pengembangan ini didasari dari tiga ide awal yakni *local receive field*, *weight sharing* dan *spatial sub sampling*. Ide tersebut dijelaskan masukan kedalam dua jenis *layer* yakni *pooling* dan *convolution layer*.



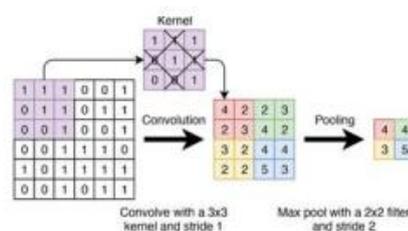
Gambar 1. Arsitektur CNN

Pada gambar 1, bagian utama dalam pengembangan CNN terdapat pada *convolutional layer* yang sebagian besar prosesnya adalah operasi matematika linear aljabar dengan mengalikan matriks dari filter ke gambar. Hal ini dilakukan dengan tujuan konvolusi pada gambar dapat mengekstraksi fitur dari gambar yang sudah di input. Bobot atau *weight* yang ada pada *layer* menyesuaikan dengan *kernel* konvolusi yang digunakan sehingga *kernel* konvolusi bisa dilatih dengan *input* pada CNN.



Gambar 2. Proses Konvolusi Layer

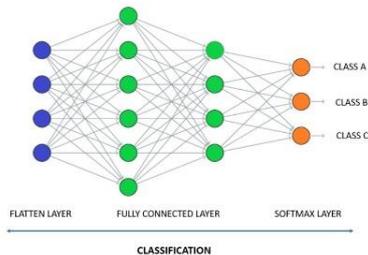
Berikutnya adalah bagian *pooling layer* yang pada prosesnya mereduksi ukuran dari data gambar dengan tujuan untuk meningkatkan invariasi posisi dari fitur serta membagi *output* dari *convolutional layer* menjadi bagian-bagian *grid* kecil dengan nilai maksimal dari setiap bagian *grid* yang selanjutnya digunakan untuk menyusun matriks gambar yang direduksi menggunakan operasi *max pooling*. Pada proses ini dapat memastikan fitur yang didapat akan sama walaupun objek mengalami pergeseran, ditunjukkan pada gambar 2 di atas.



Gambar 3. Operasi Max Pooling

Gambar 3 merupakan *pooling layer* yang pada umumnya akan mengikuti *layer* konvolusi yang digunakan untuk mengurangi dimensi pada *feature map* serta melakukan percepatan di dalam proses komputasi karena parameter yang diubah menjadi semakin sedikit. Prinsipnya *pooling layer* merupakan filter dengan *stride* dan ukuran yang akan bergeser pada seluruh area di *feature map*. Ini merupakan cara yang digunakan pada *layer max pooling* dan *average pooling*.

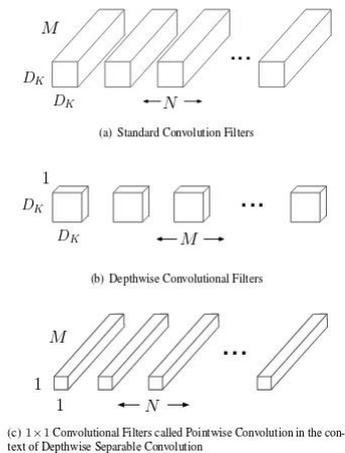
Gambar 4 di bawah merupakan *layer* terakhir yang diletakan pada akhir setiap model pada pembuatan model dengan CNN adalah *fully connected layer* yang didalamnya terhubung ke semua neuron yang ada di layer sebelumnya. Layer ini difungsikan untuk melakukan pengklasifikasian pada arsitektur CNN[10].



Gambar 4. Fully Connected Layer

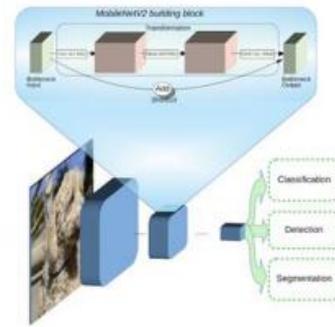
MobileNetV2

MobileNet merupakan salah satu arsitektur dari algoritma CNN yang model kecil yang memiliki latensi yang rendah daya yang diukur untuk memenuhi batasan dari sumber daya untuk berbagai macam kasus berbeda[12]. Perbedaan antara MobileNetV2 dengan CNN klasik terdapat pada bagi bagian lapisan konvolusi. Pada CNN menggunakan ketebalan filter yang sesuai pada *input* gambar sedangkan pada MobileNetV2 konvolusi terbagi menjadi *depthwise convolution* dan *pointwise convolution*.



Gambar 5. (a) Konvolusi standar pada CNN, (b) dibagi menjadi dua lapisan: *depthwise convolution* dan *pointwise convolution*, (c) untuk membuat filter terpisah secara mendalam (*depthwise*)

Pada gambar 5 MobileNetv2 hadir dengan peningkatan kinerja yang mampu membangun model seluler dengan lebih efisien[13], sesuai dengan penggunaan kata *mobile* yang ditunjukan kepada pengaplikasiannya pada ponsel atau *mobile*[14]. Berdasarkan kesamaan dengan versi sebelumnya yakni MobileNetV1 *depthwise convolution* dan *pointwise convolution*, MobileNetV2 memiliki 2 fitur tambahan yakni *linear bottleneck* dan *shortcut connections* antar *bottleneck* pada struktur dasar. Arsitektur MobileNetV2 dapat dilihat pada gambar 6.



Gambar 6. Bottleneck yang Terdapat dalam Arsitektur MobileNet V2

Pada bagian *bottleneck*, terdapat *input* dan *output* antar model, sementara bagian dalam *layer* mengenkapsulasi kemampuan dari model untuk mengubah *input* dari konsep tingkat tinggi ke tingkat yang lebih rendah.

Optimizer

Optimizer adalah suatu fungsi dalam matematika yang memiliki ketergantungan pada variabel bebas dari model yakni bias dan nilai bobot. *Optimizer* juga merupakan metode yang sering digunakan untuk meminimalisir nilai *output* dari *cost function*. Cara kerjanya adalah dengan mempelajari bagaimana caranya mengubah biasa dan nilai bobot didalam *neural network* agar dapat mengurangi *error*. Jenis *optimizer* ada beragam yang salah satunya adalah Adam (*Adaptive Moment Estimation*) yang merupakan *optimizer* yang populer dalam pengembangan model *deep learning*. Adam memiliki keuntungan karena merupakan penggabungan dari Ekstensi *optimizer* SGD yakni AdaGard dan RMS. Hal tersebut membuat penggunaan adam dapat memberikan pengoptimalan algoritma yang bisa mengatasi *sparse gradient* pada *noisy problem*[15].

Categorical Cross Entropy (CCE)

Categorical Cross Entropy (CCE) adalah suatu *loss function* yang digunakan untuk membuat model klasifikasi dengan banyak kelas. CCE melakukan pengukuran perbedaan pada distribusi probabilitas yang diprediksi menggunakan distribusi sebenarnya. Distribusi probabilitas yang diprediksi, didefinisikan sebagai rata-rata kemungkinan pada log negatif dari kelas sebenarnya, rumus CCE didefinisikan sebagai berikut:

$$L = - \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N y_{i,j} \log(P_{i,j})$$

2. METODE PENELITIAN

2.1. Metode Pengumpulan Data

Metode yang digunakan dalam penelitian ini adalah dengan menggunakan 2 cara yaitu :

2.1.1. Studi Literatur

Proses ini dilakukan oleh peneliti untuk mencari serta mengumpulkan sumber dari jurnal, buku dan juga artikel di internet yang sesuai dengan topik penelitian. Hasil dari proses ini akan membantu peneliti menemukan informasi

mengenai teori serta metode yang bisa membantu dalam proses perancangan serta penulisan di penelitian ini.

2.1.2. Data Sekunder

Data sekunder adalah data yang proses pengumpulannya dilakukan oleh orang lain. Penelitian ini menggunakan data gambar gerakan tangan bahasa isyarat SIBI alfabet dengan masing-masing huruf sebanyak 220 gambar. Yang bersumber dari Kaggle dengan *link* yang dapat diakses sebagai berikut:

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

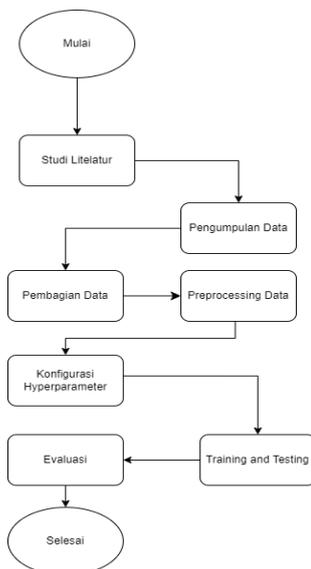
2.2. Metode Pengujian

Dalam penelitian ini pengujian ini menggunakan *statistical testing* dengan melihat tingkat persen keberhasilan model mendeteksi objek gerakan tangan bahasa isyarat SIBI. Proses menggunakan *confusion matrix* dengan matriks *accuracy* dengan cara menghitung jumlah prediksi yang benar lalu dibagi dengan jumlah keseluruhan data dengan persamaan berikut ini.

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

2.3. Tahapan Penelitian

Penelitian Perancangan Model *Deep Learning* untuk Penerjemah Bahasa Isyarat SIBI dengan Arsitektur *Transfer Learning* MobileNetV2 ini memiliki alur tahapan penelitian yang akan dilakukan pada gambar 7 sebagai berikut.



Gambar 7. Alur Penelitian

2.3.1. Studi Literatur

Pada tahap ini peneliti mengumpulkan sumber dari jurnal, buku dan artikel terkait dengan penelitian yang sedang dibuat.

2.3.2. Pengumpulan Data

Pada penelitian ini pengumpulan data dilakukan dengan menggunakan *dataset* yang bersumber dari Kaggle. Data yang digunakan adalah gambar gerakan tangan bahasa isyarat berjenis SIBI alfabet statis, akan tetapi pada data ini

tidak terdapat huruf J dan Z dikarenakan huruf tersebut tidak bersifat statis.

2.3.3. Pembagian Data

Data yang akan digunakan harus melalui proses *split* yaitu pembagian *dataset* menjadi tiga bagian, *data training*, *data testing* dan *data validation*.

2.3.4. Preprocessing Data

Pada proses ini data gambar yang sudah di *split* memasuki proses augmentasi gambar agar dapat menghasilkan variasi baru pada data dengan transformasi sederhana. Hal ini dilakukan agar mengurangi terjadinya model yang *overfitting*.

2.3.5. Proses Training dan Testing

Pada tahap ini data yang sudah siap latih yakni data *training* akan dilatih menggunakan algoritma CNN dengan model arsitektur MobileNetV2 dengan metode *transfer learning*. Hal ini dilakukan agar proses *training* bisa menjadi lebih cepat dan ringan.

2.3.6. Evaluasi

Terakhir adalah evaluasi dengan menggunakan matriks evaluasi agar model dapat dinyatakan akurat dan tervalidasi dengan matriks yang dipakai adalah *accuracy*.

3. HASIL DAN PEMBAHASAN

3.1. Data Understanding

Penelitian ini menggunakan *dataset* gambar gerakan tangan bahasa isyarat SIBI yang merupakan data sekunder yang didapat dari Kaggle dengan seluruh alfabet ada di dalamnya kecuali J dan Z dikarenakan gerakan tangan tersebut tidak bersifat statis. Pada setiap huruf terdapat dua jenis gambar yang berbeda yakni yang pertama adalah *grayscale* dan berwarna. Contoh dapat dilihat pada gambar 8.



Gambar 8. Contoh Gambar SIBI Huruf B Berwarna dan *Grayscale*

3.2. Splitting Data

Proses ini adalah pembagian data menjadi 3 bagian yakni data *training*, *testing*, dan *validation*. Proporsi yang digunakan dalam *splitting* data penelitian ini dapat dilihat pada tabel 1.

Tabel 1. Persentase Pembagian Data

Jenis Pembagian Data	Persentase
<i>Data Training</i>	80%
<i>Data Testing</i>	10%
<i>Data Validation</i>	10%

Dapat diketahui dari tabel mengenai pembagian data, data *training* memiliki persentase yang lebih besar dibandingkan dengan *testing* dan *validation* yaitu sebesar 80%. Ini

dilakukan agar model bisa melakukan proses pelatihan dengan data *training* yang banyak sehingga memberikan hasil yang baik.

3.3. Data Augmentation

Proses *data augmentation* atau augmentasi dilakukan agar *dataset* memiliki beberapa variasi bentuk agar pada saat proses *training* model dapat mempelajari keberagaman data dan dapat mendapatkan akurasi yang lebih tinggi. Proses augmentasi yang digunakan adalah *on the fly* dikarenakan penggunaan *library* keras yakni *ImageDataGenerator*. Dengan penggunaan *ImageDataGenerator* proses augmentasi terjadi pada saat proses *training* menggunakan data *training* proses ini menghasilkan variasi yang dilakukan adalah *rescale*, *shear*, *rotation range*, *width* dan *height shift*, *zoom* serta terakhir *horizontal flip*.

3.4. Loading Data dengan Tensorflow

Proses ini adalah pemuatan data, pada masing-masing bagian yang akan digunakan untuk klasifikasi gambar menggunakan *tensorflow*. Pada proses ini juga mengatur beberapa *hyperparameter* yakni jumlah *batch size* sebanyak 256, *image size* sebesar 224x224 dan jumlah kelas sebanyak jumlah kelas pada data yakni 24 serta terakhir mengatur *class mode* dengan menggunakan *categorical* karena jumlah kelas yang akan dilakukan proses klasifikasi lebih dari dua. Dengan demikian dapat diketahui jumlah pembagian data yang dapat dilihat pada tabel 2.

Tabel 2. Jumlah Data Load

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

3.5. Training Model

Pada proses *training* model pada penelitian ini menggunakan algoritma CNN namun dikarenakan menggunakan metode *transfer learning* maka menggunakan model klasifikasi pra-terlatih atau yang sebelumnya sudah dilatih dengan menggunakan data super besar yang bersumber dari ImageNet, ini juga dikarenakan arsitektur yang digunakan adalah MobileNetV2.

Model tersebut perlu di *import* menggunakan dari keras API *tensorflow*. Tahap ini merupakan tahap awal membuat model dasar dengan MobileNetV2 dengan input gambar adalah 224x224 piksel dengan menggunakan warna RGB pada gambar. Penggunaan MobileNetV2 memuat banyak *block* dengan banyak *layer*. Selanjutnya setelah pembuatan *base model*, *base model* tersebut harus di *freeze* agar tidak melakukan pelatihan kembali dan tetap mempertahankan fitur yang dimilikinya tanpa mengubah bobot pelatihannya.

Selanjutnya adalah pembuatan *dense layer* menggunakan *sequential* dari *tensorflow*. Hal yang dilakukan adalah melakukan penginputan *base_model* yang berisikan MobileNetV2, berikutnya menggunakan

GlobalAveragePooling2D yang dilakukan untuk melakukan *pooling* rata-rata hasil luaran *base_model*. Berikutnya *dense layer* 1024 dengan *activation* Relu, ini merupakan *dense* pertama setelah *base* model dan juga lapisan *fully connected layer* pertama, selanjutnya *dropout* agar dapat mengurangi *overfitting* pada model dan lapisan *dense* terakhir 24 dengan *activation softmax* sebagai klasifikasi terakhir. Untuk konfigurasi parameter yang digunakan dapat dilihat pada tabel 3.

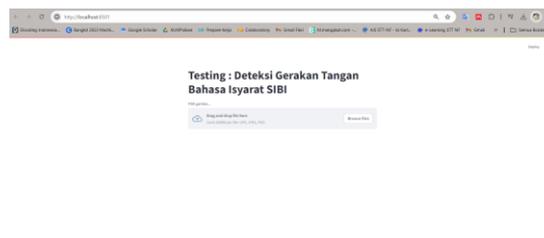
Tabel 3. Konfigurasi Hyperparameter

Jenis Pembagian Data	Jumlah
Optimizer	Adam
Loss Function	Categorical Crossentropy
Batch Size	256

Selanjutnya model dilatih menggunakan 50 *Epochs* (iterasi pelatihan).

3.6. Testing Model

Setelah model selesai melakukan proses *training* maka didapat *file* model berformat h.5. yang selanjutnya masuk ke dalam proses *testing* dengan menggunakan *framework streamlit* yang dapat membuat *app browser* sederhana yang bisa menampilkan bagaimana model bekerja untuk gerakan tangan bahasa isyarat SIBI. *Streamlit* menggunakan *localhost* yang disediakan oleh *streamlit* yang dapat dilihat pada gambar 9.



Gambar 9. Testing Model

Pengetesan dilakukan sebanyak 3 kali menggunakan gambar tangan bersumber dari kamera ponsel yang hasilnya dapat dilihat pada gambar 10.

Hasil Deteksi sumber gambar ponsel

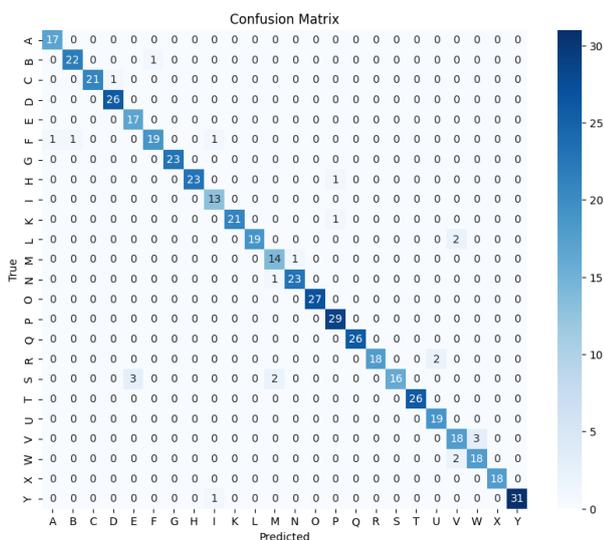


Gambar 10. Hasil Testing

Pada proses pengetesan ini *input* gambar Huruf SIBI B dapat menghasilkan persentase keyakinan sebesar 99,62%, berikutnya huruf L dengan tingkat 89,64% dan terakhir huruf W dengan persentase keyakinan 99,91%.

3.7. Evaluasi Model

Proses evaluasi menggunakan 2 pengukuran yakni menggunakan *confusion matrix* dengan matriks akurasi. Berikut *heatmap confusion matrix* hasil dari proses *training* model ini pada gambar 11.



Gambar 11. Heatmap Confusion Matrix

Hasil dari *confusion matrix* jika dilakukan perhitungan dengan matriks akurasi adalah sebagai berikut

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{504}{528} = 0.9545$$

Maka akurasi yang didapat adalah 0,9545 jika dipresentasikan maka angkanya menjadi 95,45%.

4. KESIMPULAN

Dari penelitian sudah dilakukan terdapat kesimpulan yang bisa ditarik yakni dalam Perancangan Model *Deep Learning* untuk Penerjemah Bahasa Isyarat SIBI dengan Arsitektur *Transfer Learning* MobileNetV2 mendapatkan hasil evaluasi akurasi yakni di angka 95,45% ini merupakan hasil akurasi yang baik.

Hasil dari penelitian ini diharapkan dapat dikembangkan khususnya penggunaan *dataset* yang lebih banyak agar model mendapatkan banyak variasi saat proses *training*. Terakhir harapan penelitian perancangan model ini dapat dikembangkan menjadi aplikasi penerjemah bahasa isyarat SIBI yang dapat membantu disabilitas tunarungu dan tunawicara.

DAFTAR PUSTAKA

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

[2] "Kamus SIBI." Accessed: Mar. 13, 2024. [Online]. Available: <https://pmpk.kemdikbud.go.id/sibi/profil>

[3] 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.

[4] S. Apendi and M. W. Paryasto, "Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector," 2023.

[5] 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.

[6] J. T. Terpadu, I. Arifin, R. Fakhran Haidi, And M. Dzalhaqi, "Penerapan Computer Vision Menggunakan Metode Deep Learning Pada Perspektif Generasi Ulul Albab," *Jurnal Teknologi Terpadu*, Vol. 7, No. 2, Pp. 98–107, 2021, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/jtt>

[7] A. Raup, W. Ridwan, Y. Khoeriyah, Q. Yulianti Zaqiah, and U. Islam Negeri Sunan Gunung Djati Bandung, "Deep Learning dan Penerapannya dalam Pembelajaran." [Online]. Available: <http://jiip.stkipyapisdompou.ac.id>

[8] 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.

[9] Pane Yosefan Yeremia And Shimbing Jordan Jeremia, "Klasifikasi Jenis Burung Menggunakan Metode Transfer Learning," *Jurnal Teknologi Terpadu*, Vol. 9, No. 2, Pp. 89–94, 2023, Accessed: Jul. 31, 2024. [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/jtt>

[10] 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.

[11] M. Rijal, Yani Muhammad Andi, And Rahman Abdul, "Deteksi Citra Daun Untuk Klasifikasi Penyakit Padi Menggunakan Pendekatan Deep Learning dengan Model Cnn," *Jurnal Teknologi Terpadu*, Vol. 10, No. 1, Pp. 56–62, 2024,

- Accessed: Jul. 31, 2024. [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/jtt>
- [12] 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.
- [13] 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>
- [14] 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.
- [15] 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.