

## Klasifikasi Gerak Terapi untuk Anak Autis dengan Metode CNN pada Robot SAR (*Socially Assistive Robot*)

Ekawati Prihatini<sup>1\*</sup>, Nyayu Latifah Husni<sup>2</sup>, Revaldi Andrean<sup>3</sup>

Program Studi Sarjana Terapan Teknik Elektro, Politeknik Negeri Sriwijaya, Indonesia

\*e-mail: [ekawati\\_p@polsri.ac.id](mailto:ekawati_p@polsri.ac.id)

### ABSTRAK

Anak-anak penyandang autisme seringkali mengalami kesulitan dalam berinteraksi sosial dan berkomunikasi. *Socially Assistive Robot* (SAR) dapat dimanfaatkan untuk membantu mereka mengembangkan keterampilan sosial dan komunikasi. Salah satu aspek penting dalam interaksi dengan robot adalah kemampuan robot untuk mendeteksi gerak anak autis. Penelitian ini bertujuan untuk mengimplementasikan metode *Convolutional Neural Network* (CNN) pada robot SAR untuk mendeteksi gerak anak autis. Metode CNN dipilih karena kemampuannya dalam mengenali pola gambar dan mendeteksi objek dengan akurasi yang tinggi. Dataset gambar yang digunakan dalam penelitian ini terdiri dari gambar yang diambil menggunakan *camera handphone* secara manual dengan berbagai pose dari berbagai sudut pandang. Hasil penelitian menunjukkan bahwa metode CNN mampu mengenali gerakan anak autis dengan presisi tinggi, sehingga mendukung interaksi yang lebih efektif antara robot dan anak. Tantangan seperti kebutuhan dataset yang besar dan waktu pelatihan yang signifikan berhasil diatasi melalui optimalisasi proses *training*. Pengoptimalan ini menghasilkan efisiensi model yang lebih baik, mengurangi waktu komputasi sekaligus mempertahankan akurasi yang tinggi.

**Kata Kunci:** Anak Autis, Interaksi Robot - Manusia, *Convolutional Neural Network* (CNN), Robot SAR.

### ABSTRACT

*Children with autism often have difficulties in social interaction and communication. Socially Assistive Robots (SAR) can be utilized to help them develop social and communication skills. One of the important aspects in interaction with robots is the ability of the robot to detect the motion of autistic children. This research aims to implement the Convolutional Neural Network (CNN) method on the SAR robot to detect the motion of autistic children. The CNN method was chosen because of its ability to recognize image patterns and detect objects with high accuracy. The image dataset used in this study consists of images taken using a cellphone camera manually with various poses from various points of view. The results show that the CNN method is able to recognize the movements of autistic children with high precision, thus supporting more effective interaction between the robot and the child. Challenges such as large dataset requirements and significant training time were overcome through optimization of the training process. This optimization led to improved model efficiency, reducing computation time while maintaining high accuracy.*

**Keywords:** Autistic Child, Robot - Human Interaction, *Convolutional Neural Network* (CNN), SAR Robot.

### PENDAHULUAN

Autisme menghambat perkembangan anak-anak dalam berkomunikasi dan berinteraksi sosial. [1]. Gejala autis dapat bervariasi secara signifikan antara individu dengan yang lain. Anak-anak penderita autis cenderung mengulangi gerakan tertentu yang disebut *stimming*. *Stimming* (*self-stimulatory behavior*) adalah istilah yang digunakan untuk menggambarkan gerakan berulang atau aktivitas yang dilakukan berulang kali oleh anak dengan *Autisme Spectrum Disorder* (ASD) [2-7]. Untuk itu, terapi gerak menjadi salah satu pengobatan utama yang membantu anak autisme dalam melatih kekuatan otot, keterampilan motorik, berkomunikasi, dan berinteraksi sosial. [8-9].

Penggunaan teknologi dalam pengobatan autisme yaitu terapi gerak telah menunjukkan potensi yang besar. Robot SAR (*Socially Assistive Robot*) dapat menjadi solusi yang efektif dalam terapi anak-anak autisme [10]. Robot ini mampu menyediakan interaksi yang konsisten, memberikan umpan balik secara tepat waktu, dan membuat sesi terapi lebih menyenangkan bagi anak-anak [11-12].

Robot SAR dapat dilengkapi dengan program *Mirroring Movement* yang dirancang untuk membantu anak-anak autisme belajar mengendalikan motorik dasar mereka [13]. Salah satu terapi yang dapat diterapkan pada robot SAR adalah terapi menirukan gerakan. Terapi ini melibatkan lengan robot yang dirancang untuk bergerak dan membantu anak-anak autisme meningkatkan serta mengendalikan gerakan dasar mereka [14-16]. Robot dapat mengarahkan anak-anak autisme untuk fokus pada lengan robot dan memulai meniru gerakan. Ketika anak-anak autisme berhasil meniru gerakan yang sama dengan robot selanjutnya robot akan melakukan gerakan tahap berikutnya [17].

Metode pengenalan dan klasifikasi gerak sangat penting untuk terapi menggunakan robot. Salah satu metode *Deep Learning* yang paling sering digunakan yaitu metode *Convolutional Neural Network* (CNN), metode *Convolutional Neural Network* (CNN) adalah algoritma pembelajaran mendalam yang dirancang khusus untuk tugas yang membutuhkan pengenalan objek seperti klasifikasi, deteksi, dan segmentasi gambar [18-20]. CNN juga telah terbukti efektif dalam berbagai aplikasi pengenalan pola [21]. Dengan menggunakan CNN gerakan anak-anak dapat diidentifikasi dan diklasifikasikan secara akurat, serta memungkinkan robot merespon gerakan anak apakah sudah benar atau belum dalam mengikuti gerakan yang diperintahkan robot [22-23]. Pra-pemrosesan terapi gerak yang dilakukan, yaitu mengekstraksi fitur vektor gambar menjadi 128 dimensi dan mengklasifikasikan vector gerak tersebut, untuk meningkatkan akurasi sistem pengenalan gerak menggunakan metode CNN. Hasil penelitian ini mendapatkan nilai presisi sebesar 98% akurasi *training* dan 82% validasi *accuracy*.

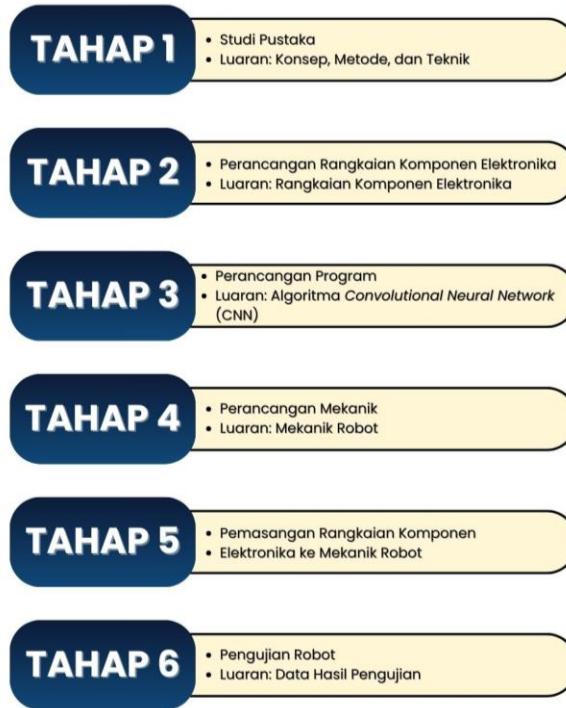
## METODE PENELITIAN

Merencanakan pembuatan *Socially Assistive Robot* memerlukan penataan tahapan yang terstruktur dan terperinci agar robot dapat dikembangkan dengan baik. Adapun metode penelitian menjalankan serangkaian langkah-langkah seperti pada gambar 1.

Gambar 1 menunjukkan proses perancangan sebuah sistem *Socially Assistive Robot* (SAR) yang dimulai dengan langkah-langkah berikut:

1. Studi literatur: Mencari dan menganalisis literatur terkait untuk mengumpulkan informasi dari kajian ilmiah yang relevan tentang intervensi SAR untuk anak-anak dengan ASD.
2. Observasi peralatan: Melakukan observasi menyeluruh terhadap peralatan yang sudah ada, baik itu robotika maupun teknologi terkait, untuk memahami kemungkinan integrasi dan peningkatan yang diperlukan.
3. Pengembangan desain mekanik: Mengembangkan desain mekanik dengan membuat kerangka atau struktur untuk menempatkan sensor-sensor yang diperlukan serta bagian-bagian lain dari perangkat keras yang dibutuhkan.
4. Desain dan simulasi: Proses desain dan simulasi bertujuan untuk mengatur koneksi kabel data dan power supply pada modul-modul yang digunakan, memastikan kecocokan dan keandalan sistem secara keseluruhan.
5. Pemrograman: Tahap ini melibatkan pengembangan perangkat lunak dan pemrograman sesuai dengan kebutuhan sistem, termasuk algoritma untuk interaksi sosial dan fungsionalitas lainnya.
6. Pengujian: Pengujian dilakukan melalui dua metode, yaitu simulasi test dan implementasi test. Simulasi test digunakan untuk memvalidasi fungsi-fungsi sistem secara virtual, sementara

implementasi test dilakukan untuk memverifikasi kinerja sistem dalam situasi nyata atau di lingkungan yang sesungguhnya.



Gambar 1. Kerangka Penelitian

### Data Sheet

Untuk melaksanakan penelitian ini, langkah pertama yang harus dilakukan adalah menyiapkan kumpulan data yang terdiri dari set gambar masing masing gerakan sebanyak 7 gerakan. Dataset ini akan digunakan sebagai input yang akan diproses dalam sistem. Setelah data dikumpulkan selanjutnya data akan dibagi menjadi data *training* dan juga data *testing* dengan perbandingan 80% data *training*, 20 % data *testing*.

Tabel 1. Jumlah dataset

Klasifikasi	Jumlah Data	Training	Persentase	Testing	Persentase
Gerakan 1	200	160	80%	40	20%
Gerakan 2	200	160	80%	40	20%
Gerakan 3	200	160	80%	40	20%
Gerakan 4	200	160	80%	40	20%
Gerakan 5	200	160	80%	40	20%
Gerakan 6	200	160	80%	40	20%
Gerakan 7	200	160	80%	40	20%

Tabel 1 menjelaskan bahwa pengujian pada penelitian ini menggunakan 7 kelas gerakan untuk memenuhi kebutuhan data klasifikasi. Pada tiap kelas gerakan diambil 200 data dan dibagi kembali menjadi data *training* dan data *testing* dengan perbandingan 80% dan 20 % dari jumlah data setiap kelas.

### Data Training

Data *training* digunakan untuk melatih atau membangun model, untuk menyesuaikan dengan data yang diberikan selama *training* algoritma *machine learning* [24]. Seperti otak manusia, dimana kepintaran manusia akan mengalami perubahan saat belajar. Tujuan dari *training* model adalah untuk menguji kemampuan model saat latihan dengan data *training* dan kemudian menguji kinerjanya dengan data validasi.

Dalam penelitian ini penulis akan membuat robot SAR yang dapat membedakan antara gerakan satu dengan gerakan lainnya. Selanjutnya penulis akan memasukan data gerakan 1 sampai dengan gerakan 7 ke dalam *machine learning*, yang kemudian akan mempelajari gambar tersebut hingga algoritma *machine learning* dapat membedakan antara gerakan 1 dengan gerakan lainnya.

### Data Testing

Setelah model dilatih dengan baik dan mampu mengenal pola dengan akurasi yang tinggi, selanjutnya mulai mempelajari data *testing* yaitu untuk menguji model setelah proses *training* selesai. Data *testing* merupakan data yang digunakan untuk menguji kinerja sebuah model *Machine Learning* yang telah dibuat [25]. Data ini bertujuan untuk menguji kemampuan model dalam mengolah data yang belum pernah ditemui sebelumnya selama *training*.

### PreProcessing

Data *Preprocessing* dalam konteks *Convolutional Neural Network* (CNN) adalah proses yang diambil untuk mempersiapkan data gambar agar dapat digunakan sebagai input untuk model CNN. Proses ini dilakukan dengan tujuan untuk memperbaiki kesalahan pada data mentah yang seringkali tidak lengkap dan memiliki format yang tidak teratur yang mengakibatkan *machine learning* sulit untuk membaca data yang ada.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale = 1.0/255,
    rotation_range = 30,
    width_shift_range = 0.2,
    height_shift_range = 0.2,
    shear_range = 0.2,
    zoom_range = 0.2,
    #horizontal_flip = True,
    #vertical_flip = True,
    fill_mode = 'nearest',
    validation_split = 0.2,
)

train_generator = train_datagen.flow_from_directory(
    base_dir,
    target_size = (128,128),
    batch_size = 32,
    #shuffle = True,
    subset = 'training',
    class_mode = 'categorical'
)
```

Gambar 2. *Preprocessing* data

Pada gambar 2 *Preprocessing* data dilakukan dengan mengubah ukuran gambar dari gerakan 1 sampai gerakan 7 menjadi ukuran yang seragam, meningkatkan kualitas data, dan mengurangi *noise*. Beberapa langkah umum dalam *preprocessing* untuk CNN meliputi *resizing*, yaitu mengubah ukuran gambar agar konsisten dengan input yang diharapkan oleh model, selanjutnya normalisasi, yang mengubah nilai piksel gambar sehingga berada dalam rentang tertentu, Biasanya, bilangan biner bernilai (0,1), sedangkan bilangan simetris berada dalam rentang (-1,1) dan augmentasi data yaitu teknik untuk memperbesar dataset dengan melakukan transformasi seperti rotasi, flip horizontal/vertikal, zoom, pergeseran, atau distorsi kecil lainnya untuk membantu model belajar menjadi lebih akurat terhadap variasi dalam data.

### Perancangan Model *Convolutional Neural Network* (CNN)

Perancangan model CNN adalah bagian dari tahapan proses dalam menyusun model yang digunakan untuk melatih data dalam mengenali objek yang diinginkan, model yang disusun dalam proses penelitian terdiri dari lapisan konvolusi MobileNetV2, lapisan *pooling*, lapisan *dropout*, lapisan *global average pooling* 2D, dan lapisan *dense*, contoh model yang dipakai pada penelitian ini bisa dilihat pada gambar 3.

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 4, 4, 1280)	2257984
dropout (Dropout)	(None, 4, 4, 1280)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout_1 (Dropout)	(None, 1280)	0
dense (Dense)	(None, 128)	163968
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 7)	903

-----  
 Total params: 2422855 (9.24 MB)  
 Trainable params: 2388743 (9.11 MB)  
 Non-trainable params: 34112 (133.25 KB)

Gambar 3. Model CNN

Gambar 3 menunjukkan perancangan model CNN yang telah selesai, Langkah berikutnya yaitu melakukan pelatihan model dengan data yang sudah dikumpulkan, untuk proses pengujian model, pada penelitian ini melakukan beberapa kali percobaan dengan *epoch* yang berbeda untuk menentukan pelatihan jaringan yang maksimal.

```

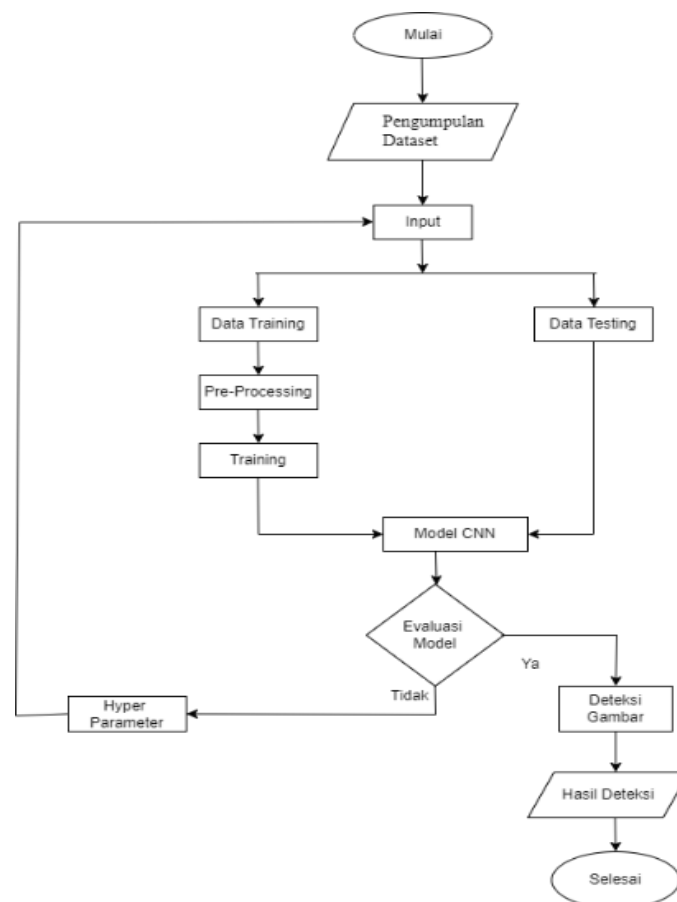
history = model.fit(
    train_generator,
    epochs=50,
    steps_per_epoch=len(train_generator),
    validation_data=validation_generator,
    validation_steps=len(validation_generator), # Jumlah langkah untuk data validasi
    verbose=2,
    #callbacks = ([early_stopping])
)

```

Gambar 4. *Training* parameter

Gambar 4 menunjukkan program yang digunakan untuk melatih sebuah *neural network* menggunakan *TensorFlow*. *TensorFlow* adalah *framework open-source* yang dikembangkan oleh Google Brain, yang digunakan untuk mengembangkan dan melatih berbagai model *machine learning*, terutama *neural networks* dan *deep learning*. Dalam program ini, *train\_generator* digunakan untuk menghasilkan data training, sementara *validation\_generator* berfungsi untuk menyediakan data validasi guna mengevaluasi performa model selama proses pelatihan, *epoch* = 50 untuk menentukan jumlah putaran seluruh dataset pelatihan yang akan dilakukan, jumlah *epoch* menyesuaikan akurasi yang dibutuhkan.

*Flowchart* berikut menggambarkan alur kerja dari proses pengembangan model CNN yang digunakan dalam penelitian ini.



Gambar 5. *Flowchart* Penelitian

Gambar 5 menunjukkan suatu *flowchart* dari penelitian yang akan dilakukan. Proses dimulai dengan langkah persiapan dan perencanaan awal untuk penelitian. Langkah pertama yaitu pengumpulan data yang diperlukan untuk melatih dan menguji model CNN. setelah dataset dikumpulkan, data selanjutnya akan dibagi menjadi dua bagian, yaitu 80% data *training* dan 20 % data *testing*, data *training* untuk melatih model, sementara data *testing* untuk menguji kinerja model. Setelah data *training* selanjutnya akan dilakukan *pre-processing* data untuk meningkatkan kualitas dan konsistensi. Langkah ini meliputi normalisasi, pengubahan ukuran gambar, Model

CNN dilatih menggunakan data *training* yang telah diproses, setelah pelatihan, model diuji dengan data *testing* untuk mengevaluasi kinerja..

Jika model menunjukkan hasil yang optimal, maka dilanjutkan ke deteksi gerakan, dimana model digunakan untuk mendeteksi gerakan secara real time. Jika hasil masih belum bagus, dilakkan penyesuaian *hyperparameter* dan proses pelatihan diulang hingga model mencapai hasil yang maksimal.

## HASIL DAN PEMBAHASAN

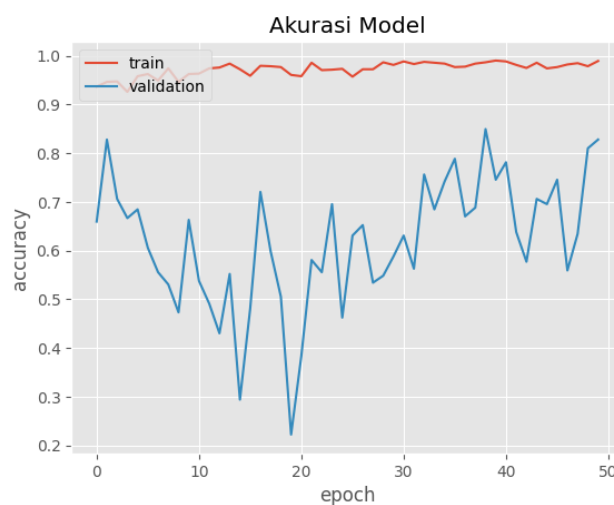
### Pengujian *Loss* dan Akurasi Data

Pada penelitian ini, model CNN dikembangkan untuk mengklasifikasikan gerakan yang diuji dengan menggunakan *loss* dan akurasi data.

Tabel 2. Perbandingan akurasi pada setiap *epoch*

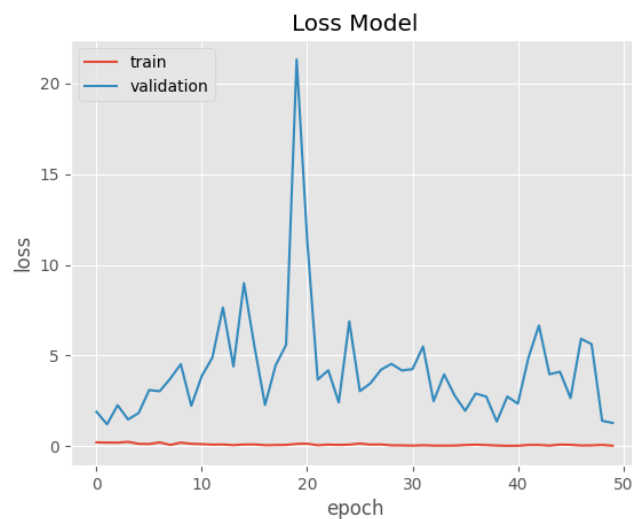
<i>Epoch</i>	<i>Loss</i>	Accuracy	Validasi <i>Loss</i>	Validasi Accuracy
5	0.1451	0.9589	5.9284	0.5125
15	0.0963	0.9786	4.4813	0.5269
25	0.1000	0.9732	6.8855	0.4624
50	0.0400	0.9893	1.2932	0.8280

Tabel 2 menunjukkan hasil proses selama *training*, nilai *loss* dihitung untuk mengetahui tingkat kesalahan pada prediksi model, nilai *loss* menurun seiring dengan bertambahnya jumlah *epoch*, yang menunjukkan bahwa model lebih baik dalam mempelajari data. Selain itu, akurasi data dihitung untuk mengetahui sejauh mana model dapat mengklasifikasikan gerakan dengan benar, hasilnya menunjukkan peningkatan yang signifikan dalam tingkat akurasi baik pada data *training* maupun data validasi, yang menunjukkan bahwa model mampu mengidentifikasi pola gerakan dengan baik. Dapat diketahui bahwa jumlah *epoch* sangat memengaruhi hasil nilai *loss* data, jika dilihat dari grafik tersebut dapat disimpulkan *epoch* 15 menghasilkan nilai *loss* tertinggi dan *epoch* 50 memberikan hasil nilai *loss* terkecil, nilai *loss* pada penelitian ini sangat menentukan hasil klasifikasi gerakan dan efektivitas model.



Gambar 6. Train dan validation akurasi model

Hasil akurasi menunjukkan persentase prediksi benar yang dilakukan model, serta memberikan gambaran langsung tentang efektivitas model dalam mengklasifikasikan data dengan benar. Dapat disimpulkan bahwa nilai *loss* yang rendah dan akurasi yang tinggi pada data, menunjukkan kemampuan model CNN untuk memprediksi dengan akurasi dan generalisasi yang baik, hal ini merupakan indikator utama keberhasilan model. Gambar 6 menunjukkan grafik akurasi *train* dan *validation* untuk 50 *epoch*, grafik akurasi model menunjukkan peningkatan akurasi dari data *train* seiring dengan bertambahnya jumlah *epoch*. Akurasi pada data validasi juga meningkat, meskipun terdapat beberapa penurunan seperti di *epoch* 20 yang menunjukkan ada beberapa gambar yang kurang jelas sehingga kemampuan generalisasi model terhadap data yang tidak terlihat selama pelatihan. Namun akurasi pada data validasi masih di kategori normal, yang menandakan bahwa model tidak mengalami *overfitting* secara signifikan.



Gambar 7. Train dan validation *loss* model

*Train* dan *validation loss* pada penelitian ini dapat dilihat pada Gambar 7, dimana permasalahannya hampir sama dengan akurasi model. Data validasi juga menunjukkan penurunan, meskipun pada *epoch* 20 naik dan *epoch* selanjutnya turun kembali.

```
Epoch 47/50
35/35 - 72s - loss: 0.0541 - accuracy: 0.9821 - val_loss: 5.9255 - val_accuracy: 0.5591 - 72s/epoch - 2s/step
Epoch 48/50
35/35 - 67s - loss: 0.0595 - accuracy: 0.9848 - val_loss: 5.6332 - val_accuracy: 0.6344 - 67s/epoch - 2s/step
Epoch 49/50
35/35 - 68s - loss: 0.0861 - accuracy: 0.9786 - val_loss: 1.4072 - val_accuracy: 0.8100 - 68s/epoch - 2s/step
Epoch 50/50
35/35 - 67s - loss: 0.0400 - accuracy: 0.9893 - val_loss: 1.2932 - val_accuracy: 0.8280 - 67s/epoch - 2s/step
```

Gambar 8. Proses *Training*

Hasil percobaan deteksi gerakan yang dilakukan pada penelitian ini menggunakan konfigurasi model MobileNetV2. Model ini mampu mendeteksi dan melacak objek yang bergerak dengan tingkat akurasi yang cukup tinggi, yaitu 98% untuk akurasi pelatihan dan 82% untuk akurasi validasi. dimana model MobileNetV2 mampu membedakan antara gerakan satu dengan

gerakan lainnya. Pengujian model dilakukan untuk menguji tingkat akurasi dari model CNN yang sudah dibuat. Akurasi ini menggambarkan sejauh mana model berhasil mengklasifikasikan jenis gerakan. Akurasi yang tinggi menandakan bahwa model mampu mengklasifikasikan ketujuh jenis gerakan dengan baik. Gambar 8 memperlihatkan hasil akhir akurasi dengan 50 *epoch*, dari gambar dapat dilihat bahwa model mencapai *train* akurasi 98% dan validasi akurasi 82% yang menunjukkan performa yang baik dalam mengklasifikasikan ketujuh jenis gerakan.

Hasil pengujian pada Gambar 9 menunjukkan hasil klasifikasi gerakan yang benar. data di atas merupakan hasil sampel yang diambil secara *real time* melalui aplikasi Pycharm dan kamera laptop. Hasil pengujian dimulai dari gerakan 1 sampai gerakan 4. Sedangkan untuk label perhitungan prediksi dimulai dari prediksi 0 yang menunjukkan gerakan 1 dan seterusnya hingga prediksi 3 yang menunjukkan gerakan 4.



Gambar 9. Contoh hasil klasifikasi gerakan yang benar

Hasil pengujian yang ditunjukkan pada Gambar 10 menunjukkan hasil klasifikasi gerakan yang salah. Ada beberapa faktor yang bisa menyebabkan hasil pengujian tidak terdeteksi atau salah, seperti gambar objek pada jarak yang cukup jauh, kondisi pencahayaan yang tidak konsisten atau terlalu gelap sehingga tidak terdeteksi dengan benar oleh model CNN, dan model CNN gagal mendeteksi objek karena latar belakang memiliki warna yang hampir sama dengan objek.



Gambar 10. Contoh hasil klasifikasi gerakan yang salah

## KESIMPULAN

Dalam penelitian ini, *convolutional neural network* (CNN) digunakan untuk klasifikasi gambar pada dataset gerakan. Model berhasil mencapai akurasi *train* 98% dan validasi akurasi 82% melalui langkah-langkah *preprocessing* seperti *resizing*, normalisasi, dan peningkatan data. Dengan menggunakan model MobileNetV2, menunjukkan performa yang sangat baik dan *robustness* terhadap variasi pada data gerakan, model ini mampu mengenali gerakan dari 1-7 dan menghasilkan prediksi yang cukup akurat. Hasil ini membuktikan bahwa CNN dapat mengekstraksi fitur penting secara otomatis tanpa menggunakan teknik manual yang rumit.

Namun, beberapa masalah muncul, seperti kebutuhan dataset yang besar dan banyak nya kelas pada penelitian ini menyebabkan jumlah waktu dan sumber daya komputasi yang diperlukan cukup lama pada *proses training*. Meskipun demikian dengan ketersediaan lebih banyak dataset masalah ini dapat diselesaikan.

Secara keseluruhan, penelitian ini menunjukkan bahwa CNN sangat efektif untuk penelitian klasifikasi gambar karena dapat memberikan hasil yang sangat akurat dengan menggunakan langkah-langkah *preprocessing* yang tepat dan arsitektur model yang efisien. Oleh karena itu, CNN sangat direkomendasikan untuk berbagai penelitian yang membutuhkan klasifikasi gambar.

## DAFTAR PUSTAKA

A. Widodo, F. M. Nugraha, and A. N. Hidayati, "Hubungan Tingkat Kemampuan Kognitif Dan Social Life Pada Anak Autism: Literature Review," *Jurnal Kesehatan Al-Irsyad*, vol. 15, no. 1, pp. 26-34, 2022.

- [1] Rahayu, S. M. (2014). Deteksi dan intervensi dini pada anak autisme. *Jurnal Pendidikan Anak*, 3(1).

- [2] D. Septia, L. Mauliani, and A. Anisa, "Pengaruh perilaku penyandang autisme terhadap desain ruang dalam studi kasus: Bangunan pendidikan," *PURWARUPA Jurnal Arsitektur*, vol. 1, no. 2, pp. 1-12, 2018.
- [3] A. P. Maharani and R. U. Panjaitan, "Resiliensi dan hubungannya dengan tingkat stres orang tua yang memiliki anak penyandang autism spectrum disorder," *\*Jurnal Ilmu Keperawatan Jiwa\**, vol. 2, no. 1, pp. 47-54, 2019.
- [4] M. H. M. Sharif and N. A. Jamil, "Cabaran ibu bapa bersama kanak-kanak kecelaruan spektrum autisme (ASD): tinjauan literatur," *Jurnal Sains Sosial*, vol. 4, no. 1, pp. 33-43, 2019.
- [5] J. Suteja and R. Wulandari, "Bentuk dan model terapi terhadap anak-anak penyandang autisme (keterbelakangan mental)," *Scientiae Educatia: Jurnal Pendidikan Sains*, vol. 2, no. 1, 2013.
- [6] J. Suteja, "Bentuk dan metode terapi terhadap anak autisme akibat bentukan perilaku sosial," *Eduksos: Jurnal Pendidikan Sosial & Ekonomi*, vol. 3, no. 1, 2014.
- [7] E. Martinez-Martin, F. Escalona, and M. Cazorla, "Socially assistive robots for older adults and people with autism: An overview," *Electronics*, vol. 9, no. 2, p. 367, 2020.
- [8] Z. Shi, T. R. Groechel, S. Jain, K. Chima, O. Rudovic, and M. J. Matarić, "Toward personalized affect-aware socially assistive robot tutors for long-term interventions with children with autism," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 11, no. 4, pp. 1-28, 2022.
- [9] C. Clabaugh, D. Becerra, E. Deng, G. Ragusa, and M. Matarić, "Month-long, in-home case study of a socially assistive robot for children with autism spectrum disorder," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, March 2018, pp. 87-88.
- [10] A. Nugroho, R. A. Yulianawan, and R. A. Firmansyah, "Rancang Bangun Robot Sar (Search And Rescue) Dengan Sistem Kendali Semi Otomatis," *SinarFe7*, vol. 2, no. 1, pp. 415-418, 2019.
- [11] A. Febtriko, W. Yulianti, and T. Rahayuningsih, "Pemanfaatan Ubtech Humanoid Robot sebagai Media Pembelajaran Kepada Guru Menuju Era Revolusi Industri 4.0 Dibidang Pendidikan untuk Anak Autis di Pekanbaru Lab School," *Jurnal Pengabdian Masyarakat Multidisiplin*, vol. 3, no. 1, pp. 42-52, 2019.
- [12] P. J. Chandra, "Deteksi Objek Manusia Dengan Metode Support Vector Machine," 2016.
- [13] J. A. Andre, "Sistem security webcam dengan menggunakan Microsoft Visual Basic (6.0)," *\*Rabit: Jurnal Teknologi dan Sistem Informasi Univrab\**, vol. 1, no. 2, pp. 48-60, 2016.
- [14] R. K. Ross, "Beyond autism treatment: The application of applied behavior analysis in the treatment of emotional and psychological disorders," *International Journal of Behavioral Consultation and Therapy*, vol. 3, no. 4, p. 528, 2007.
- [15] E. S. Dewi, "Klasifikasi Autism Spectrum Disorder Menggunakan Algoritma Naive Bayes," *MATHunesa: Jurnal Ilmiah Matematika*, vol. 9, no. 1, pp. 27-35, 2021.
- [16] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi deep learning menggunakan convolutional neural network (CNN) pada ekspresi manusia," *Algor*, vol. 2, no. 1, pp. 12-20, 2020.
- [17] S. Yuliany and A. N. Rachman, "Implementasi Deep Learning pada Sistem Klasifikasi Hama Tanaman Padi Menggunakan Metode Convolutional Neural Network (CNN)," *Jurnal Buana Informatika*, vol. 13, no. 1, pp. 54-65, 2022.

- 
- [18] E. Hermawan, "Klasifikasi Pengenalan Wajah Menggunakan Masker atau Tidak Dengan Mengimplementasikan Metode CNN (Convolutional Neural Network)," *Jurnal Industri Kreatif dan Informatika Series (JIKIS)*, vol. 1, no. 1, pp. 33-43, 2021.
- [19] H. Herdianto and D. Nasution, "Implementasi Metode CNN Untuk Klasifikasi Objek," *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 7, no. 1, pp. 54-60, 2023.
- [20] M. A. Nurdin, R. C. Wihandika, and F. Utaminigrum, "Deteksi Pergerakan Arah Mata menggunakan Convolution Neural Network berdasarkan Facial Landmark," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 10, pp. 3338-3345, 2020.
- [21] D. L. Z. Astuti and S. Samsuryadi, "Kajian Pengenalan Ekspresi Wajah menggunakan Metode PCA dan CNN," in *Annual Research Seminar (ARS)*, vol. 4, no. 1, pp. 293-297, 2019.
- [22] A. A. Lubis, "Klasifikasi Citra Multi Wajah Menggunakan Domain Adaptive Faster Region Convolutional Neural Network," *Jurnal SIFO Mikroskil*, vol. 20, no. 2, pp. 159-168, 2019.
- [23] Hidayat, M. A., Husni, N. L., and Damsi, F., "Pendeteksi Banjir Dengan Image Processing Berbasis Convolutional Neural Network (CNN) pada Kamera Pengawas: Image Processing Based Flood Detector Using Convolutional Neural Network (CNN) Within Surveillance Camera," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 2, no. 2, pp. 10-18, 2022.
- [24] M. Sholawati, K. Auliasari, and F. X. Ariwibisono, "Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (CNN)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 1, pp. 134-144, 2022.