

# Klasifikasi American Sign Language Menggunakan Convolutional Neural Network

**Tino Israldi, Elin Haerani, Suwanto Sanjaya, Fadhilah Syafria**

Fakultas Sains dan Teknologi, Teknik Informatika, UIN Sultan Syarif Kasim, Riau, Indonesia

Email: <sup>1</sup>11850114693@students.uin-suska.ac.id, <sup>2</sup>elin.haerani@uin-suska.ac.id, <sup>3</sup>suwantosanjaya@uin-suska.ac.id,

<sup>4</sup>fadhilah.syafria@uin-suska.ac.id

Email Penulis Korespondensi: 11850114693@students.uin-suska.ac.id

Submitted: 23/11/2022; Accepted: 12/12/2022; Published: 30/12/2022

**Abstrak**—Komunikasi menjadi sebuah kebutuhan dari semua kalangan ataupun individu dikarenakan tiap manusia harus berkomunikasi dengan sekitar. Komunikasi mampu membuat kita mendapatkan banyak informasi kemudian bisa menjadi acuan dalam beradaptasi. Bahasa verbal yang digunakan dengan cara berbicara dengan mengeluarkan suara merupakan cara komunikasi individu, tetapi tidak semua individu bisa melakukan komunikasi dengan hal itu, terlebih ada beberapa individu yang memiliki kekurangan ketika mendengar. Karena kekurangan itu maka ada cara lain yang dapat digunakan yaitu melalui bahasa isyarat. Bahasa Isyarat yaitu bahasa yang biasanya dipergunakan oleh beberapa orang dengan keterbatasan dalam hal mendengar ataupun berbicara dan bahasa isyarat juga mempunyai standar, standar bahasa isyarat yang lumayan terkenal yaitu standar American Sign Language (ASL). Berbeda dengan bahasa yang umum digunakan, bahasa isyarat juga kerap kali kurang diketahui oleh kebanyakan orang dikarenakan minat kebanyakan orang dengan bahasa isyarat masih kurang banyak sehingga kebanyakan orang tak mampu memahami bahasa mereka. Bahasa isyarat memiliki jenis yang banyak, contohnya yaitu bahasa isyarat ASL dengan jari yang merangkaikan angka dan huruf. Dalam mengatasi permasalahan tersebut, jadi solusinya adalah dengan membuat suatu sistem yang dapat mengenali bahasa isyarat, sistem yang dikembangkan merupakan sistem yang memanfaatkan teknologi machine learning. Penelitian ini akan mengusulkan pendekatan klasifikasi ASL melalui proses preprocessing data dan model convolutional neural network. Model yang diusulkan dapat mengklasifikasikan gambar postur tangan ASL untuk dapat diterjemahkan menjadi alfabet. Hasil penelitian ini adalah nilai akurasi sebesar 99,8% yang didapatkan dari penggabungan proses preprocessing data dan model convolutional neural network.

**Kata Kunci:** American Sign Language; Augmentasi Data; Convolutional Neural Network; Machine Learning; Mobilenetv2

**Abstract**—Communicating is a necessity for all groups or individual because each individual should communicate with their surroundings. Communicating can also make us get information so that it can be used as a reference to be able to adapt. Verbal language used by speaking out loud is a way of communicating with individuals, but not all individuals can communicate with it, especially there are some individuals who have hearing limitations. Because of these limitations, another program that can be used is through sign language. Language requirements are languages that are usually used by people with disabilities in terms of hearing or speaking and sign language also has a fairly well-known sign language standard, namely the American Sign Language (ASL) standard. Unlike languages in the world, sign language is also often of little interest to most people because people's interest in sign language is still lacking so that most people are unable to understand their language. Sign language has many types, one of which is sign language by using hands to form letters and numbers. In overcoming these problems, the solution is to create a system that can be used to recognize sign language, the system developed is a system that used machine learning technology. This study will propose an ASL classification approach through data preprocessing and a convolutional neural network model. The proposed model can classify ASL hand posture images to be translated into the alphabet. The result of this study is an model with accuracy of 99.8% obtained from the process of merging preprocessing data and the convolutional neural network model.

**Keywords:** American Sign Language; Convolutional Neural Network; Data Augmentation; Machine Learning; Mobilenetv2

## 1. PENDAHULUAN

Bahasa Isyarat merupakan bahasa sehari-hari dipergunakan kebanyakan orang yang mengalami kesusahan ketika mendengarkan ataupun ketika bicara. Tidak seperti bahasa yang digunakan di dunia, bahasa isyarat juga kurang diminati kebanyakan orang dikarenakan minat kebanyakan orang pada bahasa isyarat masih kurang banyak maka dari itu kebanyakan orang tak mampu mengerti bahasa mereka. Di Indonesia sendiri tuna rungu, dan tuna wicara banyak terdapat di berbagai daerah. Pada tahun 2007 menurut data, ada 3.861 anak penderita disabilitas rungu, serta terdapat 16.335 anak penderita disabilitas wicara dan juga 7.632 anak penderita disabilitas rungu, wicara, dan netra [1]. Sebagian besar dari penderita ini menggunakan bahasa isyarat sebagai media dalam melakukan komunikasi dengan lingkungan sekitar. Namun juga dikarenakan minat masyarakat dengan bahasa isyarat masih kurang, karena itu orang-orang tidak mampu memahami ketika mereka menggunakan bahasa isyarat. Mereka yang menderita disabilitas rungu juga kerap kesulitan untuk memahami acara berita atau debat dikarenakan jarang atau kurangnya penafsir bahasa isyarat dan juga penafsir bahasa isyarat di negara Indonesia hanya sedikit, contohnya di provinsi Sumatera barat mereka mempunyai 8 penafsir bahasa isyarat untuk 1250 disabilitas rungu [2]. Di negara Indonesia peminat dalam mempelajari bahasa isyarat terbilang cukup tinggi, tetapi minimnya media dalam hal pembelajaran bahasa isyarat di negara Indonesia menjadi masalah yang membuat masyarakat kesulitan untuk mengartikan bahasa isyarat. Kurikulum pendidikan yang digunakan di Indonesia juga tidak mampu memfasilitasi kebutuhan penyandang disabilitas rungu maupun wicara [3]. Sebagai tambahan, ada lebih dari 300 bahasa isyarat berbeda yang ada dan digunakan oleh berbagai orang tuli dan tunarungu di seluruh dunia. Bahasa isyarat tidak mempertahankan sifat tata bahasa yang sama seperti yang diucapkan bahasa. Namun demikian, bahasa isyarat mempertahankan sifat linguistik yang mirip dengan

bahasa lisan. Bahasa Isyarat Amerika (ASL) merupakan bahasa isyarat paling umum digunakan di Amerika Serikat dan beberapa bahasa lainnya. Bahkan dengan penggunaan bahasa isyarat yang paling umum juga kerap disalahpahami karena masih terbilang banyak yang kurang mengetahui arti dari ASL yang merupakan bahasa isyarat standar. Kesalahan ini terjadi pada orang yang belum pernah belajar bahasa isyarat. Permasalahan tersebut membutuhkan solusi yang dapat membantu masyarakat mengenali bahasa isyarat standar ASL. Di dalam era teknologi yang semakin maju, masalah ini bisa diatasi dengan cara mengembangkan suatu sistem yang dapat mengenali bahasa isyarat. Sistem yang dikembangkan merupakan sistem yang dapat memanfaatkan teknologi *machine learning*.

Penelitian ini mengusulkan pendekatan klasifikasi ASL melalui *convolutional neural network*. Model yang diusulkan dapat mengklasifikasikan gambar postur tangan ASL untuk dapat diterjemahkan menjadi alphabet. Penelitian ini melanjutkan beberapa penelitian terdahulu seperti salah satu contoh yang terdapat pada penelitian Kulhandjian et al. yang dilakukan pada tahun 2019 dengan menggunakan metode *Doppler Radar* yang digabungkan dengan VGG kemudian mendapat model dengan akurasi sebesar 87.5% [4]. Pada penelitian selanjutnya pada tahun 2018 yang dilakukan oleh Jalal menghasilkan model klasifikasi yang menggunakan *pretrain* model *deep neural network* sehingga mendapat model dengan akurasi 99% [5]. Penelitian selanjutnya yang diselesaikan oleh Rao et al. pada tahun 2018 dalam melakukan klasifikasi dengan media kamera ponsel, pada proses klasifikasi tersebut memperoleh nilai akurasi sebesar 92.8% [6]. Dalam penelitian yang dilakukan oleh Das et al. pada tahun 2018 yaitu penelitian *static sign language recognition using convolutional neural network*, kemudian mendapat hasil 97.5% [7]. Peneliti Ameen et al. Pada tahun 2017 [8] juga menggunakan penelitian dengan klasifikasi ASL dengan *ConvNet* dengan pemanfaatan fitur *image intensity* dan kedalaman gambar. Tetapi dalam penelitian ini tak menambah tahap *preprocessing data*. Peneliti Tolentino et al. pada penelitiannya pada tahun 2019 [9] menggunakan model *CNN* dengan *filter 16, kernel 2x2, dan max pooling 2x2* untuk melakukan klasifikasi terhadap ASL. Hasil akurasi yang didapat dari penelitian ini sebesar 93.67%. Hal ini disebabkan karena memakai arsitektur *CNN* dengan model arsitektur yang sederhana, sehingga kemungkinan dalam meningkatkan akurasi modelnya meningkat ketika menggunakan arsitektur yang lebih rumit. Peneliti yang dilakukan Daroya et al. pada tahun 2019 [10] dalam penelitiannya memakai *Densely Connected Convolutional Neural Network (DenseNet)* dan mendapatkan model dengan akurasi 90.3%. tetapi pada penelitian ini tak menjelaskan masa *training* yang harus dilakukan dalam proses pembentukan model. Mengacu pada penelitian sebelumnya sehingga dalam penelitian ini, kami membahas masalah utama dari masalah klasifikasi tanda ASL yaitu adanya keterbatasan data ketika melakukan training model, untuk mengatasi masalah tersebut penelitian ini menggunakan proses augmentasi data [11] dengan melakukan beberapa proses augmentasi untuk memecahkan keterbatasan data di ASL masalah klasifikasi huruf. Selain itu, penelitian ini juga merancang model berbasis *convolutional neural network* yang mencapai akurasi klasifikasi dan dapat dilatih cepat dibandingkan dengan model lain yang ada. Kelebihan lainnya yaitu model ini *convolutional neural network* tidak memerlukan adanya tahapan *preprocessing data* lainnya dalam melakukan proses melakukan klasifikasi berbeda dengan model lainnya yang harus melalui beberapa tahapan *preprocessing data* lainnya, selain itu model *convolutional neural network* juga dapat melakukan klasifikasi tanpa adanya tambahan algoritma segmentasi atau teknik *transfer learning*. Oleh sebab itu dalam penelitian ini *convolutional neural network* digunakan dalam sebagai model dalam melakukan klasifikasi terhadap *american sign language* yang akan membantu pembaca memahami *american sign language* dan menghilangkan hambatan dalam berkomunikasi dengan *pengguna american sign language*.

## 2. METODOLOGI PENELITIAN

Alur kerja penelitian yang digunakan yaitu pengumpulan dataset yang digunakan, *preprocessing data, training model*, dan evaluasi model. Kemudian pada tahapan ini juga menjelaskan mengenai dataset yang digunakan dan proses yang dilakukan sehingga bisa menghasilkan sebuah model *machine learning* yang dapat melakukan klasifikasi terhadap *american sign language*. Berikut merupakan tahapan dan metode yang dilakukan dalam penelitian ini.

### 2.1 Tahapan penelitian

#### a. Pengumpulan data

Tahap ini sangat penting dilakukan sebelum memulai proses analisa dalam suatu penelitian. Pada proses kali ini penulis mengumpulkan data dengan tujuan untuk dilatih supaya bisa membaca bahasa isyarat berdasarkan ASL (*american sign language*). Bahasa ini didapatkan melalui gerakan tangan yang akan mewakili masalah banyak-kelas dengan 24 kelas huruf (untuk huruf tidak akan termasuk J dan Z karena pada huruf ini membutuhkan gerakan). Bentuk pada kumpulan data nantinya akan disesuaikan pada pola supaya sesuai dengan *MNIST* klasik. Untuk setiap kasus pada pelatihan dan pengujian akan mewakili 26 label yaitu (0-25) sebagai peta satu-ke-satu untuk setiap huruf alfabet A-Z (untuk huruf tidak akan termasuk 9=J dan 25= Z karena pada huruf ini membutuhkan gerakan). Data pelatihan yang akan penulis gunakan berjumlah 27.455 kasus dan untuk data uji berjumlah 7172 kasus, kira-kira setengah ukuran *MNIST* standar tetapi sebaliknya serupa dengan baris *header label, pixel1,pixel2....pixel784* yang mewakili gambar 28x28 piksel tunggal dengan nilai skala abu-abu antara 0-255. Data gambar merupakan gerakan tangan asli yang mewakili beberapa pengguna yang terus mengulangi gerakan dengan latar belakang yang berbeda. Data *MNIST* bahasa isyarat berasal dari perluasan sejumlah kecil (1704) gambar berwarna yang disertakan sebagai tidak dipotong di sekitar wilayah tangan yang diinginkan. Untuk membuat data baru, saluran gambar digunakan berdasarkan

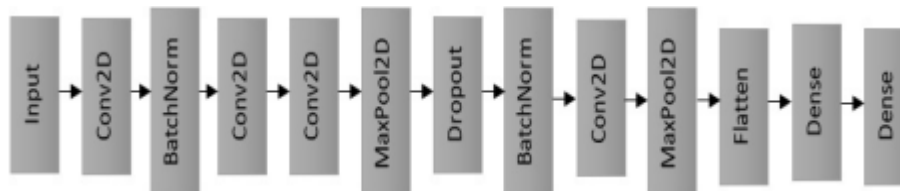
*ImageMagick* dan menyertakan pemangkasan ke tangan saja, penskalaan abu-abu, pengubahan ukuran, dan kemudian membuat setidaknya 50+ variasi untuk memperbesar kuantitas. Strategi modifikasi dan perluasannya adalah *filter* ('Mitchell', 'Robidoux', 'Catrom', 'Spline', 'Hermite'), bersama dengan pikselasi acak 5%, kecerahan/kontras +/- 15%, dan akhirnya rotasi 3 derajat. Karena ukuran gambar yang kecil, modifikasi ini secara efektif mengubah resolusi dan pemisahan kelas dengan cara yang menarik dan dapat dikontrol.

#### b. Preprocessing data

*Preprocessing data* merupakan sebuah proses yang sangat penting untuk dilakukan guna mempermudah proses analisa data. Proses data preprocessing ini bertujuan untuk menyeragamkan formatnya ke dalam satu set data dan menyeleksi data dari berbagai sumber. Untuk kasus kali ini penulis menerapkan teknik augmentasi data dalam proses data preprocessing. Tujuan utama dari proses ini adalah untuk melakukan modifikasi pada data asli. Teknik ini sangat efektif digunakan jika memiliki kekurangan ketersediaan data pelatihan untuk melatih model yang telah ditetapkan. Dengan menerapkan hanya beberapa modifikasi ini ke data pelatihan penulis, penulis dapat dengan mudah menggandakan atau melipatgandakan jumlah contoh pelatihan dan membuat model yang sangat kuat. Selain itu, augmentasi data bertindak sebagai model regularizer yang membantu mengurangi dan mencegah masalah overfitting selama pelatihan model [12]. Pada penelitian ini diterapkan augmentasi seperti memutar secara acak beberapa gambar latih sebesar 10 derajat, secara acak memperbesar 10% beberapa gambar latih, melakukan pergeseran gambar secara horizontal secara acak sebesar 10% dari lebarnya, secara acak menggeser gambar secara vertikal sebesar 10% dari ketinggiannya. Pada penelitian ini tidak menerapkan *vertical\_flip* atau *horizontal\_flip* karena dapat menyebabkan salah mengklasifikasikan.

#### c. Training model

*Convolutional Neural Network (CNN)* adalah model yang dipergunakan pada proses klasifikasi. Model ini dijalankan menggunakan manipulasi matriks matematika khusus untuk memproses data dari gambar [13]. Desain model yang diusulkan terdiri dari arsitektur 13 lapis. Untuk semua lapisan konvolusi, ukuran kernel ditetapkan menjadi 3×3. Jumlah *kernel (filter)* di lapisan konvolusi diatur ke 25, 50, 70, dan 512. Kami menggunakan *glorot\_uniform* [14] untuk menginisialisasi bobot kernel dan *bias* diinisialisasi dengan nol. *Dropout* diatur ke 20%. Kami menggunakan fungsi optimasi Adam dengan initial *learning rate*  $\alpha = 0.01$ , dan  $\beta_1 = 0.9$ , dan  $\beta_2 = 0.999$ . Pool pooling maksimum diatur ke 2×2, padding disetel sebagai *valid*, dan *strides* disetel ke 1×1. Lapisan *Flatten* digunakan untuk menyesuaikan ukuran data masukan sebelumnya lapisan padat yang terhubung sepenuhnya. Unit lapisan *dense* diatur ke 512. Bobot lapisan padat diinisialisasi menggunakan fungsi *glorot\_uniform*, dan fungsi *ReLU* tadi digunakan sebagai fungsi aktivasi karena dapat minimalisir fungsi aktivasi dibandingkan dengan fungsi aktivasi non-linear lainnya [15]. Akhirnya, lapisan *softmax* digunakan untuk mengklasifikasikan 24 huruf dari ASL. Gambar 1 adalah struktur arsitektur dari model yang digunakan.



Gambar 1. Arsitektur Model CNN

#### d. Evaluasi model

Nilai prediksi atau evaluasi diperoleh berasal dari proses *training* model yang telah dibuat sebelumnya. Evaluasi model menggunakan data latih 27.455 kasus dalam masing masing kategori kelas pada bahasa isyarat amerika (ASL). Yang dicari pada proses ini adalah nilai *epoch output* yang telah dilatih, nantinya penulis akan menjalankan 20 kali *epoch* dengan menggunakan *learning rate reduction* yaitu contohnya parameter dalam *training* yang dipergunakan untuk mengukur nilai koreksi bobot pada saat proses *training* model. Nilai *learning rate* terletak pada jangkauan nol (0) hingga (1). Jika bertambah tinggi nilai *learning rate*, jadi proses *training* akan berjalan lebih cepat. Untuk melakukan perhitungan akurasi, penulis akan menggunakan *confusion matrix* agar mendapatkan nilai *true positif*, *true negatif*, *false positif*, dan *false negatif* untuk mengukur keakurasian dengan performa yang didapatkan dari model ini [16]. Selain itu penggunaan *confusion matrix* bertujuan untuk mendapatkan nilai akurasi dari data yang tidak seimbang.

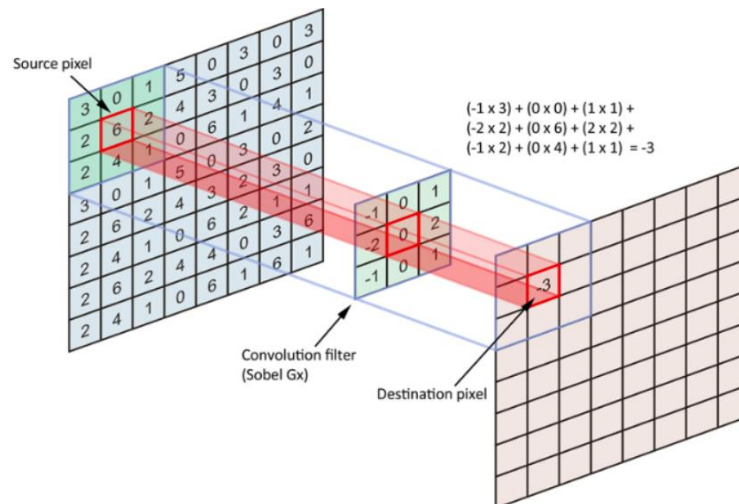
## 2.2 Machine Learning

*Machine Learning* yaitu teknik yang berasal dari kecerdasan buatan yang dikembangkan dengan mengumpulkan data dengan pendekatan matematika. Metode pembelajaran mesin meniru cara manusia belajar untuk pengenalan atau klasifikasi benda. Inti dari pembelajaran mesin adalah proses yang bertujuan untuk menciptakan model (matematika) yang menggambarkan pola-pola suatu objek yang dijadikan acuan [17]. Aplikasi pembelajaran mesin membutuhkan arsitektur untuk mempelajari karakteristik suatu objek. Pada penelitian ini arsitektur yang digunakan untuk machine learning adalah *Convolutional Neural Network (CNN)*.

*Convolutional Neural Network (CNN)* yaitu arsitektur *machine learning* yang berfungsi untuk mengetahui informasi yang dimaksudkan memprediksi bentuk suatu objek. Kemampuan *CNN* untuk mengenali objek berbeda dengan posisi *input* data [18]. Kemampuan ini membuat *Convolutional Neural Network (CNN)* saat ini banyak digunakan di berbagai bidang. *CNN* memiliki berbagai komponen yaitu *Convolutional layer*, *pooling layer*, dan *fully connected layer*.

### 2.3. Convolutional Neural Network

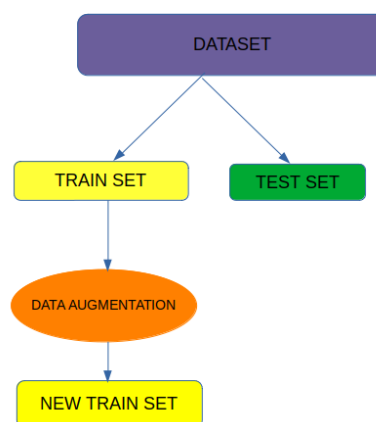
*Convolutional Neural Network* adalah tipe khusus dari implementasi kecerdasan Buatan yang menggunakan manipulasi matriks matematika khusus yang disebut operasi konvolusi untuk memproses data dari gambar. Model yang digunakan didasarkan pada konvolusi arsitektur jaringan saraf (*CNN*) sebagai algoritma yang kuat untuk tugas klasifikasi gambar [19]. Ringkasan model yang dipergunakan ditunjukkan pada gambar 2.



Gambar 2. Ringkasan model yang digunakan

### 2.4 Augmentasi data

Augmentasi data adalah proses yang bertujuan untuk meningkatkan jumlah data yang ada dengan melakukan beberapa modifikasi yang berbeda pada salinan data asli. Augmentasi data telah digunakan dalam beberapa analisis data dan metode *machine learning* di mana ada kekurangan ketersediaan data pelatihan untuk melatih modelnya. Untuk menghindari masalah *overfitting*, kita perlu memperluas dataset kita secara artifisial [20]. Proses ini dapat membuat kumpulan data yang ada menjadi lebih besar. Idennya adalah mengubah data pelatihan dengan transformasi kecil untuk memproduksi variasi. Pendekatan yang mengubah data pelatihan dengan cara yang mengubah representasi *array* sambil mempertahankan label yang sama dikenal sebagai teknik augmentasi data. Beberapa augmentasi populer yang digunakan orang adalah *grayscale*, *horizontal flips*, *vertical flips*, *random crops*, *color jitters*, *translations*, *rotations*, dan banyak lagi. Proses augmentasi dapat dilihat pada Gambar 3.



Gambar 3. Proses augmentasi data

## 3. HASIL DAN PEMBAHASAN

Hasil dari model yang dikembangkan yang terdiri dari dua tahap yaitu yang pertama tahapan pelatihan kemudian tahap pengujian. Tahap pelatihan terdiri dari beberapa proses. Prosesnya adalah proses memasukkan *dataset*, *preprocessing*,

training dengan mobilenet2, save model sedangkan tahap pengujian adalah proses memasukkan citra, preprocessing, pengenalan objek, dan hasil. Berikut merupakan pembahasan tentang bagaimana proses pembangunan model dari penggunaan dataset hingga melakukan proses evaluasi terhadap model yang sudah dibangun.

### 3.1 Pembahasan dan Hasil

#### 3.1.1 Load ASL dataset

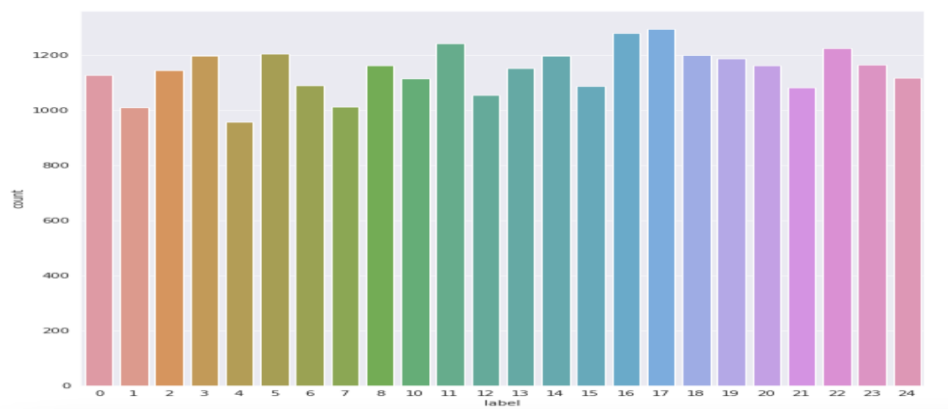
Hal pertama yang harus dilakukan adalah melakukan load pada dataset csv yang telah dipersiapkan. Data pada penelitian ini berisi 5 baris dan 785 kolom. Dataset gambar MNIST asli dari angka tulisan tangan adalah tolok ukur populer untuk metode machine learning berbasis gambar, tetapi para peneliti telah memperbaruinya sebagai upaya untuk memperbaikinya dan mengembangkan serta sebagai pengganti drop-in yang lebih menantang untuk computer vision dan secara realtime untuk aplikasi dunia nyata. Sebagaimana tercatat dalam sebuah dataset ini yang disebut dataset Fashion-MNIST, para peneliti Zalando melakukan klaim mengejutkan bahwa "Sebagian besar pasangan digit MNIST (784 total piksel per sampel) dapat dibedakan dengan cukup baik hanya dengan satu piksel". Untuk mengembangkan komunitas agar dapat mengembangkan lebih banyak pengganti drop-in, MNIST Bahasa Isyarat disajikan di sini dan mengikuti format CSV yang sama dengan label dan nilai piksel dalam satu baris. Basis data huruf Bahasa Isyarat Amerika dari gerakan tangan mewakili sebuah multi-kelas dengan 24 kelas huruf dalam dataset ini dimana tidak termasuk huruf J dan Z karena membutuhkan gerakan. Format kumpulan data dipolakan agar sesuai dengan MNIST klasik. Setiap kasus pelatihan dan pengujian mewakili label (0-25) sebagai peta satu-ke-satu untuk setiap huruf alfabet A-Z (dan tidak ada kasus untuk 9=J atau 25=Z karena gerakan isyarat). Gambar 4 yaitu image yang ada pada dataset.



Gambar 4. Image yang terdapat pada dataset

#### 3.1.2 Visualisasi data dan preprocessing data

Preprocessing data merupakan tahap awal untuk melakukan latihan data. Datasets akan di muat dan dilakukan scaling ulang melalui kode program dengan tujuan menyesuaikan nilai masukan pada model yang dibuat dan menyelaraskan semua ukuran citra. Selain itu citra juga digambarkan dalam bentuk matriks RGB dengan range nilai 0 sampai 1200 yang kemudian di ubah ke dalam range 0 sampai 1 seperti pada gambar 5.



Gambar 5. Sebaran data 0-24 dalam RGB

### 3.1.3 Hasil Training model

Pada bagian ini dilakukan untuk mempelajari fitur-fitur dan kelas dari citra. Tujuan dari latihan ini yaitu agar model mampu mengenali ciri untuk melakukan klasifikasi huruf-huruf alfabet. Selanjutnya pada tahap ini dilakukan pelatihan dengan *dataset* yang telah dimuat dan dipisahkan terhadap model yang telah dibuat. Proses pertama adalah menyiapkan *sequential model* yang dapat diperhatikan pada tabel 1.

**Tabel 1.** *Sequential Model*

Layer (type)	Output Shape	Param
Conv2d 1 (Conv2D)	(None, 28, 28, 75)	750
Batch normalization 1	(Batch (None, 28, 28, 75)	300
Max pooling2d_1	(MaxPooling2 (None, 14, 14, 75)	0
conv2d_2 (Conv2D)	(None, 14, 14, 50)	33800
Dropout 1 (Dropout)	(None, 14, 14, 50)	0
Batch normalization 2	(Batch (None, 14, 14, 50)	200
Max pooling2d 2	(MaxPooling2 (None, 7, 7, 50)	0
Conv2d 3 (Conv2D)	(None, 7, 7, 25)	11275
Batch normalization 3	(Batch (None, 7, 7, 25)	100
Max pooling2d 3	(MaxPooling2 (None, 4, 4, 25)	0
Flatten 1 (Flatten)	(None, 400)	0
Dense 1 (Dense)	(None, 512)	205312
Dropout 2 (Dropout)	(None, 512)	0
Dense 2 (Dense)	(None, 24)	12312
Total params: 264,049		
Trainable params: 263,749		
Non-trainable params: 300		

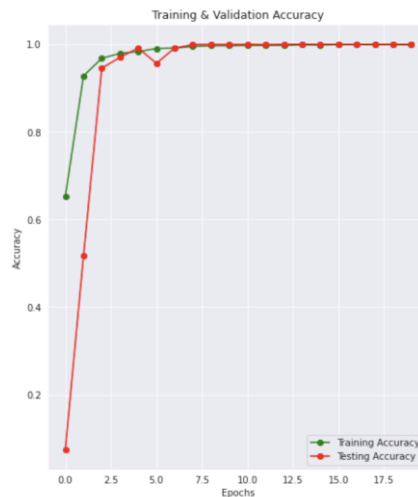
Berdasarkan data proses pelatihan terakhir dengan *epoch* sebanyak 20 proses pelatihan tersebut menghasilkan nilai *loss* sebesar 0.0047, nilai keakurasian sebesar 0.9985, nilai *Val Loss* sebesar 6.6754e dan Nilai *Val accuracy* sebesar 1.000. hasil tersebut merupakan hasil terhadap *dataset* yang telah di proses. Adapun hasil dari setiap percobaan yang dilakukan dapat diperhatikan pada tabel 2 dibawah.

**Tabel 2.** Hasil *training epoch*

Epoch	Loss	Accuracy	Val Loss	Val Accuracy
1	1.1045	0.6530	3.7369	0.0752
2	0.2220	0.9278	1.5874	0.5170
3	0.1011	0.9678	0.1524	0.9452
4	0.0647	0.9790	0.0890	0.9699
5	0.0529	0.9831	0.0279	0.9912
6	0.0329	0.9901	0.1260	0.9564
7	0.0283	0.9911	0.0286	0.9911
8	0.0175	0.9949	0.0058	0.9992
9	0.0118	0.9965	0.0025	1.0000
10	0.0116	0.9965	0.0039	0.9994
11	0.0100	0.9970	0.0018	1.0000
12	0.0095	0.9971	0.0040	0.9990
13	0.0079	0.9974	0.0016	1.0000
14	0.0056	0.9986	0.0011	1.0000
15	0.0056	0.9983	7.5067e	1.0000
16	0.0044	0.9992	7.9233e	1.0000
17	0.0041	0.9991	4.5513e	1.0000
18	0.0042	0.9991	4.8956e	1.0000
19	0.0040	0.9991	6.5453e	1.0000
20	0.0047	0.9985	6.6754e	1.0000

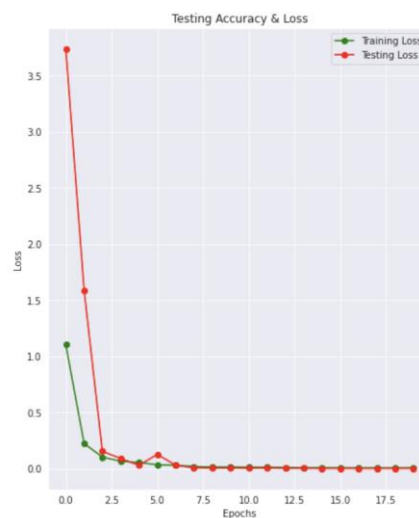
### 3.1.4 Hasil Evaluasi model

Pembuatan grafik hasil latih data ini akan membuat dan menyimpan gambar grafik *training & validasi accuracy* dan juga grafik *accuracy* dan *loss* dari hasil latih data yang telah dilakukan. Grafik yang dibuat berdasarkan dari hasil *training* yang telah dilakukan dengan data latih sebelumnya. Akurasi dan *loss* mengalami kenaikan dan penurunan pada setiap *epochs*. Untuk lebih lengkapnya dapat diperhatikan pada gambar 6 dan 7 dibawah.



**Gambar 6.** Visualisasi *training* dan *validation accuracy*

Pada gambar 6 Visualisasi *training* dan validasi akurasi dapat dilihat bahwa mulai dari *epoch* 2 memiliki kenaikan pada nilai akurasi, yaitu hampir mendekati nilai 1. Pada *epoch* ke 5 mengalami sedikit penurunan dan kembali stabil pada *epoch* ke 7 dan seterusnya.



**Gambar 7.** Visualisasi *testing accuracy* dan *loss*

Bisa dilihat pada gambar 7 yaitu Visualisasi tes akurasi dan *loss*, pada *epoch* 2 mengalami penurunan tajam dibawah angka 0.5 dan sama seperti pada gambar 3, *epoch* mengalami kenaikan pada proses ke 5 dan kembali stabil pada *epoch* ke 7 dan seterusnya. Untuk hasil *training* per kelas dapat dilihat bahwa nilai rata - rata *recall*, *precision*, dan *f1-score* dari semua kelas adalah 1, hasil ini sempurna dari semua sisi. Hasil *training* ini dapat diperhatikan pada tabel 3 dibawah ini.

**Tabel 3.** Hasil *training F1-Score*

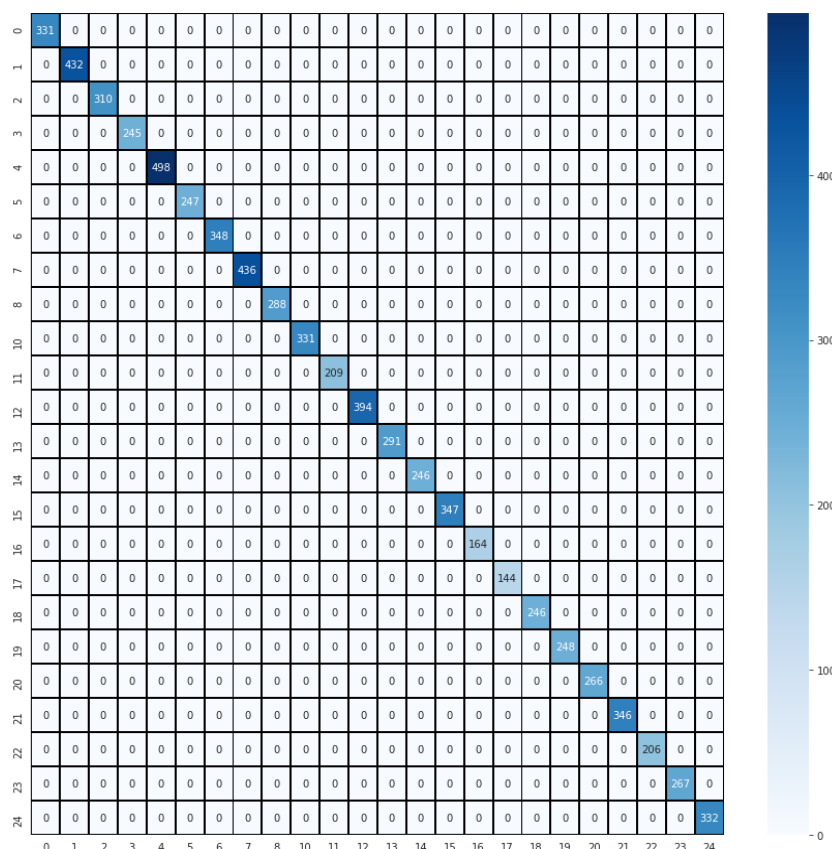
	Precision	Recall	F1-score	Support
Class 0	01.00	01.00	01.00	331
Class 1	01.00	01.00	01.00	432
Class 2	01.00	01.00	01.00	310
Class 3	01.00	01.00	01.00	245
Class 4	01.00	01.00	01.00	498
Class 5	01.00	01.00	01.00	247
Class 6	01.00	01.00	01.00	348
Class 7	01.00	01.00	01.00	436
Class 8	01.00	01.00	01.00	288
Class 10	01.00	01.00	01.00	331
Class 11	01.00	01.00	01.00	209
Class 12	01.00	01.00	01.00	394



Class 13	01.00	01.00	01.00	291
Class 14	01.00	01.00	01.00	246
Class 15	01.00	01.00	01.00	347
Class 16	01.00	01.00	01.00	164
Class 17	01.00	01.00	01.00	144
Class 18	01.00	01.00	01.00	246
Class 19	01.00	01.00	01.00	248
Class 20	01.00	01.00	01.00	266
Class 21	01.00	01.00	01.00	346
Class 22	01.00	01.00	01.00	206
Class 23	01.00	01.00	01.00	267
Class 24	01.00	01.00	01.00	332
Accuracy			01.00	7172
Macro AVG	01.00	01.00	01.00	7172
Weighted AVG	01.00	01.00	01.00	7172

3.1.5 Confusion Matrix

Dari hasil *training* tersebut dapat diperoleh nilai akurasi berdasarkan hasil *training* pada data validasi. Didapatkan nilai akurasi dari model dengan menghitung sejumlah data validasi yang telah terklasifikasikan dengan benar kemudian dibagi dengan jumlah keseluruhan dari data validasi. Nilai tersebut digunakan untuk mengetahui tingkat kesuksesan dari klasifikasi terkait model yang didapat dari hasil pelatihan. Evaluasi ini dilakukan terhadap kinerja model yang didapatkan berdasarkan jumlah data validasi yang diklasifikasikan secara tepat dan tidak tepat oleh model. Dalam melakukan analisa akurasi model yang diperoleh, akurasi bisa divisualisasikan menggunakan metode *confusion matrix* [21]. Tabel *confusion matrix* ini terdiri dari baris dan kolom dari jumlah kelas yang memberikan nilai *false positives*, *false negatives*, *true positives*, dan *true negatives*. *Confusion Matrix* akan dipergunakan ketika nilai akurasi yang didapat dari model tak sesuai karena adanya data yang kurang tepat dalam proses pengukuran akurasinya. Visualisasi menggunakan *confusion matrix* dapat diperhatikan pada gambar 8 dibawah.



Gambar 8. Visualisasi *confusion matrix*

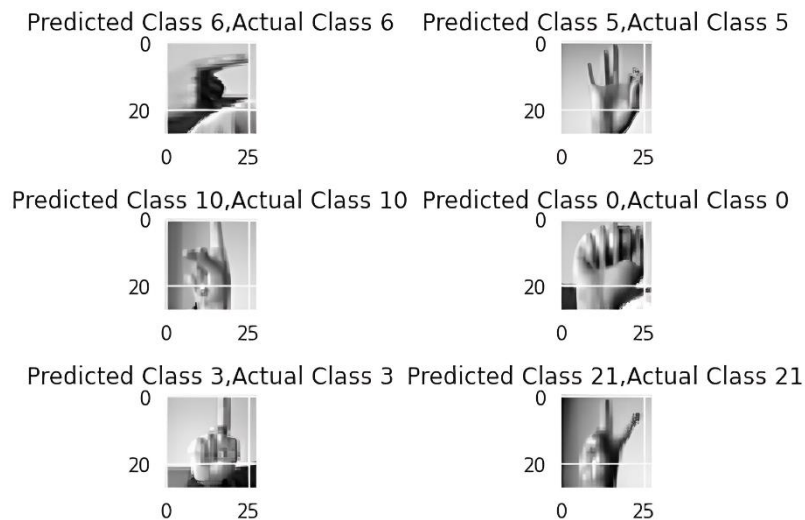
3.1.6 Percobaan Kelas yang Diprediksi dengan Benar

Setelah data di training dilakukan percobaan langsung pada gambar untuk menguji hasil yang didapatkan berdasarkan kelas, diantaranya adalah:



- a. Kelas 6 mewakili huruf G
- b. Kelas 5 mewakili huruf F
- c. Kelas 10 mewakili huruf L
- d. Kelas 0 mewakili huruf A
- e. Kelas 3 mewakili huruf D
- f. Kelas 21 mewakili huruf V

Untuk lebih jelasnya akan disajikan pada gambar 9 hasil percobaan yang telah diprediksi dengan benar berdasarkan kelas.



**Gambar 9.** Kelas yang diprediksi dengan benar

#### 4. KESIMPULAN

Penelitian ini telah mengembangkan suatu sistem untuk pengenalan atau klasifikasi *american sign language*. Sistem pengenalan yang dikembangkan adalah sistem yang menggunakan *convolutional neural network*, proses ini juga harus didukung oleh *preprocessing* seperti augmentasi data untuk meningkatkan hasil klasifikasi atau akurasi klasifikasi. Alasan inilah yang membuat penelitian ini terfokus pada mengembangkan teknik *preprocessing* augmentasi data yang dapat meningkatkan hasil akurasi. Penelitian ini dilakukan untuk tujuan bisa membaca bahasa isyarat tangan oleh para penyandang tuna rungu atau tuna wicara. Bahasa ini didapatkan melalui gerakan tangan yang akan mewakili masalah banyak-kelas dengan 24 kelas huruf. Data pelatihan yang penulis gunakan berjumlah 27.455 kasus dan untuk data uji berjumlah 7172 kasus. Model yang digunakan adalah dengan menggunakan metode *convolutional neural network (CNN)*. Implementasi metode *convolutional neural network* dengan percobaan 20 kali dengan *epochs* sebanyak 1 hingga 20 dan menghasilkan nilai *loss* sebesar 0.0047 dengan nilai keakurasian sebesar 0.9985. Untuk nilai rata - rata *recall*, *precision*, dan *f1-score* dari semua kelas adalah 1. Untuk percobaan langsung yang dilakukan berdasarkan kelas, terdapat 6 kelas yang pasti benar yaitu pada kelas 6, 10, 3, 5, 0 dan kelas 21. Dengan menggunakan metode *confusion matrix* dapat menghitung tiga jenis pengukuran yaitu pengukuran *accuracy*, *precision*, dan *recall* dengan menghasilkan nilai rata-rata sebesar 99.8%. Dengan ini dapat ditarik kesimpulan penerapan model *convolutional neural network* dapat digunakan bagi para masyarakat untuk menterjemahkan bahasa isyarat dalam bentuk huruf-huruf alfabet.

#### REFERENCES

- [1] K. K. Republik Indonesia, "Situasi Penyandang Disabilitas," *Pusdatin*, 2014. [Online]. Available: <https://pusdatin.kemkes.go.id/download.php?file=download/pusdatin/buletin/buletin-disabilitas.pdf>. [Accessed: 21-Jun-2022].
- [2] Republika, "Sumbar Hanya Punya 8 Penerjemah Bahasa Isyarat," *Republika Online*, 2019. [Online]. Available: <https://www.republika.co.id/berita/pzjwqr335/sumbar-hanyapunya-8-penerjemah-bahasa-isyarat>. [Accessed: 26-Jun-2022].
- [3] K. S. R. Indonesia, "Sistem Informasi Management Penyandang Disabilitas," *Kementerian Pendidikan dan Budaya Republik Indonesia*, 2021. [Online]. Available: <https://simpd.kemensos.go.id>. [Accessed: 10-May-2022].
- [4] H. Kulhandjian, P. Sharma, M. Kulhandjian, and C. D'Amours, "Sign language gesture recognition using doppler radar and deep learning," *2019 IEEE Globecom Work. GC Wkshps 2019 - Proc.*, 2019.
- [5] M. A. Jalal, R. Chen, R. K. Moore, and L. Mihaylova, "American Sign Language Posture Understanding with Deep Neural Networks," *2018 21st Int. Conf. Inf. Fusion, FUSION 2018*, pp. 573–579, 2018.
- [6] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," *2018 Conf. Signal Process. Commun. Eng. Syst. SPACES 2018*, vol. 2018-Janua, pp. 194–197, 2018.
- [7] A. Dasl, S. Gawde, K. Suratwala, and D. Kalbande, "Sign Language Recognition Using Deep Learning on Custom Processed



- Static Gesture Images,” *2018 Int. Conf. Smart City Emerg. Technol. ICSCET 2018*, 2018.
- [8] S. Ameen and S. Vadera, “A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images,” *Expert Syst.*, vol. 34, no. 3, 2017.
- [9] L. K. S. Tolentino, R. O. Serfa Juan, A. C. Thio-ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, “Static sign language recognition using deep learning,” *Int. J. Mach. Learn. Comput.*, vol. 9, no. 6, pp. 821–827, 2019.
- [10] R. Daroya, D. Peralta, and P. Naval, “Alphabet Sign Language Image Classification Using Deep Learning,” *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2018-Octob, no. October, pp. 646–650, 2019.
- [11] K. Lata, M. Dave, and N. K.N., “Data Augmentation Using Generative Adversarial Network,” *SSRN Electron. J.*, pp. 1–14, 2019.
- [12] N. Elsayed, A. S. Maida, and M. Bayoumi, “Empirical activation function effects on unsupervised convolutional LSTM learning,” *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 2018-Novem, pp. 336–343, 2018.
- [13] F. Sultana, A. Sufian, and P. Dutta, “Advancements in image classification using convolutional neural network,” *Proc. - 2018 4th IEEE Int. Conf. Res. Comput. Intell. Commun. Networks, ICRCICN 2018*, pp. 122–129, 2018.
- [14] X. Luo, X. Qin, Z. Wu, F. Yang, M. Wang, and J. Shang, “Sediment Classification of Small-Size Seabed Acoustic Images Using Convolutional Neural Networks,” *IEEE Access*, vol. 7, pp. 98331–98339, 2019.
- [15] S. Z. M. Zaki, M. A. Zulkifley, M. Mohd Stofa, N. A. M. Kamari, and N. A. Mohamed, “Classification of tomato leaf diseases using mobilenet v2,” *IAES Int. J. Artif. Intell.*, vol. 9, no. 2, pp. 290–296, 2020.
- [16] I. K. G. Darma Putra, I. P. D. Jayantha Putra, R. Fauzi, and D. Witarsyah, “Classification of Tomato Plants Diseases Using Convolutional Neural Network,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 10, no. 5, pp. 1821–1827, 2020.
- [17] A. Roihan, P. A. Sunarya, and A. S. Rafika, “Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper,” *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 5, no. 1, pp. 75–82, 2020.
- [18] A. Kurniadi, “Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras,” *DoubleClick J. Comput. Inf. Technol.*, vol. 4, no. 1, p. 25, 2020.
- [19] S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” *J. Big Data*, vol. 6, no. 1, 2019.
- [20] B. Hanin and D. Rolnick, “How to start training: The effect of initialization and architecture,” *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, pp. 571–581, 2018.
- [21] D. Normawati and S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 5, no. 2, pp. 697–711, 2021.