

Sistem Pengenalan Gerak Bahasa Isyarat Dengan *Colored Motion History Image* dan *Convolutional Neural Network*

Haiqal Ramanizar Al Fajri¹, Jayanta², Bambang Tri Wahyono³

Program Studi Informatika / Fakultas Ilmu Komputer

Universitas Pembangunan Nasional Veteran Jakarta

Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

haiqalramanizaralfajri@gmail.com¹, anta.jayanta@gmail.com², bambangtriwahyono@upnvj.ac.id³

Abstrak. Penelitian ini dilakukan untuk membuat sistem pengenalan bahasa isyarat untuk mengenali gerakan bahasa isyarat pada sistem BISINDO. Bahasa isyarat merupakan metode berkomunikasi bagi penyandang tunarungu memahami dan menyampaikan informasi yang diterima menggunakan gerak tubuh. Dalam penelitian ini, penulis menggunakan metode CNN untuk melakukan proses pengenalan gerak. Selain itu, dalam merepresentasikan gerakan dari video menjadi satu gambar penulis menggunakan metode *Colored MHI*. Metode *Colored MHI* melakukan pengubahan warna yang dilakukan oleh MHI yang menggunakan skala abu-abu menjadi format warna RGB. Data didapatkan melalui pengambilan gambar video pada 15 subjek dengan 5 kelas gerakan dan menghasilkan total 450 data. Data video yang telah didapat dilakukan *cropping*, diekstraksi menjadi satu gambar dengan metode *Colored MHI*. Hasil pembuatan model CNN dengan data latih, diuji dengan data uji yang telah melalui tahap *Colored MHI*. Hasil penelitian ini menunjukkan metode CNN dan *Colored MHI* mendapat akurasi dan *loss* sebesar 0.8533 dan 0.4741.

Kata Kunci: *Convolutional Neural Network* (CNN), *Motion History Image* (MHI), Bahasa Isyarat, BISINDO.

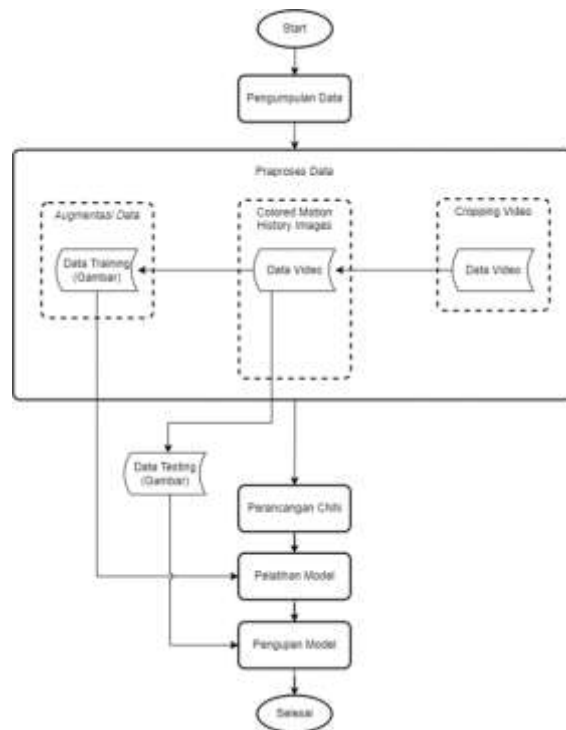
1 Pendahuluan

Dalam berkomunikasi, bahasa menjadi perantara antara satu individu dengan individu lain dalam memahami satu sama lain. Bagi penyandang tunawicara menggunakan bahasa isyarat untuk berkomunikasi. Bahasa isyarat merupakan cara berkomunikasi menggunakan gerak tubuh, bibir, dan bukan suara. Kelompok utama yang menggunakan bahasa isyarat ini untuk berkomunikasi adalah dari kelompok penyandang tunarungu [1]. Bahasa Isyarat Indonesia (BISINDO) merupakan bahasa alami yang digunakan penyandang tunarungu dalam bahasa pergaulan sehari-hari [2]. Namun masyarakat awam kurang memahami dengan baik terkait sistem bahasa isyarat tersebut sehingga menjadi hambatan dalam berkomunikasi dengan penyandang tunawicara. Untuk mengatasi hal tersebut dibutuhkan solusi yang dapat mengidentifikasi arti dari setiap isyarat dari sistem yang ada. Perkembangan teknologi saat ini memungkinkan identifikasi tangkapan citra dapat dilakukan dengan lebih baik. Identifikasi tersebut dapat dilakukan dengan metode klasifikasi pada citra yang ditangkap. Citra digital merupakan barisan bilangan yang setiap bilangannya mewakili suatu bit [3].

Dalam melakukan klasifikasi pada citra terdapat metode yang dapat digunakan di antaranya metode *Convolutional Neural Network* (CNN). *Convolutional Neural Network* (CNN) merupakan metode yang memiliki arsitektur jaringan yang berjumlah hingga ratusan layer. Proses yang dilakukan CNN adalah dengan memproses citra melalui *network* layer sehingga menghasilkan *output* pada kelas tertentu. Setiap layer akan menggunakan *input* yang diberikan untuk menghasilkan *output* yang dapat digunakan pada layer selanjutnya. Pada awal *network*, fitur-fitur dihasilkan secara sederhana. Fitur-fitur tersebut dapat berupa warna, kecerahan (*brightness*), dan tepi (*edges*). Lalu, pada tahap selanjutnya *network* akan memberikan hasil fitur yang lebih kompleks [4].

Selain itu, untuk melakukan pengolahan video berupa gerakan bahasa isyarat sehingga mendapatkan satu gambar tunggal yang bisa diolah dapat menggunakan metode *Motion History Image* (MHI). *Motion History Image* (MHI) merupakan metode representasi gerakan dari gambar bergerak atau video dapat direpresentasikan dalam satu gambar yang berisi aliran gerakan serta bagian yang bergerak dalam satu video [5]. Pada penelitian penggunaan improvisasi dari MHI konvensional dengan melakukan representasi gambar hasil MHI pada RGB *channel* menunjukkan peningkatan akurasi dari 73.4% pada MHI konvensional menjadi 84% pada *Colored MHI* atau MHI improvisasi pada RGB *channel* [6]. Selain itu, pada penelitian mengenai pengenalan bahasa isyarat pada perangkat seluler menggunakan metode CNN menunjukkan hasil performa sebesar 95,13% dengan *success rate* sebesar 100% [7]. Sehingga dari beberapa penelitian tersebut, penelitian ini melakukan metode CNN dengan *Colored MHI* untuk dapat melakukan identifikasi pengenalan gerak bahasa isyarat.

2 Metodologi Penelitian



Gambar. 1. Alur penelitian diawali dari identifikasi masalah, studi pustaka, pengumpulan data, praproses data, perancangan CNN, pelatihan model, dan pengujian model.

2.1 Pengumpulan Data

Pengumpulan data didapatkan melalui tatap muka dengan jumlah subjek sebanyak 15 orang. Pada setiap subjek, data diambil dalam bentuk video berupa gerakan isyarat. Pengambilan video dilakukan dengan 3 sudut pengambilan video dan 2 ketinggian kamera. Dalam pengambilan video, resolusi yang digunakan pada pengambilan video ini sebesar 1920×1080 dengan jumlah *frame* per detiknya sebanyak 25 *frame*. Latar belakang diatur menggunakan kain putih untuk menyeragamkan setiap video.

Gerakan isyarat yang ditetapkan pada penelitian ini berjumlah 5 kelas gerakan, yaitu gerakan untuk kata ‘terima kasih’, ‘makan’, ‘minum’, ‘tidur’, dan ‘sama-sama’. Total data yang akan digunakan berjumlah 450 data video.

2.2 Praproses Data

Tahapan ini terdiri dari proses cropping video, proses perubahan data video menjadi citra tunggal menggunakan metode *Colored Motion History Image (Colored MHI)*, dan tahap augmentasi data.

a) *Cropping Video*

Pada proses ini seluruh data video akan dipotong resolusinya menjadi ukuran yang sama dengan rasio 1:1 atau sebesar 1080×1080 *pixel*. Cropping digunakan untuk kebutuhan lapisan masukan dari CNN.

b) *Colored Motion History Image*

Hasil dari MHI konvensional berupa gambar tunggal dengan skala abu-abu statis, dengan *pixel* yang menunjukkan gerak terbaru dengan skala abu-abu yang lebih cerah [8]. Pada persamaan (1) dapat menentukan hasil gambar dari MHI konvensional [5].

$$H_t(x, y, t) = \begin{cases} \tau, & \text{if } \Psi(x, y, t) = 1 \\ \max(0, H_t(x, y, t - 1) - \delta), & \text{otherwise} \end{cases} \quad (1)$$

Hasil penerapan rumus persamaan di atas dapat diilustrasikan pada gambar 2.

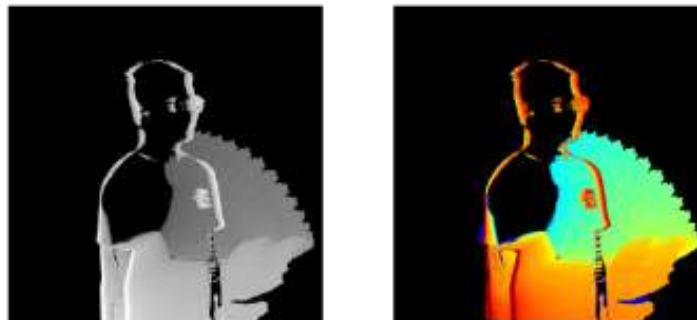


Gambar. 2. Ilustrasi MHI konvensional sebelum (tiga gambar kiri) dan setelah (tiga gambar kanan) penerapan rumus MHI konvensional.

Pada penelitian ini menggunakan improvisasi MHI berupa perubahan representasi warna dengan *rainbow pseudo-color* atau *Rainbow-MHI* [6] serta perubahan representasi warna dengan RGB channel atau RGB-MHI [8] untuk menghasilkan model terbaik yang bisa didapatkan. Pada *Rainbow-MHI* [6] menggunakan representasi warna *rainbow pseudo-color* sesuai dengan gambar 3 dan hasil penerapannya terdapat pada gambar 4.

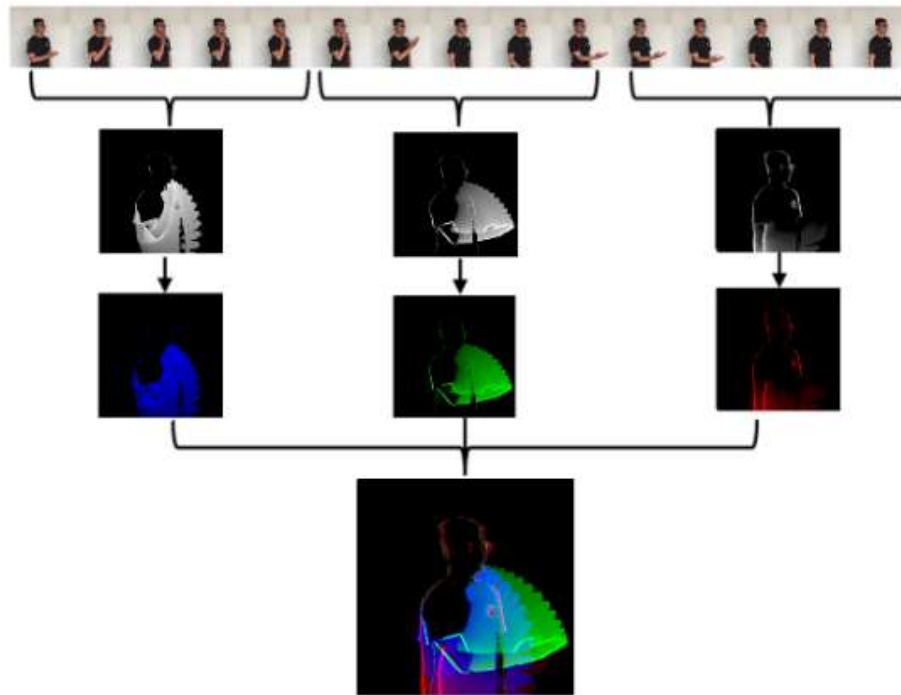


Gambar. 3. *Rainbow pseudo-colormap*.



Gambar. 4. Hasil penerapan *Rainbow-MHI*.

Pada RGB-MHI [8] menggunakan representasi RGB *channel* dengan membagi video menjadi 3 bagian, masing-masing bagian diubah menjadi gambar hasil MHI dan menjadi perwakilan dari ketiga nilai RGB *channel* yaitu merah, hijau, dan biru. Pada gambar 5 menunjukkan alur perubahan pada metode RGB-MHI.



Gambar. 5. Hasil penerapan *Rainbow-MHI*.

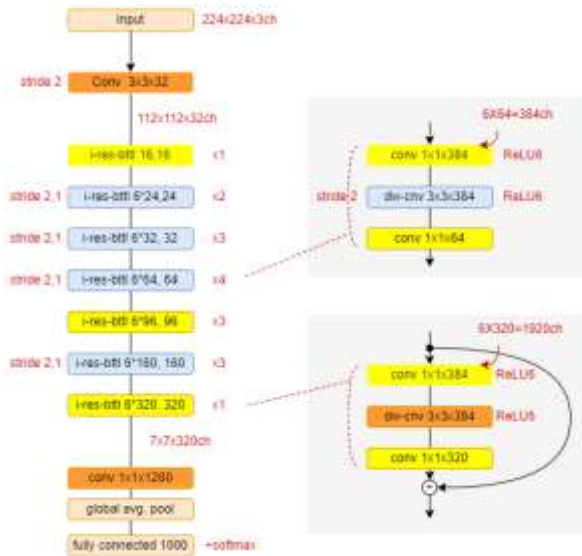
c) *Augmentasi Data*

Demi memperbanyak variasi dalam data citra, maka diperlukan proses augmentasi data. Data augmentasi dapat membantu dalam peningkatan akurasi pada *deep learning* [4]. Augmentasi data merupakan teknik pengurangan *overfitting* melalui penambahan ukuran dari dataset dengan cara seminimum mungkin [9]. Operasi yang digunakan pada tahap ini berupa *scaling* dengan nilai skala pada sumbu x dan sumbu y antara 0.7 sampai 1.1, *translation* dengan nilai skala antara -10% sampai 10%, dan *rotation* dengan nilai skala antara -10° sampai 10° . Penggunaan tersebut akan diterapkan secara acak pada data.

Pada tahap ini, data yang akan diproses hanya data yang akan dilatih pada tahap pelatihan model. Data uji yang telah dibagi hanya akan melalui tahap *Colored MHI*. Setelah tahap ini, maka data akan diteruskan pada model yang akan dibuat.

2.3 Perancangan CNN

Pada tahapan ini, model CNN akan dirancang hingga dapat digunakan pada tahap pelatihan model. Arsitektur yang digunakan pada penelitian ini adalah arsitektur MobilenetV2. MobilenetV2 merupakan model *deep learning* yang didasari pada CNN [10]. Arsitektur MobilenetV2 [11] dapat diilustrasikan pada gambar 6 berikut.



Gambar. 6. Arsitektur MobileNetV2.

Pada penelitian ini parameter yang akan ditetapkan sebagai parameter observasi yaitu pada ukuran *input* citra dan *batch size*. Ukuran *input* citra berupa nilai *pixel width* x *pixel height* dari gambar, dengan nilai *pixel width* sama dengan *pixel height* karena memiliki rasio sebesar 1:1. Nilai *batch size* dengan nilai yang ditentukan merupakan nilai yang habis membagi total data yang akan dilatih. Pada penelitian ini ukuran *input* citra yang diobservasi berukuran 64, 96, 128, 160, 192, dan 224, yang berarti sebagai contoh citra dengan *input* 64 berarti gambar dengan ukuran 64x64 *pixel*. Selanjutnya, untuk ukuran *batch* yang akan diobservasi berukuran 16, 32, 64, 96, dan 120, yang berarti sebagai contoh citra dengan *batch size* berukuran 16 dapat habis membagi total data sebanyak 960 data. Berikut ringkasan arsitektur dari model yang digunakan dalam observasi pada penelitian ini yang ditampilkan pada gambar 7.

```

Model: "mobilenetv2_1.00_rainbow_224-120-5"
=====
Layer (type)                Output Shape                Param #
-----
input_2 (InputLayer)        [(None, 224, 224, 3)]      0
mobilenetv2_1.00_224        (None, 7, 7, 1280)         2257984
(Functional)
global_average_pooling2d    (None, 1280)                0
(GlobalAveragePooling2D)
dropout (Dropout)           (None, 1280)                0
dense (Dense)                (None, 5)                   6405
=====
Total params: 2,264,389
Trainable params: 2,230,277
Non-trainable params: 34,112

```

Gambar. 7. Ringkasan arsitektur model.

3 Hasil dan Pembahasan

3.1 Pelatihan Model

Pada tahap ini, model akan dilatih menggunakan hasil proses augmentasi data di tahap praproses dengan total data sebanyak 1200 data gambar. Data ini terdiri dari 2 bagian yaitu data latih dan data validasi dengan ukuran sebesar 80% sebagai data latih dan 20% sebagai data validasi. Model akan dilatih menggunakan Adam optimizer sebagai *optimizer function* dan *categorical cross-entropy* sebagai *loss function*. Output metrik yang ditampilkan berupa *categorical accuracy* dan *loss value*. Seluruh skenario menggunakan *epoch* sebesar 100. Pada tabel 1—4 ditampilkan hasil akurasi dan *loss* pada tiap model berdasarkan skenario yang akan diobservasi berupa *batch size* dan ukuran *input* citra serta ekstraksi video menggunakan metode *Rainbow-MHI* dan *RGB-MHI*.

Tabel 1. Hasil pelatihan model menggunakan data *Rainbow-MHI*.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	<i>val loss</i>	1.2268	0.9426	0.7968	0.4287	0.331	0.5779
	<i>val acc</i>	0.7833	0.8542	0.8708	0.8958	0.9375	0.875
32	<i>val loss</i>	0.9844	1.0119	0.7901	0.4992	0.3612	0.2588
	<i>val acc</i>	0.875	0.8839	0.875	0.9062	0.9419	0.933
64	<i>val loss</i>	1.005	1.1665	0.9219	0.6977	0.3639	0.8333
	<i>val acc</i>	0.8594	0.8125	0.8698	0.8958	0.8958	0.8802
96	<i>val loss</i>	1.5094	1.0269	0.9601	0.4645	0.7519	0.5749
	<i>val acc</i>	0.7917	0.8594	0.8646	0.9010	0.875	0.9062
120	<i>val loss</i>	0.923	0.8363	0.9265	0.4961	0.3057	0.4767
	<i>val acc</i>	0.8417	0.875	0.85	0.9208	0.9083	0.8958

Tabel 2. Hasil pelatihan model menggunakan data metode *RGB-MHI*.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	<i>val loss</i>	0.7692	1.0848	0.47	0.5821	0.2977	0.301
	<i>val acc</i>	0.8583	0.8625	0.9083	0.9083	0.925	0.9458
32	<i>val loss</i>	0.7292	0.4934	0.6061	0.2953	0.354	0.3421
	<i>val acc</i>	0.8884	0.9062	0.8973	0.9286	0.942	0.9241
64	<i>val loss</i>	0.7897	0.7798	0.372	0.5377	0.4244	0.3104
	<i>val acc</i>	0.875	0.8802	0.9115	0.9219	0.9219	0.9323
96	<i>val loss</i>	0.8848	0.8073	0.6583	0.3803	0.193	0.3804
	<i>val acc</i>	0.8906	0.8698	0.901	0.9219	0.9427	0.9219
120	<i>val loss</i>	1.1687	0.7374	0.4432	0.5713	0.2947	0.3147
	<i>val acc</i>	0.8417	0.8833	0.9166	0.8917	0.95	0.9458

Tabel 3. Hasil pelatihan model pada epoch terbaik menggunakan data *Rainbow-MHI*.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	<i>val loss</i>	0.6210	0.5129	0.5167	0.3505	0.2961	0.4288
	<i>val acc</i>	0.8375	0.8375	0.8583	0.8667	0.9333	0.8667
	<i>epoch</i>	33	30	31	49	82	31
32	<i>val loss</i>	0.5653	0.4961	0.4527	0.4155	0.3028	0.2582
	<i>val acc</i>	0.8571	0.8125	0.8661	0.8884	0.9286	0.933
	<i>epoch</i>	89	12	42	65	68	99
64	<i>val loss</i>	0.5285	0.6686	0.5433	0.3165	0.2613	0.4998
	<i>val acc</i>	0.849	0.7552	0.8125	0.9010	0.9010	0.8542
	<i>epoch</i>	64	17	22	47	62	13
96	<i>val loss</i>	0.6575	0.4849	0.4780	0.3987	0.3776	0.4174
	<i>val acc</i>	0.7917	0.8646	0.8594	0.8906	0.8646	0.875
	<i>epoch</i>	30	65	29	47	21	29
120	<i>val loss</i>	0.6311	0.5082	0.5131	0.4140	0.2523	0.4271
	<i>val acc</i>	0.8417	0.8375	0.8333	0.9042	0.9167	0.8958
	<i>epoch</i>	90	35	17	42	48	44

Tabel 4. Hasil pelatihan model pada epoch terbaik menggunakan data RGB-MHI.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	val loss	0.6982	0.5156	0.2991	0.3461	0.2363	0.2493
	val acc	0.8625	0.8167	0.9125	0.9083	0.9250	0.9417
	epoch	82	40	57	39	69	65
32	val loss	0.4492	0.3335	0.3714	0.2632	0.2624	0.2405
	val acc	0.8705	0.8929	0.8929	0.9241	0.9152	0.9241
	epoch	95	73	87	59	34	31
64	val loss	0.5322	0.3221	0.3389	0.3817	0.3305	0.2564
	val acc	0.8802	0.8906	0.9168	0.9062	0.8906	0.9375
	epoch	33	32	56	27	20	39
96	val loss	0.5715	0.4288	0.3326	0.2817	0.1538	0.2844
	val acc	0.8750	0.8438	0.8750	0.8854	0.9583	0.9167
	epoch	57	18	17	46	43	11
120	val loss	0.7274	0.5214	0.3352	0.3710	0.2357	0.2431
	val acc	0.8125	0.8375	0.8875	0.8917	0.9417	0.9250
	epoch	30	16	16	27	51	20

Pada kedua metode RGB-MHI dan *Rainbow*-MHI dari tabel di atas dapat diketahui bahwa model dengan performa terbaik yang bisa diperoleh pada data dari metode RGB-MHI dengan konfigurasi *input* sebesar 192 dan *batch size* sebesar 96 di *epoch* ke-43. Model ini mendapatkan tingkat akurasi tertinggi sebesar 0.9583 dan mendapatkan tingkat *loss* yang rendah juga sebesar 0.1538.

3.2 Pengujian Model

Pada tahap pengujian, model akan divalidasi performanya menggunakan data uji. Data uji yang digunakan adalah data dari 5 subjek dengan total data sebanyak 150 data gambar yang tidak melalui tahap augmentasi. Performa akan dilihat berdasarkan nilai akurasi dan *loss* pada model setelah dilakukan *fit* pada data uji. Pada tabel 5—9 berikut menjelaskan hasil pengujian pada setiap model dalam skenario dengan metode ekstraksi video *Rainbow*-MHI dan RGB-MHI.

Tabel 5. Hasil pengujian model menggunakan data metode *Rainbow*-MHI.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	val loss	1.2809	1.0281	0.9635	1.0497	0.9537	0.9358
	val acc	0.72	0.8267	0.84	0.8333	0.8533	0.8333
32	val loss	1.2921	1.4578	0.9498	1.0759	0.9571	0.8158
	val acc	0.8266	0.7867	0.8733	0.8333	0.86	0.86
64	val loss	1.309	1.3593	1.3124	0.8508	0.76	0.8153
	val acc	0.8266	0.8267	0.8733	0.8467	0.8466	0.8266
96	val loss	2.202	1.2922	1.5060	0.9753	1.0629	1.0555
	val acc	0.7266	0.8399	0.8	0.84	0.84	0.8266
120	val loss	1.9574	1.4337	1.5638	0.9188	1.1775	0.8084
	val acc	0.7266	0.78	0.8267	0.8199	0.8066	0.8266

Tabel 6. Hasil pengujian model menggunakan data metode RGB-MHI.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	val loss	1.3718	1.2423	1.2135	1.1862	1.7761	1.0105
	val acc	0.8067	0.88	0.8266	0.88	0.8133	0.82
32	val loss	1.2472	1.6252	1.6041	0.8396	0.8252	0.9773
	val acc	0.7933	0.8067	0.8266	0.86	0.84	0.8733
64	val loss	1.5	1.7118	0.8608	0.9228	1.0537	1.0534
	val acc	0.8	0.8133	0.8399	0.8533	0.8467	0.82
96	val loss	1.8973	1.4497	1.08	1.0017	1.3588	0.8798
	val acc	0.7533	0.82	0.8333	0.8466	0.8533	0.8667

120	val loss	1.8529	1.7598	1.0527	1.4357	1.0705	1.2760
	val acc	0.7733	0.7799	0.8333	0.8067	0.8667	0.8133

Tabel 7. Hasil pengujian model di epoch terbaik menggunakan data metode *Rainbow*-MHI.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	val loss	1.0120	0.8175	0.6199	0.7735	0.8449	0.6731
	val acc	0.7600	0.7800	0.8200	0.8067	0.8467	0.7933
	epoch	33	30	31	49	82	31
32	val loss	1.0108	0.9335	0.5657	1.0090	0.8633	0.8160
	val acc	0.7867	0.7600	0.8133	0.7800	0.8467	0.8533
	epoch	89	12	42	65	68	99
64	val loss	0.9618	1.0619	0.5132	0.8352	0.4741	0.7427
	val acc	0.7933	0.7133	0.8400	0.8267	0.8533	0.7600
	epoch	64	17	22	28	62	13
96	val loss	0.8185	0.7977	0.6654	0.8345	0.5413	0.8190
	val acc	0.7400	0.8600	0.7933	0.7867	0.7933	0.8200
	epoch	30	65	29	47	21	29
120	val loss	1.0582	1.1921	0.9269	0.5989	1.0525	0.7402
	val acc	0.7467	0.7133	0.7666	0.8467	0.7933	0.8067
	epoch	90	35	17	42	48	44

Tabel 8. Hasil pengujian model di epoch terbaik menggunakan data metode *RGB*-MHI.

<i>Batch Size</i>		<i>Ukuran Input</i>					
		64	96	128	160	192	224
16	val loss	1.2914	0.7169	0.6208	0.8674	1.2880	0.7572
	val acc	0.8067	0.8133	0.8600	0.7933	0.7867	0.8333
	epoch	82	40	57	39	69	65
32	val loss	0.7698	0.8851	0.8676	0.6145	0.7111	0.7729
	val acc	0.7933	0.7933	0.8200	0.8800	0.8267	0.8600
	epoch	95	73	87	59	34	31
64	val loss	1.1162	1.3076	0.7310	0.7030	0.6602	0.9207
	val acc	0.7667	0.7467	0.8600	0.7867	0.8200	0.8267
	epoch	33	32	56	27	20	39
96	val loss	1.4750	1.0200	0.6803	1.3966	1.1684	0.6504
	val acc	0.7333	0.7733	0.8467	0.7733	0.8267	0.7933
	epoch	57	18	17	46	43	11
120	val loss	1.0100	1.5794	0.6533	1.0475	0.8975	0.9208
	val acc	0.7267	0.7000	0.7933	0.7867	0.8200	0.7733
	epoch	30	16	16	27	51	20

Dari kedua metode *RGB*-MHI dan *Rainbow*-MHI, didapatkan bahwa data dari metode yang terbaik pada proses pengujian tidak sama dengan ketika pelatihan. Pada proses pengujian ini model terbaik didapatkan pada data dengan metode *Rainbow*-MHI serta menggunakan konfigurasi *input* sebesar 192 dan *batch size* sebesar 64.

4 Simpulan dan Saran

4.1 Simpulan

Berdasarkan hasil penelitian dan pembahasan yang dilakukan penulis terkait sistem pengenalan bahasa isyarat dengan menggunakan metode *Colored Motion History Image (Colored MHI)* dan *Convolutional Neural Network (CNN)*, dapat disimpulkan menjadi sebagai berikut:

- Penerapan *Colored MHI* dan *CNN* digunakan untuk melakukan pengenalan bahasa isyarat dari video dapat dilakukan dengan tahapan yaitu, melakukan pengambilan data sebanyak 450 data video mencakup 5 gerakan bahasa isyarat, 15 subjek, 2 ketinggian kamera, dan 3 sudut pengambilan video. Kemudian, data video diubah menjadi gambar menggunakan 2 metode *Colored MHI*. Selanjutnya, pembagian pada data gambar yang

didapatkan menjadi 5 subjek untuk proses pengujian dan 10 subjek untuk pelatihan. Dari data untuk proses pelatihan akan dilakukan augmentasi, hingga total data pelatihan sebanyak 1200 data. Data ini dibagi lagi sebesar 80% data latih dan 20% data validasi. Data latih hasil pembagian tersebut digunakan untuk dilatih pada CNN dengan arsitektur MobilenetV2.

- b) Tingkat akurasi dan loss terbaik menggunakan metode CNN sebagai metode untuk mengenali gerakan bahasa isyarat dari video yang telah diubah menjadi gambar menggunakan *Colored MHI* mendapatkan nilai sebesar 0.9583 dan 0.1538 pada proses pelatihan serta sebesar 0.8533 dan 0.4741 pada proses pengujian. Model tersebut menggunakan metode CNN pada arsitektur MobilenetV2 dengan konfigurasi hyperparameter, yaitu pada input 64, 96, 128, 160, 192, dan 224, serta pada batch size 16, 32, 64, 96, dan 120 dengan epoch yang digunakan sebanyak 100 kali. Data video sebelumnya diubah menjadi data gambar tunggal dengan skala warna RGB channel dengan menggunakan metode *Colored MHI* yang digunakan model pada metode CNN untuk dikenali gerakan bahasa isyaratnya.

4.2 Saran

Berdasarkan pembahasan yang dijelaskan, dapat diterapkan saran-saran berupa penambahan kelas yang akan diklasifikasi, menambah subjek data, menerapkan segmentasi pada data, menggunakan hyperparameter lain yang dapat digunakan, menerapkan implementasi pada perangkat *mobile*, dan meneliti terkait pengenalan bahasa isyarat secara *realtime*.

Referensi

- [1] N. Lakshita, "Belajar Bahasa Isyarat Untuk Anak Tunarungu (Menengah)." Javalitera, Yogyakarta, 2012.
- [2] D. Rahmawati, *Panduan Bahasa Isyarat untuk Pendamping Penyandang Tuli*. Tangerang: Albasih Aksara CV, 2018.
- [3] M. R. V. Aditya, N. L. Husni, D. A. Pratama, dan A. S. Handayani, "Penerapan Sistem Pengolahan Citra Digital Pendeteksi Warna pada Starbot," *J. Tek.*, vol. 14, no. 02, hal. 185–191, 2020.
- [4] W. Setiawan, *Deep Learning menggunakan Convolutional Neural Network: Teori dan Aplikasi*. Media Nusa Creative (MNC Publishing), 2021.
- [5] M. A. R. Ahad, *Motion History Images for Action Recognition and Understanding*. London: Springer London, 2013.
- [6] Q. Chun dan E. Zhang, "Human action recognition based on improved motion history image and deep convolutional neural networks," *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-Janua, hal. 1–5, 2018, doi: 10.1109/CISP-BMEI.2017.8302061.
- [7] P. Yugopuspito, I. Made Murwantara, dan J. Sean, "Mobile sign language recognition for Bahasa Indonesia using convolutional neural network," *ACM Int. Conf. Proceeding Ser.*, hal. 84–91, 2018, doi: 10.1145/3282353.3282356.
- [8] O. M. Sincan dan H. Y. Keles, "Using *Motion History Images* with 3D Convolutional Networks in Isolated Sign Language Recognition," hal. 1–14, 2021, [Daring]. Tersedia pada: <http://arxiv.org/abs/2110.12396>.
- [9] R. Z. Fadillah, A. Irawan, dan M. Susanty, "Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO)," vol. 8, no. 2, hal. 208–214, 2021.
- [10] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, dan J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustain. Cities Soc.*, vol. 66, no. December 2020, hal. 102692, 2021, doi: 10.1016/j.scs.2020.102692.
- [11] V. Lakshmanan, M. Görner, dan R. Gillard, *Practical Machine Learning for Computer Vision: End-to-End Machine Learning for Images*. 2021.