FORMAT: Jurnal Ilmiah Teknik Informatika
Akreditasi Sinta 5 RISTEKBRIN No. 164/E/KPT/2021; E-ISSN 2722-7162 | P-ISSN 2089-5615

Vol. 14, No. 1,
January 2025,
81-89. 2025

Machine Learning System untuk Mendeteksi Gerakan Tubuh Menggunakan Library Mediapipe

Irfan Nurdiansyah [1] ; Reni Utami [2] ; Muchamad Sandy[3]

Fakultas Ilmu Komputer, Universitas Dian Nusantara, Jl. Tj, Duren Barat. 2 No.1, RT.1/RW.5

[1] irfan.nurdiansyah@dosen.undira.ac.id , , [2] reni.utami@dosen.undira.ac.id , [3] muchamad.sandy@dosen.undira.ac.id

Kata kunci:
Machine learning, Bahasa Isyarat, Mediapipe, ASL

Abstract

Communication with people with hearing and speech disabilities is often challenging. Sign language is the primary tool that helps them convey thoughts and feelings, but it is often difficult for those who are not used to it to understand. This project aims to develop a machine learning model to recognize hand gestures in spelling fingers using American Sign Language (ASL). The model uses image data and Computer Vision techniques to train a deep learning algorithm that can recognize signals in real-time through a camera. The system utilizes deep neural networks that work through layers of nodes to process, classify, and predict cues accurately

Pendahuluan

Bahasa isyarat, sebagai modalitas komunikasi utama bagi komunitas tunarungu, memiliki peran krusial dalam interaksi sosial dan aksesibilitas informasi. Namun, perbedaan antara bahasa isyarat dan bahasa lisan/tulis seringkali menjadi penghalang komunikasi [1]. Oleh karena itu, pengembangan sistem machine learning untuk deteksi bahasa isyarat menjadi penting guna menjembatani kesenjangan komunikasi ini. Sistem ini diharapkan mampu menerjemahkan bahasa isyarat ke dalam bahasa yang dapat dipahami oleh masyarakat umum, sehingga meningkatkan inklusi dan partisipasi komunitas tunarungu dalam berbagai aspek kehidupan.

Penelitian ini difokuskan pada perancangan dan implementasi sistem machine learning [2] yang mampu mengenali bahasa isyarat. Sistem ini memanfaatkan pemrosesan citra [3] dan deep learning [4] untuk mengidentifikasi gerakan tangan [5], ekspresi wajah, dan elemen visual lainnya yang merupakan ciri khas bahasa isyarat. Tujuan utama adalah menciptakan sistem yang akurat dan efisien dalam menerjemahkan bahasa isyarat ke dalam teks atau suara [6], sehingga memudahkan komunikasi antara individu tunarungu dan masyarakat umum.

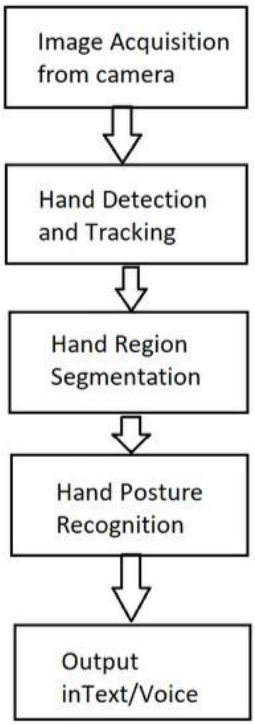
Dalam penelitian ini, kami memanfaatkan MediaPipe [7], sebuah framework sumber terbuka yang dikembangkan oleh Google, untuk memfasilitasi deteksi dan pelacakan gerakan tangan serta ekstraksi fitur-fitur penting dari bahasa isyarat. MediaPipe menyediakan solusi yang efisien dan real-time untuk pemrosesan data multimodal, termasuk video. Modul Hand Pose Estimation [8] dari MediaPipe memungkinkan identifikasi titik-titik kunci (landmark) pada tangan dengan akurasi yang cukup tinggi. Informasi ini kemudian digunakan sebagai input untuk model machine learning yang bertugas menerjemahkan bahasa isyarat.

Penggunaan MediaPipe [9] dalam penelitian ini memberikan beberapa keuntungan. Pertama, framework ini mudah diintegrasikan dengan berbagai bahasa pemrograman dan platform, sehingga mempercepat proses pengembangan sistem. Kedua, MediaPipe telah dioptimalkan untuk kinerja tinggi, memungkinkan pemrosesan video secara real-time tanpa mengorbankan akurasi. Ketiga, ketersediaan model pre-trained untuk deteksi tangan [10] dan estimasi pose mengurangi kebutuhan sumber daya komputasi dan waktu pelatihan model. Dengan memanfaatkan MediaPipe, penelitian ini dapat lebih fokus pada pengembangan algoritma machine learning yang efektif untuk penerjemahan bahasa isyarat, alih-alih membangun sistem deteksi tangan dari awal.

Oleh karena itu, batasan penelitian ini meliputi kemampuan sistem dalam mengenali satu huruf untuk setiap input, keterbatasan deteksi tubuh, dan belum mampunya sistem untuk merangkai beberapa huruf menjadi kata utuh. Meskipun demikian, penelitian ini merupakan langkah awal yang penting dalam pengembangan sistem deteksi bahasa isyarat yang lebih komprehensif. Penelitian lebih lanjut akan difokuskan pada peningkatan akurasi dan kemampuan sistem dalam mengenali kata dan kalimat, serta memperluas cakupan deteksi bahasa isyarat.

Metode penelitian

Untuk metode penelitian pada sistem ini terdiri dari dua bagian utama, yaitu perangkat perekam dan komputer. Perangkat perekam bertugas menangkap gerakan isyarat yang dilakukan oleh penyandang tunarungu, kemudian mengirimkan informasi tersebut ke komputer. Komputer kemudian menganalisis data tersebut untuk mengungkap makna dari isyarat yang dilakukan. Penjelasan ini biasanya dilengkapi dengan ilustrasi visual.

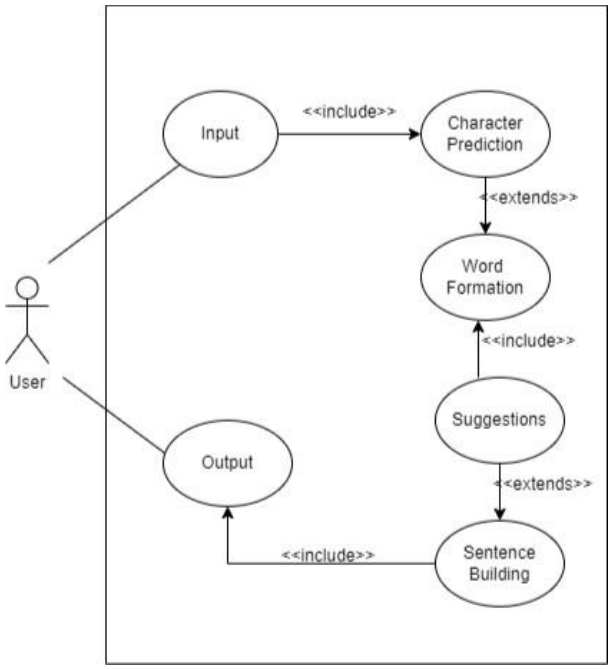


Gambar 1. Proses dalam proses penelitian.

A. **Tahapan Penelitian**

Berdasarkan diagram alur atau flowchart yang menggambarkan proses pengenalan atau penerjemahan bahasa isyarat menjadi teks atau suara. Diagram ini terdiri dari beberapa tahapan yang berurutan, dimulai dari akuisisi gambar hingga menghasilkan keluaran berupa teks atau suara, antara lain :

1. **Image Acquisition from Camera (Pengambilan Gambar dari Kamera)** : Tahap ini merupakan langkah awal dalam proses. Kamera digunakan untuk mengambil gambar atau video dari bahasa isyarat yang dilakukan oleh pengguna. Gambar atau video ini akan menjadi input untuk tahapan selanjutnya.
2. **Hand Detection and Tracking (Deteksi dan Pelacakan Tangan)** : Setelah gambar diambil, sistem akan mendeteksi dan melacak keberadaan tangan dalam gambar atau video. Proses ini melibatkan identifikasi posisi tangan, ukuran, dan gerakan tangan dari waktu ke waktu.
3. **Hand Region Segmentation (Segmentasi Wilayah Tangan)** : Pada tahap ini, sistem akan memisahkan atau mengekstrak wilayah tangan dari bagian gambar atau video lainnya. Tujuannya adalah untuk memfokuskan analisis pada tangan yang melakukan bahasa isyarat.
4. **Hand Posture Recognition (Pengenalan Postur Tangan)** : Setelah wilayah tangan tersegmentasi, sistem akan mengenali postur atau bentuk tangan yang digunakan dalam bahasa isyarat. Setiap postur tangan memiliki arti atau simbol tertentu dalam bahasa isyarat.
5. **Output in Text/Voice (Keluaran dalam Teks/Suara)** : Tahap terakhir adalah menghasilkan keluaran berdasarkan postur tangan yang dikenali. Keluaran ini dapat berupa teks yang ditampilkan pada layar atau suara yang dihasilkan oleh speaker. Teks atau suara ini merupakan terjemahan dari bahasa isyarat yang dilakukan oleh pengguna.



Gambar 2. Use case Penelitian

B. **Use Case Penelitian**

Diagram use case ini memetakan interaksi antara aktor (pengguna) dan sistem deteksi bahasa isyarat, yang merupakan fokus utama penelitian ini. Interaksi dimulai dengan input pengguna, yaitu gerakan isyarat tangan yang ditangkap oleh perangkat input. Input ini kemudian diproses melalui serangkaian use case yang menggambarkan tahapan-tahapan dalam sistem.

Use case "**Character Prediction**" memegang peranan krusial dalam penelitian ini, karena akurasi identifikasi karakter isyarat akan sangat mempengaruhi kinerja sistem secara keseluruhan. Penelitian ini bertujuan untuk mengembangkan algoritma yang robust dan efisien untuk melakukan prediksi karakter dengan tingkat ketepatan yang tinggi, bahkan dalam kondisi variasi pencahayaan, latar belakang, dan gaya isyarat individu.

Use case "**Word Formation**" dan "**Sentence Building**" menunjukkan bagaimana sistem melangkah lebih jauh dari sekadar identifikasi karakter individual. Penelitian ini juga mengeksplorasi metode untuk mengintegrasikan karakter-karakter yang terprediksi menjadi kata dan kalimat yang bermakna. Hal ini melibatkan penggunaan model bahasa dan teknik pemrosesan bahasa alami untuk memastikan bahwa output sistem relevan dan mudah dipahami.

Use case "**Suggestions**" merupakan fitur tambahan yang dapat meningkatkan pengalaman pengguna. Penelitian ini mempertimbangkan pengembangan mekanisme pemberian saran kata atau frasa yang relevan, sehingga pengguna dapat dengan cepat dan mudah membentuk kalimat yang diinginkan.

Pada bagian, use case "**Output**" merepresentasikan hasil akhir dari proses penerjemahan bahasa isyarat, yaitu teks atau suara yang dihasilkan oleh sistem. Penelitian ini menekankan pentingnya output yang akurat, jelas, dan mudah diakses oleh pengguna.

Secara keseluruhan, diagram use case yang dibuat pada penelitian ini memberikan kerangka kerja yang komprehensif untuk memahami interaksi antara pengguna dan sistem deteksi bahasa isyarat dalam penelitian ini. Diagram ini menyoroti aspek-aspek penting yang menjadi fokus penelitian, yaitu akurasi prediksi karakter, integrasi karakter menjadi kata dan kalimat, pemberian saran, dan kualitas output

Hasil dan diskusi

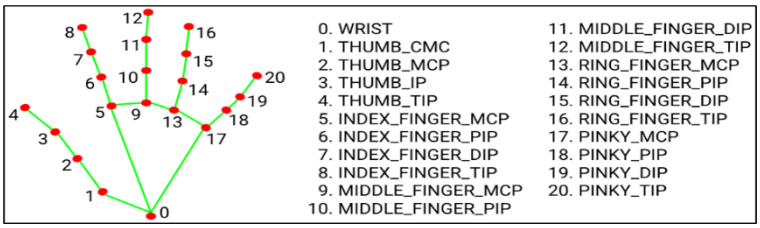
Penelitian ini dikembangkan dengan menggunakan bahasa pemrograman Python serta memanfaatkan alat serta teknik pembelajaran mendalam untuk berbagai fungsi, termasuk Pelacakan Tangan. Proses pada penelitian ini yaitu sistem menangkap gambar dari kamera, kemudian mendeteksi titik-titik kunci gerakan tangan, serta memproses data gambar, dan membuat keputusan berdasarkan model yang telah dilatih dengan metode Machine Learning.

Pada penelitian ini, proses implementasi sistem dirancang untuk mengenali 24 huruf yang digunakan dalam bahasa isyarat, dengan langkah awal dengan pengambilan gambar gestur tangan, kemudian melakukan beberapa proses implementasi, antara lain :

Gambar 3. Gestur tangan yang digunakan dalam proses penelitian

1. Deteksi Titik Utama

Langkah awal pada proses implementasi adalah menangkap gerakan tangan menggunakan kamera, dengan bantuan pustaka Python OpenCV untuk menghasilkan gambar dalam format BGR. Setelah itu, dilakukan penyesuaian seperti orientasi gambar dan konversi warna dari BGR ke RGB, yang melibatkan identifikasi titik-titik penting pada setiap tangan. Dengan menggunakan *MediaPipe*, sistem mampu mendeteksi 21 landmark atau titik penting pada masing-masing tangan. Setiap titik ini diwakili oleh koordinat dua dimensi (x, y), dan kombinasi dari semua koordinat tersebut membentuk pola atau gerakan tangan yang spesifik.



Gambar 4. Koordinat tangan untuk deteksi gerakan

2. Menghitung *landmark*

Sebagai komponen kunci untuk melatih model dan memprediksi gerakan baru di masa depan, 21 landmark yang terdeteksi harus disimpan dengan format yang benar. *MediaPipe* memberikan landmark dalam bentuk objek JSON atau kamus Python dalam format tiga dimensi. Namun, untuk keperluan pelatihan model atau prediksi, diperlukan data dalam bentuk Array Landmark.

Pada proses ini dilakukan dengan iterasi pada semua landmark untuk menghitung koordinat yang akurat pada bingkai gambar utama. Koordinat tersebut kemudian diubah menjadi daftar Python sederhana dengan format [x, y], dan setiap koordinat ditambahkan ke dalam daftar lain yang berisi keseluruhan 21 koordinat tangan.

```python
def calc_landmark_list(image, landmarks) -> List:
    #: Getting image width & height
    image_width = image.shape[1]
    image_height = image.shape[0]

    landmark_point = []

    #: Keypoint
    for _, landmark in enumerate(landmarks.landmark):
        landmark_x = min(int(landmark.x * image_width), image_width - 1)
        landmark_y = min(int(landmark.y * image_height), image_height - 1)
        #: landmark_z = landmark.z

        landmark_point.append([landmark_x, landmark_y])

    return landmark_point
```

Gambar 5. Salah satu kode untuk menentukan koordinat dalam deteksi gerakan

3. Pra-Proses Landmark

Tahap berikutnya dalam penelitian ini yaitu pra-pemrosesan, sistem mengambil salinan dalam data asli array 2D untuk digunakan, kemudian mengubah semua koordinat menjadi koordinat relative sehingga menjadi frame independen. Sehingga akan mengambil koordinat relatif dari koordinat pergelangan tangan. Setelah proses tersebut mendapatkan semua koordinat relatif, kemudian meratakan semua data menggunakan NumPy. Sehingga mengubah array 2 dimensi menjadi array satu dimensi, serta membuat semua titik data antara nilai 0-1, sebagai format yang tepat untuk melatih model.

```python
def pre_process_landmark(landmark_list) -> List:
    temp_landmark_list = copy.deepcopy(landmark_list)

    #: Convert to relative coordinates
    base_x, base_y = 0, 0


for index, landmark_point in enumerate(temp_landmark_list):
    if index == 0:
       base_x, base_y = landmark_point[0], landmark_point[1]

        #: Overriding coordinates
    temp_list[index][0] = temp_landmark_list[index][0] - base_x
    temp_list[index][1] = temp_landmark_list[index][1] - base_y

  #: Convert into a one-dimensional list
  temp_landmark_list = list(itertools.chain.from_iterable(temp_list))

  #: Normalization (0 - 1)
  max_value = max(list(map(abs, temp_list)))

  def normalize_(n):
      return n / max_value

  temp_list = list(map(normalize_, temp_list))

  return temp_landmark_list
```

Gambar 6. Salah satu kode untuk pre-proses landmark

4. Pengumpulan Data

Pada langkah berikutnya, sistem yang telah menghasilkan array satu dimensi yang sudah dinormalisasi, kemudian bisa digunakan untuk mengumpulkan data dan memprediksi gerakan tangan yang baru. Selain itu juga, sistem dilengkapi dengan fitur untuk beralih antara dua mode, yaitu prediksi dan pencatatan.

Dalam proses pengumpulan data, pada penelitian ini, pada sistem yang digunakan mengaktifkan mode Logging sehingga dapat menginformasikan gerakan tangan serta proses manual pada sistem yang hasil landmark akan didapatkan melalui semua langkah pra-pemrosesan dan menyimpan dalam bentuk file CSV.

Index	0.x	0.y	1.x	1.y	1.x	2.y	2.x	3.y	4.x	4.y	5.x	5.y	6.x	6.y	7.x	7.y	8.x	8.y	9.x
0	0	0	-0.2605	-0.12605	-0.48739	-0.42017	-0.59664	-0.67227	-0.48739	-0.83193	-0.36134	-0.77311	-0.36134	-1	-0.34454	-0.7395	-0.33613	-0.60504	-0.15126
0	0	0	-0.24576	-0.15254	-0.4661	-0.42373	-0.60169	-0.67797	-0.5	-0.85593	-0.33898	-0.76271	-0.35593	-1	-0.33051	-0.74576	-0.31356	-0.59322	-0.12712
0	0	0	-0.24167	-0.11667	-0.475	-0.41667	-0.6	-0.68333	-0.49167	-0.86667	-0.36667	-0.75833	-0.38333	-1	-0.35833	-0.725	-0.34167	-0.59167	-0.15833
0	0	0	-0.23729	-0.13559	-0.4661	-0.41525	-0.60169	-0.66949	-0.5	-0.84746	-0.35593	-0.75424	-0.36441	-1	-0.33051	-0.73729	-0.32203	-0.58475	-0.13559
0	0	0	-0.26271	-0.12712	-0.49153	-0.39831	-0.61017	-0.66102	-0.52542	-0.85593	-0.38983	-0.74576	-0.38136	-1	-0.34746	-0.74576	-0.33898	-0.58475	-0.16949
1	0	0	-0.15	-0.10909	-0.22273	-0.28636	-0.18182	-0.45455	-0.08182	-0.52273	-0.20455	-0.51364	-0.23182	-0.70909	-0.23636	-0.82727	-0.23182	-0.92727	-0.1
1	0	0	-0.15421	-0.09813	-0.23364	-0.27103	-0.18692	-0.42523	-0.09346	-0.50467	-0.21028	-0.50935	-0.24299	-0.70561	-0.25234	-0.8271	-0.24299	-0.92991	-0.1028
1	0	0	-0.15814	-0.09767	-0.23256	-0.27907	-0.19535	-0.44186	-0.10233	-0.50698	-0.2093	-0.52093	-0.24186	-0.71163	-0.25116	-0.82791	-0.24651	-0.93023	-0.10698
1	0	0	-0.15138	-0.09633	-0.22477	-0.27064	-0.19266	-0.42661	-0.12385	-0.50917	-0.20642	-0.51376	-0.23853	-0.70642	-0.24312	-0.8211	-0.23853	-0.92202	-0.09633
1	0	0	-0.15668	-0.09677	-0.23041	-0.27189	-0.20737	-0.43318	-0.14286	-0.51613	-0.20276	-0.51613	-0.23963	-0.70968	-0.24885	-0.82488	-0.24885	-0.93088	-0.10138
2	0	0	-0.28467	-0.07299	-0.51825	-0.19708	-0.71533	-0.24818	-0.86861	-0.35766	-0.40876	-0.64234	-0.52555	-0.90511	-0.67153	-0.91241	-0.78832	-0.84672	-0.32117
2	0	0	-0.28058	-0.07194	-0.5036	-0.20863	-0.69784	-0.26619	-0.83453	-0.38129	-0.40288	-0.64748	-0.52518	-0.89928	-0.67626	-0.91367	-0.78417	-0.84892	-0.31655
2	0	0	-0.26619	-0.06475	-0.48201	-0.21583	-0.67626	-0.27338	-0.79137	-0.38849	-0.38129	-0.65468	-0.4964	-0.91367	-0.64029	-0.92086	-0.7482	-0.84892	-0.29496
2	0	0	-0.27536	-0.06522	-0.5	-0.2029	-0.69565	-0.26812	-0.82609	-0.3913	-0.39855	-0.64493	-0.51449	-0.9058	-0.66667	-0.92754	-0.77536	-0.86232	-0.31159
2	0	0	-0.27536	-0.06522	-0.50725	-0.21014	-0.7029	-0.26812	-0.81159	-0.39855	-0.3913	-0.65217	-0.51449	-0.9058	-0.66667	-0.92754	-0.78261	-0.86232	-0.30435
3	0	0	-0.15979	-0.10825	-0.26804	-0.29381	-0.30928	-0.45876	-0.24742	-0.58247	-0.2268	-0.52577	-0.29381	-0.74227	-0.34536	-0.87629	-0.37113	-1	-0.1134
3	0	0	-0.1641	-0.10769	-0.26667	-0.30256	-0.31282	-0.46154	-0.25128	-0.58462	-0.23077	-0.52821	-0.29744	-0.74872	-0.34359	-0.88205	-0.36923	-1	-0.11795
3	0	0	-0.1641	-0.11282	-0.26667	-0.30769	-0.31282	-0.46667	-0.25641	-0.59487	-0.2359	-0.52821	-0.30256	-0.74872	-0.34872	-0.88205	-0.37436	-1	-0.12308
3	0	0	-0.15464	-0.10825	-0.26289	-0.30412	-0.31443	-0.45876	-0.26289	-0.58763	-0.2268	-0.52577	-0.28866	-0.74742	-0.34021	-0.88144	-0.36598	-1	-0.1134
3	0	0	-0.15625	-0.10417	-0.26042	-0.30208	-0.30729	-0.45833	-0.25	-0.57292	-0.22396	-0.52604	-0.29167	-0.75	-0.34375	-0.88021	-0.36458	-1	-0.10938
4	0	0	-0.27083	-0.09722	-0.4375	-0.33333	-0.40278	-0.56944	-0.23611	-0.66667	-0.43056	-0.60417	-0.5	-0.86111	-0.52083	-0.75694	-0.47917	-0.61806	-0.27778
4	0	0	-0.25874	-0.08392	-0.41958	-0.34965	-0.37063	-0.58042	-0.21678	-0.67133	-0.41958	-0.61538	-0.4965	-0.87413	-0.5035	-0.78322	-0.46154	-0.64336	-0.27273
4	0	0	-0.25	-0.09722	-0.40972	-0.33333	-0.375	-0.56944	-0.21528	-0.67361	-0.40972	-0.61111	-0.47917	-0.875	-0.49306	-0.76389	-0.45139	-0.61806	-0.26389
4	0	0	-0.25874	-0.09091	-0.42657	-0.34965	-0.38462	-0.58741	-0.2028	-0.66434	-0.42657	-0.60839	-0.4965	-0.87413	-0.5035	-0.76923	-0.45455	-0.62937	-0.27972
4	0	0	-0.25	-0.09028	-0.42361	-0.34028	-0.38889	-0.58333	-0.20833	-0.65278	-0.41667	-0.61111	-0.48611	-0.86111	-0.49306	-0.74306	-0.45139	-0.61111	-0.27083
5	0	0	-0.24051	-0.07595	-0.44304	-0.20886	-0.56962	-0.36709	-0.57595	-0.53797	-0.37975	-0.46835	-0.50633	-0.62025	-0.5443	-0.61392	-0.5443	-0.56329	-0.26582
5	0	0	-0.24684	-0.06329	-0.4557	-0.20886	-0.59494	-0.36709	-0.60127	-0.53797	-0.39873	-0.46203	-0.52532	-0.62025	-0.56962	-0.62658	-0.57595	-0.58228	-0.27848
5	0	0	-0.23899	-0.06289	-0.44654	-0.20126	-0.58491	-0.3522	-0.59119	-0.51572	-0.38365	-0.45912	-0.50943	-0.61635	-0.55346	-0.61006	-0.56604	-0.55975	-0.27044
5	0	0	-0.24359	-0.05128	-0.46154	-0.19231	-0.59615	-0.34615	-0.60256	-0.51923	-0.39744	-0.44872	-0.51923	-0.59615	-0.57051	-0.60256	-0.57692	-0.5641	-0.28205

Gambar 7. Hasil dari pemrosesan data

5. Fase Pelatihan

Pada penelitian ini, hasil dari proses pengumpulan data digunakan untuk melatih model dengan menggunakan metode Sequential Keras Tensorflow [11] sebagai sistem yang digunakan untuk melatih Jaringan Neural.

```
# Model Building
model = tf.keras.models.Sequential([
    tf.keras.layers.Input((21 * 2, )),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(20, activation='relu'),
    tf.keras.layers.Dropout(0.4),
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(NUM_CLASSES, activation='softmax')
])


# Model compilation
model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)


# Feeding data to the Model
model.fit(
    X_train,
    y_train,
    epochs=1000,
    batch_size=128,
    validation_data=(X_test, y_test),
    callbacks=[cp_callback, es_callback]
)
```

Gambar 7. Kode untuk proses pelatihan data

6. Fase Pengujian

Pada tahapan terakhir pada penelitian ini, yaitu proses menganalisa gambar dengan menggunakan sistem deteksi tangan dengan melakukan proses ulang untuk pengumpulan data yang sama dengan menggabungkan array landmark dalam kumpulan data, kemudian di proses array dari setiap frame ke Model Jaringan Syaraf Tiruan yang telah dilatih untuk memprediksi kelas yang sesuai dari tanda atau isyarat tangan yang dilakukan.

Kemudian, hasil dari array tersebut dikirim ke Model Jaringan Syaraf Tiruan atau Otak Proyek, dengan membandingkan data yang diberikan sebelumnya. Jika hasil dari proses tersebut ditemukan kecocokan dengan kelas alfabet, maka indeks kelas alfabet yang terdeteksi, sehingga akan memudahkan untuk menemukan alfabet. Setelah kita mendapatkan indeks kelas, kita dapat dengan mudah menemukan alfabetnya.

Kesimpulan

Sistem deteksi bahasa isyarat memiliki potensi besar untuk merevolusi komunikasi dan mendorong inklusivitas bagi pengguna bahasa isyarat. Teknologi ini dapat menjembatani kesenjangan antara bahasa isyarat dan bahasa lisan/tulisan, memungkinkan komunikasi yang lebih lancar dalam berbagai situasi. Dengan mengatasi hambatan komunikasi, sistem ini memberdayakan pengguna bahasa isyarat di berbagai sektor, seperti pendidikan, pekerjaan, dan layanan kesehatan, sehingga menciptakan masyarakat yang lebih inklusif dan setara.

Selain itu, MediaPipe dengan modul Hand Pose Estimation-nya, menyediakan kemampuan untuk mengidentifikasi titik-titik kunci (landmark) pada tangan dengan akurasi tinggi. Data landmark ini kemudian menjadi input penting bagi model machine learning yang bertugas menerjemahkan bahasa isyarat. Penggunaan MediaPipe memberikan keuntungan signifikan dalam penelitian ini, termasuk kemudahan integrasi, kinerja real-time yang optimal, dan ketersediaan model pre-trained yang mengurangi kebutuhan sumber daya komputasi.

Serta, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan sistem deteksi bahasa isyarat yang inklusif dan efektif. Meskipun masih terdapat batasan, seperti kemampuan sistem dalam mengenali satu huruf untuk setiap input, penelitian ini menjadi landasan penting untuk pengembangan sistem yang lebih canggih di masa depan. Pemanfaatan MediaPipe dalam penelitian ini juga menunjukkan potensi besar teknologi ini dalam mendukung pengembangan aplikasi-aplikasi berbasis visi komputer, khususnya dalam konteks penerjemahan bahasa isyarat.

Sehingga, pengembangan sistem deteksi bahasa isyarat yang berkelanjutan, melalui riset dan kolaborasi, sangat penting untuk mengoptimalkan potensinya. Solusi ini menjadi fondasi bagi pengembangan antarmuka pengguna berbasis pengenalan bahasa isyarat, yang dapat diadaptasi untuk melatih isyarat-isyarat baru di masa mendatang. Inovasi dan upaya yang konsisten akan membuka jalan bagi komunikasi yang inklusif dan efektif di era modern

Referensi

[1] I. B. A. Peling, I. M. P. A. Ariawan, and G. B. Subiksa, "Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL)," *J. Krisnadana*, vol. 3, no. 2, pp. 90–100, 2024, doi: 10.58982/krisnadana.v3i2.534.

[2] D. A. Primadiansyah *et al.*, "Literature Review : Implementasi Machine Learning Dalam Pengenalan Bahasa Isyarat Indonesia (BISINDO)," vol. 2, no. 5, pp. 821–824, 2024.

[3] N. Alexander, R. B. Widodo, and W. Swastika, "Penggunaan Machine Learning Dalam Klasifikasi Bahasa Isyarat BISINDO Menggunakan Kamera," *Pros. Semin. Nas. Inform. Sist. Inf.*, vol. 3, no. 1, pp. 11–26, 2023.

[4] Agus Nugroho, "Deteksi Bahasa Isyarat Bisindo Menggunakan Metode Machine Learning," *J. Process.*, vol. 18, no. 2, pp. 152–158, 2023, doi: 10.33998/processor.2023.18.2.1380.

[5] F. Rachardi, "Deteksi Gambar Gestur Kosakata Bahasa Isyarat Indonesia dengan Convolutional Neural Network," *Institutional Repos. UIN Syarif Hidayatullah*, p. 192, 2020.

[6] S. N. Budiman, S. Lestanti, H. Yuana, and B. N. Awwalin, "SIBI (Sistem Bahasa Isyarat Indonesia) berbasis Machine Learning dan Computer Vision untuk Membantu Komunikasi Tuna Rungu dan Tuna Wicara," *J. Teknol. dan Manaj. Inform.*, vol. 9, no. 2, pp. 119–128, 2023, doi: 10.26905/jtmi.v9i2.10993.

[7] I. Suyudi, S. Sudadio, and S. Suherman, "Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model Random Forest dan Multinomial Logistic Regression," *J. Ilmu Siber dan Teknol. Digit.*, vol. 1, no. 1, pp. 65–80, 2023, doi: 10.35912/jisted.v1i1.1899.

[8] A. R. Ardiansyah, A. H. Nur'azizan, and R. Fernandis, "Implementasi Deteksi Bahasa Isyarat Tangan Menggunakan OpenCV dan MediaPipe," *Stain. (Seminar Nas. Teknol. Sains)*, vol. 3, no. 1, pp. 331–337, 2024.

[9] N. Anam, "Sistem Deteksi Simbol Pada Sibi (Sistem Isyarat Bahasa Indonesia) Menggunakan Mediapipe Dan Resnet-50," 2022.

[10] R. Kumar, A. Bajpai, and A. Sinha, "Mediapipe and CNNs for Real-Time ASL Gesture Recognition." 2023, doi: 10.48550/arXiv.2305.05296.

[11] A. H. Gustsa and G. S. Permadi, "Sistem Deteksi Bahasa Isyarat Secara Realtime Dengan Tensorflow Object Detection dan Python Menggunakan Metode Convolutional Neural Network,"

Irfan Nurdiansyah S.Kom., M.kom.
Universitas Dian Nusantara
Fakultas Ilmu Komputer
Jl.Tanjung Duren Barat 2 No.1

Reni Utami S.SI., M.Kom.
Universitas Dian Nusantara
Fakultas Ilmu Komputer
Jl.Tanjung Duren Barat 2 No.1

Muchamad Sandy S.Kom., M.M.SI
Universitas Dian Nusantara
Fakultas Ilmu Komputer
Jl.Tanjung Duren Barat 2 No.1