



Penerapan *Library Tensorflow, Cvzone, dan Numpy* pada Sistem Deteksi Bahasa Isyarat Secara Real Time

I Nyoman Tri Anindia Putra^{1*}, Ketut Sepdyana Kartini², Yoga Kristian Suyitno³, I Made Sugiarta⁴
Ni Kadek Era Puspita⁵

^{1*,2,3,4,5}Institut Bisnis dan Teknologi Indonesia

trianindiaputra@instiki.ac.id, sepdyana@instiki.ac.id, vogakristiansuyitno04@gmail.com,
madesugiarta97531@gmail.com, erapuspita98@gmail.com

Abstract

As social beings, humans naturally need to interact with each other through communication, by being in society only to share information or respond to each other's needs, such as the learning process at school. However, it is difficult for deaf people to fully communicate because the environment is not friendly for them and one of them is that very few understand sign language. The Python-based program "My Signs" was developed to educate the public about Indonesian Sign Language (BISINDO), with features such as recognizing sign language using machine learning models and translating words using image output, as well as using Python. Services in software infrastructure. The development of this application uses the open CV library which is a library form of machine learning methods with python language, so that with the development of this application it is hoped that it will create for a deaf-friendly environment.

Keywords: *Machine Learning, Python, Open Cv, Sign Language*

Abstrak

Sebagai makhluk sosial, manusia secara alami perlu berinteraksi satu sama lain melalui komunikasi, dengan berada di masyarakat hanya untuk berbagi informasi atau saling menanggapi kebutuhan, seperti proses belajar di sekolah. Namun penyandang tunarungu sulit untuk berkomunikasi secara utuh karena lingkungan yang tidak ramah bagi mereka dan salah satunya adalah sangat sedikit yang memahami bahasa isyarat. Program berbasis Python "My Signs" dikembangkan untuk mengedukasi masyarakat tentang Bahasa Isyarat Indonesia (BISINDO), dengan fitur-fitur seperti mengenali bahasa isyarat menggunakan model pembelajaran mesin dan menerjemahkan kata menggunakan output gambar, serta menggunakan Python. Layanan dalam infrastruktur perangkat lunak. Pengembangan aplikasi ini menggunakan library open CV yang merupakan

bentuk library dari metode machine learning dengan Bahasa python, sehingga dengan berkembangnya aplikasi ini diharapkan terciptanya lingkungan yang ramah bagi tunarungu.

Kata Kunci: *Machine Learning, Python, Open Cv, Bahasa Isyarat*

This Journal is licensed under a Creative Commons Attribution 4.0 International License [Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/) license

PENDAHULUAN

Pada metode bicara dialek tuli yaitu terdapat kekurangan dalam pemahaman orang tentang isyarat yang digunakannya. Isu-isu tersebut membuat penyandang tunarungu sangat terbebani dengan masalah dialog dengan masyarakat pada umumnya [1]. Orang tuli berkomunikasi dalam bahasa isyarat [2]. Bahasa isyarat biasanya berupa gerakan tangan, gerak tubuh, simbol pembentuk tubuh yang berarti huruf atau kata[3] Penelitian tentang bahasa isyarat telah dilakukan di berbagai negara, dengan menggunakan metode pengolahan citra dan computer vision [4]. Di era Teknologi 4.0, machine learning menjadi salah satu metode yang paling populer. Pembelajaran mesin pada dasarnya adalah sebuah proses dimana komputer mempelajari hal-hal yang tidak pernah diketahui dengan menggunakan banyak sumber informasi [5]. Bahasa isyarat merupakan salah satu alat komunikasi yang digunakan bagi penyandang disabilitas atau penyandang tunarungu [6].

Di Indonesia, ada dua sistem yang digunakan yaitu Bahasa Isyarat Indonesia (BISINDO) dan Sistem Isyarat Indonesia (SIBI)[7]. BISINDO yang digunakan sehari-hari oleh teman-teman tuna rungu ini berisi kosa kata tanda-tanda simbolik. Sepanjang arti kata itu terwakili, kosa kata simbolis sederhana BISINDO sudah cukup[3], [8]. Bahasa isyarat dapat dikenali dengan bantuan pembelajaran mesin menggunakan beberapa teknik pembelajaran mesin. Dalam hal ini, machine learning dengan perintahnya sendiri mampu menangkap data yang ada[9]. Tugas yang dapat dilakukan pembelajaran mesin sangat bervariasi tergantung pada apa yang dipelajarinya. Teknologi *Machine Learning* (ML) adalah mesin yang dirancang untuk belajar dengan sendirinya tanpa instruksi pengguna[10]. Sejak komputer ditemukan, orang mulai mempelajari cara kerja komputer. Dimulai ketika Arthur Samuel berhasil membuat program Checkers untuk komputer IBM pada tahun 1952. pada dasarnya prinsip-prinsip pembelajaran mesin masih sama, meliputi akuisisi data, eksplorasi data, pemilihan model atau teknik, pelatihan model yang dipilih, dan evaluasi hasil ML[11].

Berdasarkan penjelasan maka tujuan penelitian yaitu membuat sistem yang bisa mendeteksi Bahasa isyarat yaitu “Deteksi Bahasa Isyarat Menggunakan Metode Detection and Classification Secara Real Time”. Manfaat penelitian ini adalah untuk mempermudah mendeteksi Bahasa isyarat bagi mereka yang belum atau bahkan tidak mengetahui tentang Bahasa isyarat.

METODE PENELITIAN

Tinjauan Pustaka

Secara pengertian, bahasa Python adalah kode sumber yang telah diterjemahkan ke dalam format bytecode sebelum diterjemahkan ke dalam bahasa manusia[12]. Sebagai contoh penggunaan Python pada

software *WorkFlow* adalah pengembangannya dengan menambahkan RTI (*Run Time Interaction*), RTCF (*Run Time Control Functions*), dan BTF (*Build Time Function*). Bahasa pemrograman dapat diartikan sebagai susunan kode yang digunakan untuk menjalankan program komputer. Kode-kode ini berisi berbagai instruksi yang dikenal sebagai sintaks dan semantik. Selanjutnya, sintaks dan semantik dimasukkan ke dalam program komputer. Baru setelah itu ada reaksi terhadap perubahan sistem atau pemrograman yang ada[13]. Pycharm adalah salah satu IDE Python. Pycharm terdiri dari penganalisis kode, debugger, alat pengujian, dan opsi kontrol versi. PyCharm menawarkan beberapa fitur terbaik untuk pengguna dan pengembangnya dengan mendukung pemrograman web dan kerangka kerja seperti Django dan Flask. Ini juga membantu pengembang membuat ekstensi Python menggunakan berbagai API yang tersedia. Pycharm tersedia untuk Windows, macOS dan Linux.

Library yang digunakan adalah OpenCV (Open Source Computer Vision) adalah pustaka sumber terbuka yang dikembangkan oleh Intel yang berfokus pada penyederhanaan pemrograman gambar digital. Intel Research secara resmi merilis OpenCV pada tahun 1999 sebagai kelanjutan dari bagian proyek yang menghasilkan aplikasi intensif CPU, pelacakan sinar waktu nyata, dan dinding visual 3D[14].

Pengenalan wajah, pelacakan wajah, pengenalan wajah, pemfilteran Kalman dan berbagai metode AI (Kecerdasan Buatan). Dan memaparkan algoritme sederhana visi komputer ke API tingkat rendah. *Classification* merupakan proses memprediksi kelas atau kategori dari nilai yang diamati dan titik data yang diberikan. Deteksi objek menggabungkan klasifikasi dan lokalisasi untuk menentukan objek apa yang ada dalam gambar atau video dan di mana mereka berada dalam gambar[15].v. Ini menerapkan klasifikasi ke objek yang berbeda dan menggunakan kotak pembatas. Pendekatan paling populer untuk visi komputer adalah klasifikasi dan deteksi objek untuk mengidentifikasi objek yang ada dalam gambar dan menentukan posisinya. Tetapi banyak kasus penggunaan memerlukan analisis gambar pada level yang lebih rendah dari itu sehingga segmentasi gambar dapat dilakukan[14].

Tahapan Penelitian

Pada tahapan penelitian ini, penulis melakukan beberapa tahap penelitian terhadap sistem ini yaitu

- Mengumpulkan data / contoh Bahasa isyarat
- Mempersiapkan aplikasi pycharm
- Mempersiapkan script code
- Mengumpulkan data untuk testing
- Testing sistem
- Mengumpulkan tingkat akurasi

Rancangan Penelitian

Rancangan penelitian yang dilakukan pada penelitian ini adalah membuat flowchart alur. Menentukan sample kode tangan yang digunakan dari sistem tersebut. Gambar di bawah merupakan gambar yang digunakan untuk sampel kode Bahasa isyarat.[16].

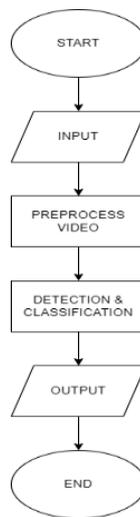
Dataset adalah kumpulan data yang telah dianalisis atau diproses dalam beberapa cara. Penambangan data adalah proses menemukan informasi penting dari sejumlah besar data. Ini dilakukan dengan menggunakan

metode statistik, matematika, pembelajaran mesin, dan teknologi kecerdasan buatan. Penambangan data digunakan untuk mengelompokkan kumpulan data.



Gambar 1. Sampel Kode Tangan

Berikut merupakan alur dari sistem yang penulis rancang. Dibuat dalam bentuk flowchart. Sistem ini menggunakan metode Detection and Classification.



Gambar 2. Flowchart Rancangan Penelitian

Pada gambar 2 dapat dijelaskan tahapan rancangan penelitian dimulai dengan menentukan inputan data, setelah itu melakukan *preprocess* video yang akan dianalisis, tahap selanjutnya melanjutkan proses *detection* dan klasifikasi dengan penerapan *library* sehingga dapat menghasilkan output berupa deteksi bahasa isyarat.

Teknik Pengumpulan dan Analisis Data

Pada penelitian ini kode tangan bahasa isyarat Bisindo digunakan sebagai bahan masukan. Yang mana sebelum pelatihan terlebih dahulu diujicobakan untuk mendapatkan citra yang sesuai dengan huruf abjad bahasa isyarat Bisindo, kemudian diambil 150 gambar setiap hurufnya. Diambil dan kemudian disimpan dalam folder dengan huruf alfabet nama. Contoh gambar contoh latihan. Informasi kode gerakan tangan kompatibel dengan kode tangan alfabet.[17]

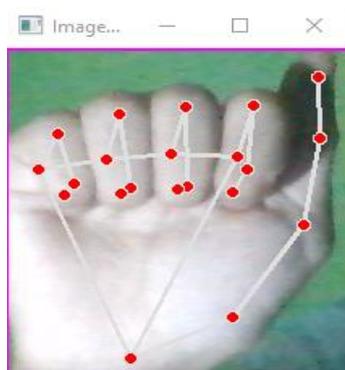
HASIL DAN PEMBAHASAN

Antarmuka Sistem

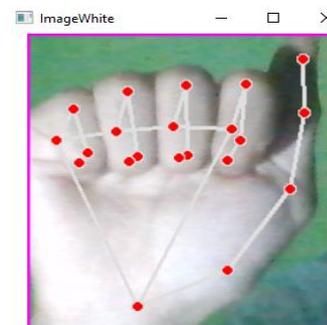
Berikut tampilan antarmuka system yang sudah diterapkan telah melalui proses editing. Berikut tampilan dari antarmuka dari system pendeteksi objek secara real-time menggunakan webcam bawaan dari laptop sebagai media pengaplikasian system. [18]



Gambar 1. Antarmuka Sistem 1



Gambar 2. Antarmuka Sistem 2



Gambar 3. ImageWhite

Implementasi

Tahap selanjutnya yaitu tahap implementasi dari sistem yang sudah dirancang

a) Library yang digunakan :

1. Tensorflow

Dengan Tensorflow, pengguna dapat membuat grafik yang menunjukkan bagaimana data berpindah melalui tampilan grafis atau serangkaian node pemrosesan. Setiap node dalam grafik mewakili operasi matematika, dan setiap koneksi atau ujung antara node ini adalah array data multidimensi.

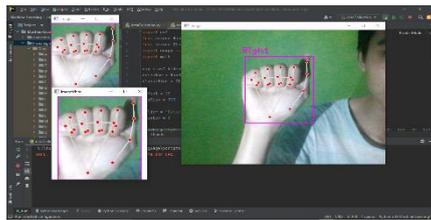
2. Cvzone

CVZone merupakan sebuah library Python yang digunakan untuk hal-hal seperti Artificial Intelligence, Computer Vision, dan Deep Learning. CVZone gratis untuk digunakan, tidak memiliki bug yang diketahui, dan memiliki file instalasi siap pakai yang tersedia.

3. Numpy

NumPy adalah pustaka fungsi yang memungkinkan Anda melakukan banyak tugas manipulasi data umum dengan Python. Banyak interaksi antara array NumPy dan Python sangat mirip dengan apa yang akan Anda lakukan dengan variabel Python biasa. Untuk tugas yang lebih lanjut, NumPy juga memiliki fungsi yang bekerja dengan aljabar linier, transformasi Fourier, dan matriks.

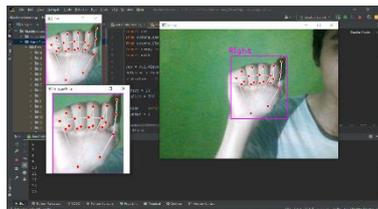
- b) Setelah codingan sudah diketik, debug codingan pada aplikasi Pycharm dan akan muncul tampilan seperti gambar di bawah:



Gambar 4. Testing 1

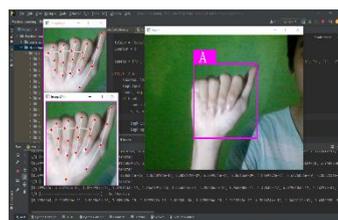
Contoh pengambilan data untuk huruf A, pada window Image adalah display utama kita dalam melihat hasil deteksi tangan, window ImageWhite adalah hasil gambar hanya tangan kita yang diluar area-nya diberikan warna putih untuk mengisi kosongnya pixel pada area tersebut, dan diatas ImageWhite adalah ImageCrop yang dimana itu adalah hasil gambar hanya tangan kita yang di crop secara otomatis dan jika terdapat area yang pixel-nya kosong akan diisi oleh ImageWhite

- c) Saat melakukan pengambilan data gambar akan muncul angka pada terminal. Seperti gambar di bawah yaitu angka 1 sampai 14 dengan menekan lama huruf s pada keyboard untuk mengambil data. Angka tersebut muncul pada bagian bawah.



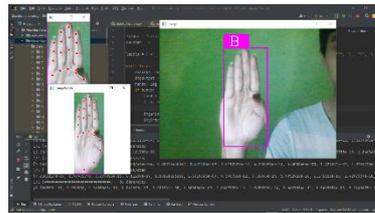
Gambar 5. Testing 2

- d) Setelah gambar sudah diambil dan dilatih dan ini adalah contoh hasil dari olahan data tersebut. Kode tersebut bisa terdeteksi saat setelah sistem di debug. Huruf yang muncul yaitu huruf A.



Gambar 6. Testing Huruf A

Setelah gambar sudah diambil dan dilatih dan ini adalah contoh hasil dari olahan data tersebut. Kode tersebut bisa terdeteksi saat setelah sistem di debug. Huruf yang muncul yaitu huruf B.



Gambar 7. Testing Huruf B

C. Pengujian

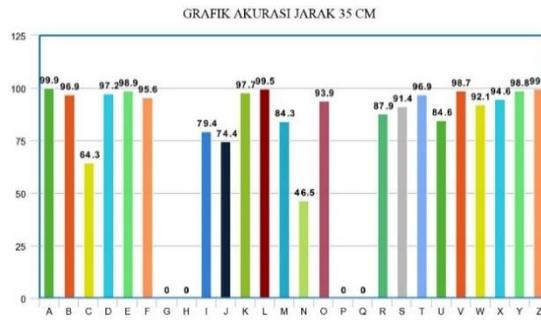
Tahap terakhir yaitu tahap pengujian. Pengujian wajib dilakukan agar mengetahui apakah sistem layak digunakan atau tidak. Pada tahap ini penulis telah melakukan pengujian sebanyak 20 kali pada setiap alphabet dan mendapatkan tingkat akurasi yang berbeda. Penulis melakukan pengujian sebanyak 20 kali agar hasil maksimal. Pengujian ini dibuat dalam bentuk tabel dan grafik

Berikut tabel pengujian dengan jarak 35 cm dengan melakukan uji coba sebanyak 20 kali. Cara melihat akurasi yaitu pada saat melakukan uji coba muncul angka akurasi pada terminal. Setelah itu dibuatkan grafik.

Tabel 1. Tingkat Akurasi Jarak 35 cm

Huruf	Uji Coba	Jarak	Akurasi %	Akurasi
A	20	35cm	99.9953125%	0.999953125
B	20	35cm	96.9938672%	0.999938672
C	20	35cm	64.3765108%	0.643765108
D	20	35cm	97.2628462%	0.972628462
E	20	35cm	98.9901782%	0.989901782
F	20	35cm	95.6797642%	0.956797642
G	20	35cm	-	-
H	20	35cm	-	-
I	20	35cm	79.4260713%	0.794260713
J	20	35cm	74.4326321%	0.744326321
K	20	35cm	97.7836585%	0.977836585
L	20	35cm	99.5376868%	0.995376868
M	20	35cm	84.3573563%	0.843573563
N	20	35cm	46.5875354%	0.465875354
O	20	35cm	93.9362147%	0.939362147
P	20	35cm	-	-
Q	20	35cm	-	-
R	20	35cm	87.9754257%	0.879754257
S	20	35cm	91.4268794%	0.914268794
T	20	35cm	96.9864275%	0.969864275
U	20	35cm	84.6557246%	0.846557246
V	20	35cm	98.7917452%	0.987917452
W	20	35cm	92.1487653%	0.921487653
X	20	35cm	94.6861842%	0.946861842
Y	20	35cm	98.8542405%	0.988542405
Z	20	35cm	99.7652378%	0.997652378
Total	520	35cm	Tidak bisa dihitung	Tidak bisa dihitung

Dari tabel di atas terdapat tanda (-) pada tabel akurasi itu berarti huruf tidak mengeluarkan akurasi tetapi kode tagan dapat dideteksi. Sehingga tingkat akurasi keseluruhan tidak dapat dihitung. Itu terjadi dikarenakan pencahayaan yang kurang pada saat uji coba dan huruf susah dideteksi. Dari keseluruhan terdapat 4 huruf yang susah dideteksi dengan jarak 35 cm. Setelah dibuat dalam bentuk tabel, penulis juga membuat dalam bentuk grafik.



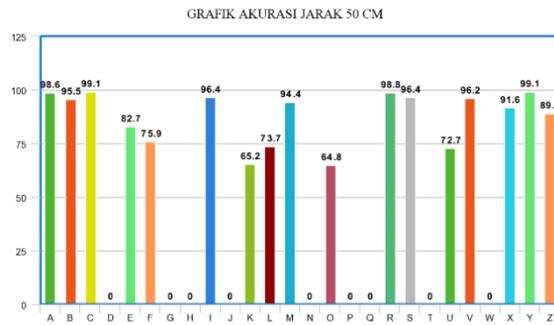
Gambar 8. Tingkat Akurasi Dengan Jarak 35 cm

Berikut tabel pengujian dengan jarak 50 cm dengan melakukan uji coba sebanyak 20 kali. Cara melihat akurasi yaitu pada saat melakukan uji coba muncul angka akurasi pada terminal. Setelah itu dibuatkan grafik.

Tabel 2. Tingkat Akurasi Jarak 50 cm

Huruf	Uji Coba	Jarak	Akurasi %	Akurasi
A	20	50cm	98.6472652%	0.986472652
B	20	50cm	95.5667342%	0.955667342
C	20	50cm	99.1425173%	0.991425173
D	20	50cm	-	-
E	20	50cm	82.7536287%	0.827536287
F	20	50cm	75.9134627%	0.759134627
G	20	50cm	-	-
H	20	50cm	-	-
I	20	50cm	96.4249761%	0.964249761
J	20	50cm	-	-
K	20	50cm	65.2908825%	0.652908825
L	20	50cm	73.7765754%	0.737765754
M	20	50cm	94.4808727%	0.944808727
N	20	50cm	-	-
O	20	50cm	64.8729585%	0.648729585
P	20	50cm	-	-
Q	20	50cm	-	-
R	20	50cm	98.8256512%	0.988256512
S	20	50cm	96.4288568%	0.964288568
T	20	50cm	-	-
U	20	50cm	72.7256520%	0.727256520
V	20	50cm	96.2576751%	0.962576751
W	20	50cm	-	-
X	20	50cm	91.6752548%	0.916752548
Y	20	50cm	99.1342642%	0.991342642
Z	20	50cm	89.2562563%	0.892562563
Total	520	50cm	Tidak bisa dihitung	Tidak bisa dihitung

Dari tabel di atas terdapat tanda (-) pada tabel akurasi itu berarti huruf tidak mengeluarkan akurasi tetapi kode tagan dapat dideteksi. Sehingga tingkat akurasi keseluruhan tidak dapat dihitung. Itu terjadi dikarenakan pencahayaan yang kurang pada saat uji coba dan huruf susah dideteksi. Dari keseluruhan terdapat 9 huruf yang susah dideteksi dengan jarak 50 cm. Setelah dibuat dalam bentuk tabel, penulis juga membuat dalam bentuk grafik.



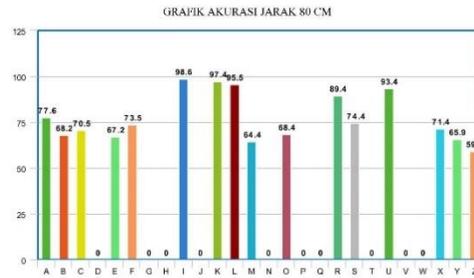
Gambar 9. Tingkat Akurasi Dengan Jarak 50 cm

Berikut tabel pengujian dengan jarak 80 cm dengan melakukan uji coba sesuai dengan sistem mendeteksinya. Cara melihat akurasi yaitu pada saat melakukan uji coba muncul angka akurasi pada terminal. Setelah itu dibuatkan grafik.

Tabel 3. Tingkat Akurasi Jarak 80 cm

Huruf	Uji Coba	Jarak	Akurasi %	Akurasi
A	5	80cm	77.6426756%	0.776426756
B	7	80cm	68.2854247%	0.682854247
C	6	80cm	70.5165424%	0.705165424
D	n/a	80cm	-	-
E	9	80cm	67.2446586%	0.672446586
F	6	80cm	73.5614871%	0.735614871
G	10	80cm	-	-
H	10	80cm	-	-
I	8	80cm	98.6246552%	0.986246552
J	n/a	80cm	-	-
K	12	80cm	97.4625865%	0.974625865
L	7	80cm	95.5927164%	0.955927164
M	8	80cm	64.4687614%	0.644687614
N	n/a	80cm	-	-
O	10	80cm	68.4226852%	0.684226852
P	n/a	80cm	-	-
Q	n/a	80cm	-	-
R	14	80cm	89.4244517%	0.894244517
S	7	80cm	74.4775201%	0.744775201
T	n/a	80cm	-	-
U	9	80cm	93.4765249%	0.934765249
V	n/a	80cm	-	-
W	n/a	80cm	-	-
X	6	80cm	71.4681441%	0.714681441
Y	10	80cm	65.9141845%	0.659141845
Z	9	80cm	59.3476548%	0.593476548
Total	153	80cm	47.5357951%	0.475357951

Dari tabel di atas terdapat tanda (-) pada tabel akurasi itu berarti huruf tidak mengeluarkan akurasi tetapi kode tagan dapat dideteksi. Itu terjadi dikarenakan pencahayaan yang kurang pada saat uji coba dan huruf susah dideteksi. Dari keseluruhan terdapat 10 huruf yang susah dideteksi dengan jarak 80 cm. Akurasi yang didapatkan secara keseluruhan yaitu 47,5%. Setelah dibuat dalam bentuk tabel, penulis juga membuat dalam bentuk grafik.[19]



Gambar 10. Tingkat Akurasi Dengan Jarak 80 cm

Keterangan tambahan

*ps : Hasil pendeteksian kedepannya mungkin berbeda

*pss : n/a artinya tidak terdeteksi sama sekali

*psss : - artinya Tidak ada nilai

*pssss : Total akurasi Tabel 1 dan 2 tidak bisa dihitung karena huruf berhasil di deteksi tapi angka tingkat keakurasiannya tidak muncul

Berdasarkan tabel di atas, ada 4 huruf yang dapat dideteksi tetapi tingkat akurasinya tidak muncul dari 3 jarak yang sudah diuji yaitu huruf G, H, P, dan Q. Pada saat pengujian, keempat huruf tersebut bisa dideteksi dan muncul huruf , akan tetapi hasil akurasinya tidak muncul sehingga 2 tabel di atas tidak menghasilkan akurasi secara keseluruhan. Pada tabel dengan jarak 80cm mendapatkan hasilakurasi secara keseluruhan yaitu 47,5 %.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian sistem tersebut dapat disimpulkan bahwa tingkat akurasi sistem deteksi Bahasa isyarat dari 26 alphabet dan 20 kali uji coba terdapat 4 huruf yang tidak bisa terdeteksi. Hal ini dikarenakan pencahayaan yang kurang pada saat melakukan uji coba, selain itu ada beberapa huruf yang bentuk tangannya hampir sama dengan huruf lainnya. Tingkat akurasi yang didapatkan dari keseluruhan alphabet dan uji coba yaitu 47,5%. Dengan adanya sistem ini dengan tingkat akurasi yang cukup relevan, dapat membantu teman-teman/saudara kita yang berkebutuhan khusus untuk berkomunikasi menggunakan bahasa isyarat. Dari hasil yang dihasilkan oleh sistem “Deteksi Bahasa Isyarat Menggunakan Metode Detection And Classification Secara Real Time”, banyak kekurangan dari sistem ini terutama pada tingkat akurasinya. Untuk itu penulis memberikan saran sebagai berikut (1) Sistem ini perlu dirancang ulang dengan metode selain Detection and Classification agar mendapatkan hasil yang maksimal. (2) Pada database, diharapkan agar tidak menggunakan 1 tangan yang sama untuk mempermudah sistem untuk mendeteksi bentuk tangan. (3) Perlu dibuatkan berbasis android agar pengguna lebih mudah mengaplikasikan sistem ini.

DAFTAR PUSTAKA

- [1] “View of Aplikasi Penerjemah Bahasa Isyarat Indonesia Ke Dalam Huruf Abjad.”
- [2] “View of PENGEMBANGAN APLIKASI PENGENALAN BAHASA ISYARAT ABJAD SIBI MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN).”
- [3] “View of KLASIFIKASI OBJEK KODE TANGAN PADA PENGENALAN ISYARAT ALPHABET BAHASA ISYARAT INDONESIA (BISINDO).”
- [4] O. Dwi Nurhayati, D. Eridani, and M. Hafiz Tsalavin, “SISTEM ISYARAT BAHASA INDONESIA (SIBI) METODE CONVOLUTIONAL NEURAL NETWORK SEQUENTIAL SECARA REAL TIME A REAL-TIME INDONESIAN LANGUAGE SIGN SYSTEM USING THE CONVOLUTION NEURAL NETWORK METHOD”, doi: 10.25126/jtiik.202294787.
- [5] “View of Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network | Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi).”
- [6] “Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan Menggunakan Metode Convolutional Neural Network dan Recurrent Neural Network | Yolanda | Jurnal Infra.”
- [7] “Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network | Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi).”
- [8] R. Sinukun and Y. Darise, “Aplikasi Bahasa Isyarat Sederhana Berbasis Android,” *J. Teknol. Inf. Indones.*, vol. 2, no. 1, pp. 20–26, 2017.
- [9] S. T. Isma, “Meneliti bahasa isyarat dalam perspektif variasi bahasa,” *Kongr. Bhs. Indones.*, pp. 1–14, 2018.
- [10] R. I. Borman and B. Priyopradono, “Implementasi Penerjemah Bahasa Isyarat Pada Bahasa Isyarat Indonesia (BISINDO) Dengan Metode Principal Component Analysis (PCA),” *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 103–108, 2018.
- [11] N. Huda, “Aplikasi Bahasa Isyarat Pengenalan Huruf Hijaiyah Bagi Penyandang Disabilitas Tuna Rungg,” *Apl. Bhs. Isyarat Pengenalan Huruf Hijaiyah Bagi Penyandang Disabil. Tuna Rungg*, 2022.
- [12] D. Sarkar, R. Bali, and T. Sharma, “Practical machine learning with Python,” *A Probl. Guid. To Build. Real-World Intell. Syst. Berkely Apress*, 2018.
- [13] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. “O’Reilly Media, Inc.,” 2016.
- [14] S. Raschka and V. Mirjalili, “Python machine learning: Machine learning and deep learning with python,” *Scikit-Learn, TensorFlow. Second Ed. ed*, vol. 3, 2017.
- [15] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.
- [16] I. N. T. A. Putra, “Face Recognition Based Mobile Using Fisherface and Distance Classifier,” *JST (Jurnal Sains dan Teknol.*, vol. 7, no. 1, pp. 135–145, 2018, doi: 10.23887/jst-undiksha.v7i1.13267.
- [17] I. N. Tri Anindia Putra, “Perancangan dan Pengembangan Sistem Absensi Realtime Melalui Metode

- Pengenalan Wajah,” *JST (Jurnal Sains dan Teknol.*, vol. 3, no. 2, pp. 450–467, 2015, doi: 10.23887/jst-undiksha.v3i2.4480.
- [18] I. N. T. A. Putra and E. D. Krisna, “Implementasi Sistem Surveillance Berbasis Pengenalan Wajah pada STMIK STIKOM Indonesia,” *J. Ilmu Komput.*, vol. 13, no. 2, p. 8, 2020, doi: 10.24843/jik.2020.v13.i02.p01.
- [19] I. N. T. A. Putra and K. S. Kartini, “Perbandingan Metode Pengenalan Wajah Melalui Surveillance Berbasis Pengenalan Wajah,” *SINTECH (Science Inf. Technol. J.*, vol. 4, no. 1, pp. 88–98, 2021, doi: 10.31598/sintechjournal.v4i1.660.