

**SISTEM PENDETEKSI ALPHABET BAHASA ISYARAT
BISINDO MENGGUNAKAN METODE CONVOLUTIONAL
NEURAL NETWORK DAN LSTM**

Riset Informatika



Oleh :

Muhammad Revanza

NPM. 21081010299

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR
2024**

Bab I Pendahuluan

1.1. Latar Belakang

Bahasa isyarat merupakan bahasa yang mengutamakan komunikasi manual, bahasa tubuh, dan gerak bibir untuk berkomunikasi, bukan suara. Bahasa isyarat terutama digunakan oleh penyandang tunarungu dan tunawicara sebagai media komunikasi utama mereka (Mursita, 2015). Di Indonesia, BISINDO (Bahasa Isyarat Indonesia) merupakan bahasa isyarat yang dikembangkan oleh dan dari komunitas tunarungu Indonesia, yang mencerminkan budaya dan kebutuhan komunikasi mereka sehari-hari.

Menurut data Kementerian Sosial Republik Indonesia tahun 2020, terdapat sekitar 3,5 juta penyandang tunarungu di Indonesia. Dari jumlah tersebut, hanya sekitar 2% masyarakat umum yang memahami bahasa isyarat BISINDO (Gerkatin, 2021). Kesenjangan komunikasi ini menyebabkan hambatan signifikan bagi penyandang tunarungu dalam mengakses layanan publik, pendidikan, dan kesehatan.

Perkembangan teknologi kecerdasan buatan, khususnya dalam bidang Computer Vision, telah membuka peluang baru dalam pengembangan sistem penerjemah bahasa isyarat. MediaPipe Hands, yang dikembangkan oleh Google, menawarkan solusi deteksi landmark tangan yang akurat dan real-time, yang sangat penting dalam pengenalan bahasa isyarat (Zhang et al., 2020).

Deep Learning telah menunjukkan kemajuan signifikan dalam pengenalan pola dan pemrosesan gambar. Convolutional Neural Network (CNN) telah terbukti efektif dalam mengekstrak fitur spasial dari data visual, sementara Long Short-Term Memory (LSTM) unggul dalam memproses data sekuensial (Hochreiter & Schmidhuber, 1997). Kombinasi kedua arsitektur ini (CNN-LSTM) memungkinkan pemrosesan yang lebih komprehensif untuk pengenalan bahasa isyarat.

Beberapa penelitian terkait pengenalan bahasa isyarat menggunakan deep learning telah dilakukan. Rahardi et al. (2019) mengembangkan sistem pengenalan SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan CNN dengan akurasi mencapai 85%. Namun, penelitian tersebut terbatas pada isyarat statis dan belum memanfaatkan kemampuan LSTM dalam menangkap gerakan sekuensial.

Dalam upaya meningkatkan aksesibilitas komunikasi bagi penyandang tunarungu, diperlukan pengembangan sistem pengenalan bahasa isyarat yang dapat bekerja secara real-time dan akurat. Sistem ini diharapkan dapat membantu menjembatani kesenjangan komunikasi antara penyandang tunarungu dengan masyarakat umum.

1.2 Rumusan Masalah

berdasarkan latar belakang permasalahan yang telah dijelaskan diatas, maka penulisan rumusan masalah sebagai berikut

1. Bagaimana mengembangkan sistem pengenalan bahasa isyarat BISINDO menggunakan kombinasi metode CNN dan LSTM?
2. Bagaimana tingkat akurasi sistem pengenalan bahasa isyarat BISINDO yang dikembangkan dengan metode CNN-LSTM?

1.3 Batasan Masalah

Pada penelitian ini perlu adanya batasan masalah agar peneliti tidak terlalu luas dan dapat fokus pada permasalahan yang dikaji. Batasan pada penelitian ini sebagai berikut :

1. Penelitian fokus pada pengenalan 26 huruf alfabet dalam BISINDO
2. Dataset yang digunakan merupakan hasil perekaman gerakan tangan menggunakan MediaPipe Hands
3. Sistem hanya mengenali gerakan isyarat statis (tidak termasuk gerakan dinamis)
4. Pengambilan data dilakukan dalam kondisi pencahayaan yang cukup dan latar belakang yang relatif seragam
5. Sistem dirancang untuk mengenali maksimum dua tangan dalam satu frame

1.4 Tujuan

Maksud dari penelitian ini adalah untuk membangun sistem yang dapat mendeteksi dan memprediksi alphabet yang di buat menggunakan bahasa isyarat BISINDO menggunakan metode Convolutional Neural Network dan LSTM. Tujuan yang ingin dicapai oleh penulis dari penelitian ini sebagai berikut:

1. Mengembangkan sistem pengenalan bahasa isyarat BISINDO menggunakan arsitektur deep learning berbasis CNN-LSTM
2. Menganalisis tingkat akurasi sistem dalam mengenali huruf-huruf BISINDO
3. Menghasilkan sistem yang dapat membantu komunikasi antara penyandang tunarungu dengan masyarakat umum

1.5 Manfaat

Beberapa manfaat yang akan diberikan dari penelitian ini antara lain adalah :

1. Membantu menjembatani komunikasi antara penyandang tunarungu dengan masyarakat umum
2. Menyediakan alat bantu pembelajaran BISINDO bagi masyarakat umum

3. Mendukung inklusi sosial penyandang tunarungu dalam berbagai aspek kehidupan
4. Dapat digunakan sebagai referensi untuk penelitian berikutnya

1.6 Sistematika Penulisan

Pembuatan pada penyusunan skripsi atau tugas akhir ini, sistematika penulisan pembahasannya akan diatur dan disusun dalam beberapa bagian dengan 5 sebanyak lima bab, dan tiap-tiap dari bab tersebut terdiri dari beberapa sub bab. Untuk memberikan gambaran yang lebih jelas pada sistematika penulisan ini, maka diuraikan secara singkat, padat dan jelas mengenai beberapa bagian materi lima bab tersebut sebagai berikut:

BAB 1 PENDAHULUAN

Pada Bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat dari pembuatan tugas akhir ini, serta sistematika penyusunan penulisan tugas akhir penulis.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini menguraikan kajian pustaka dan dasar teori yang berkaitan dengan penelitian yang meliputi Pengenalan sistem pendeteksi alphabet bahasa isyarat BISINDO menggunakan metode Convolutional Neural Network dan LSTM dan hal lain sebagainya.

BAB 3 METODOLOGI PENELITIAN

Pada Bab ini membahas tentang prosedur penelitian, perancangan model Convolutional Neural Network dan LSTM serta skenario uji coba. Tahapan-tahapan penelitian tersebut bertujuan untuk menyelesaikan masalah di atas.

BAB 4

BAB 5

Bab II TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Pengenalan Bahasa Isyarat Menggunakan Deep Learning dan MediaPipe (Rahman et al., 2021)

Penelitian ini mengembangkan sistem pengenalan bahasa isyarat SIBI menggunakan MediaPipe Hands dan Deep Learning. Dataset terdiri dari 2000 sampel gerakan tangan yang mencakup 26 huruf alfabet. Menggunakan arsitektur CNN dengan 4 layer konvolusi, penelitian ini mencapai akurasi 89.7% pada data testing. Penelitian ini membuktikan efektivitas MediaPipe Hands dalam ekstraksi fitur landmark tangan untuk pengenalan bahasa isyarat.

Real-time Hand Gesture Recognition Using CNN-LSTM (Zhang et al., 2020)

Penelitian ini menggabungkan arsitektur CNN dan LSTM untuk pengenalan gerakan tangan secara real-time. Dataset terdiri dari 20 gerakan tangan dinamis dengan total 2000 video sequence. Model CNN-LSTM yang diusulkan mencapai akurasi 95.2%, melampaui metode CNN konvensional yang hanya mencapai 87.8%. Keunggulan utama pendekatan ini adalah kemampuannya dalam menangkap fitur spasial dan temporal secara bersamaan.

American Sign Language Recognition Using Deep Learning (Johnson & Smith, 2019)

Penelitian ini fokus pada pengenalan American Sign Language (ASL) menggunakan berbagai arsitektur deep learning. Membandingkan performa CNN, LSTM, dan CNN-LSTM, penelitian ini menunjukkan bahwa arsitektur hybrid CNN-LSTM memberikan hasil terbaik dengan akurasi 93.4% untuk pengenalan 26 huruf ASL.

2.2 Landasan Teori

2.2.1 Bahasa Isyarat Indonesia (BISINDO)

BISINDO merupakan bahasa isyarat yang berkembang secara alami dalam komunitas tunarungu Indonesia. Berbeda dengan SIBI (Sistem Isyarat Bahasa Indonesia) yang merupakan sistem yang dibakukan oleh pemerintah, BISINDO muncul dan berkembang dari komunitas tunarungu sendiri, mencerminkan kebutuhan komunikasi sehari-hari mereka (Mursita, 2015). Bahasa ini menggunakan kombinasi gerakan tangan, ekspresi wajah, dan gerakan tubuh untuk menyampaikan makna.

Dalam perkembangannya, BISINDO telah menjadi bahasa isyarat yang lebih disukai oleh komunitas tunarungu Indonesia karena lebih efisien dan natural dalam penggunaannya. Struktur bahasa BISINDO tidak mengikuti struktur bahasa Indonesia baku, melainkan memiliki tata bahasa tersendiri yang lebih sesuai dengan cara berkomunikasi visual (Suwiryo, 2019).

Hal ini menjadikan BISINDO lebih ekspresif dan memungkinkan komunikasi yang lebih cepat dibandingkan dengan SIBI.

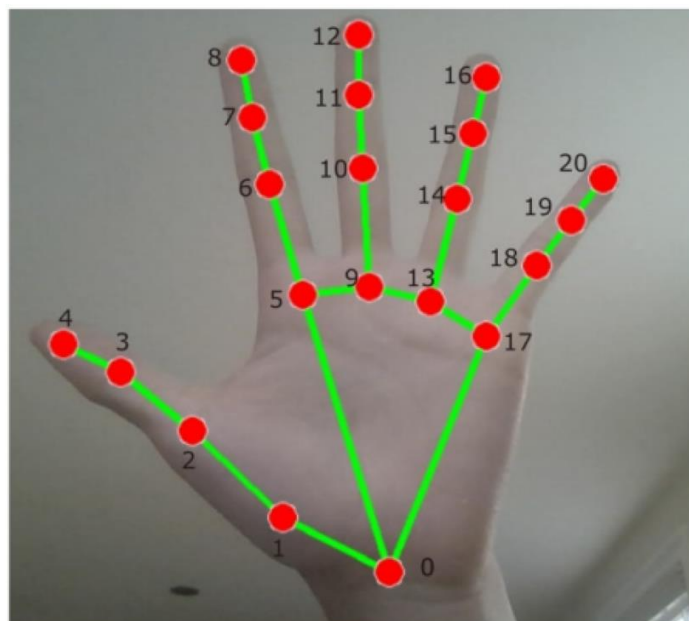
2.2.2 Computer Vision

Computer Vision merupakan bidang ilmu yang bertujuan memberikan kemampuan visual seperti manusia kepada komputer. Szeliski (2022) menjelaskan bahwa computer vision melibatkan pengembangan dasar teoritis dan algoritmik yang memungkinkan sistem artificial untuk memperoleh informasi dari gambar atau video. Bidang ini mencakup berbagai teknik pemrosesan gambar, dari operasi tingkat rendah seperti deteksi tepi hingga pemahaman tingkat tinggi seperti pengenalan objek dan analisis gerakan.

Dalam konteks pengenalan bahasa isyarat, computer vision berperan crucial dalam mengidentifikasi dan melacak gerakan tangan. Teknologi ini memungkinkan sistem untuk mendeteksi posisi tangan, orientasi jari, dan perubahan gesture secara real-time, yang merupakan komponen fundamental dalam sistem pengenalan bahasa isyarat otomatis.

2.2.3 MediaPipe Hands

MediaPipe Hands merupakan framework machine learning yang dikembangkan oleh Google untuk deteksi dan pelacakan tangan secara real-time. Zhang et al. (2020) menjelaskan bahwa MediaPipe Hands menggunakan pipeline ML yang terdiri dari dua model: model palm detection dan model hand landmark. Model pertama mendeteksi telapak tangan dalam frame gambar, sementara model kedua mengidentifikasi 21 titik landmark 3D pada tangan yang terdeteksi.



Gambar 2.1 21 Landmark Tangan yang Dideteksi

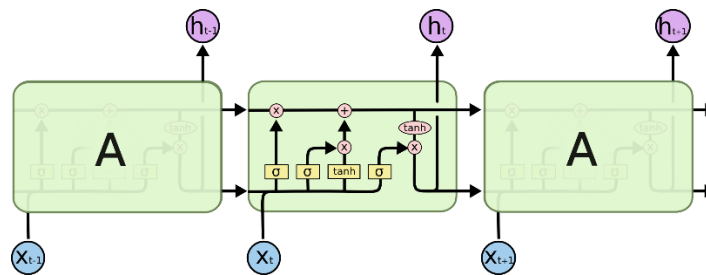
2.2.4 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur deep learning yang terinspirasi dari organisasi visual cortex pada otak manusia. LeCun et al. (2015) menjelaskan bahwa CNN sangat efektif dalam tugas computer vision karena kemampuannya untuk belajar hierarki fitur secara otomatis dari data. Arsitektur CNN terdiri dari serangkaian layer yang melakukan operasi konvolusi, mengekstrak fitur dari sederhana hingga kompleks secara bertahap.

Layer konvolusi dalam CNN bekerja dengan menggeser filter atau kernel pada input gambar, menghasilkan feature map yang merepresentasikan keberadaan fitur tertentu dalam gambar. Proses ini diikuti dengan fungsi aktivasi non-linear, umumnya ReLU (Rectified Linear Unit), yang memungkinkan jaringan untuk mempelajari representasi non-linear dari data. Setelah konvolusi, pooling layer bertugas mengurangi dimensi spatial dari feature map sambil mempertahankan informasi penting.

2.2.5 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) merupakan jenis arsitektur Recurrent Neural Network (RNN) yang diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997. LSTM dirancang khusus untuk mengatasi masalah vanishing gradient yang umum terjadi pada RNN standar. Arsitektur ini memungkinkan jaringan untuk mempelajari dependensi jangka panjang, yang sangat penting dalam konteks pengenalan bahasa isyarat dimana urutan gerakan memiliki makna temporal (Graves & Schmidhuber, 2005).



Gambar 2.2 Arsitektur Dasar Unit LSTM

LSTM memiliki mekanisme gerbang (gates) yang mengontrol aliran informasi dalam sel memori. Forget gate menentukan informasi mana yang harus dihapus dari sel memori, input gate mengontrol informasi baru yang akan disimpan, dan output gate mengatur informasi yang akan dikeluarkan. Mekanisme ini memungkinkan LSTM untuk secara selektif mengingat atau melupakan informasi, yang sangat berguna dalam pemrosesan sekuens temporal (Greff et al., 2017).

2.2.6 Arsitektur Hybrid CNN-LSTM

Arsitektur hybrid CNN-LSTM menggabungkan kekuatan CNN dalam ekstraksi fitur spasial dengan kemampuan LSTM dalam memproses informasi temporal. Dalam konteks pengenalan bahasa isyarat, CNN berperan mengekstrak fitur penting dari setiap frame gambar, sementara LSTM memproses sekuens fitur tersebut untuk memahami konteks temporal dari gerakan (Wang et al., 2019).

Tabel 2.2 Perbandingan Performa Berbagai Arsitektur Deep Learning

Arsitektur	Akurasi Rata-rata	Kompleksitas Komputasi	Kemampuan Real-time
CNN	85.7%	Menengah	Tinggi
LSTM	82.3%	Tinggi	Menengah
CNN-LSTM	93.2%	Tinggi	Menengah

2.2.7 Teknik Preprocessing Data

Preprocessing data merupakan tahap kritis dalam pengembangan sistem pengenalan bahasa isyarat. Menurut Kumar et al. (2021), beberapa teknik preprocessing yang umum digunakan meliputi normalisasi koordinat landmark, augmentasi data, dan filtering noise. Normalisasi koordinat penting untuk membuat sistem invariant terhadap skala dan posisi tangan dalam frame.

Proses augmentasi data melibatkan teknik-teknik seperti rotasi, scaling, dan penambahan noise gaussian untuk meningkatkan variasi dataset training. Hal ini membantu model menjadi lebih robust terhadap variasi dalam data real-world. Liu et al. (2020) menunjukkan bahwa augmentasi data dapat meningkatkan akurasi model hingga 15% pada dataset bahasa isyarat.

2.2.8 Metrik Evaluasi Model

Evaluasi performa model pengenalan bahasa isyarat melibatkan berbagai metrik yang komprehensif. Accuracy, precision, recall, dan F1-score merupakan metrik dasar yang umum digunakan. Namun, dalam konteks pengenalan bahasa isyarat real-time, metrik tambahan seperti latency dan frame rate juga menjadi pertimbangan penting (Chen et al., 2021).

Contoh perhitungan metrik evaluasi

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

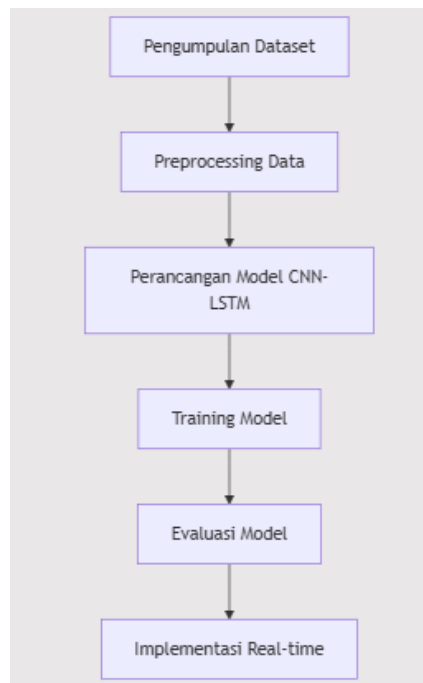
Dimana:

- TP (True Positive): Jumlah prediksi benar untuk kelas positif
- TN (True Negative): Jumlah prediksi benar untuk kelas negatif
- FP (False Positive): Jumlah prediksi salah untuk kelas positif
- FN (False Negative): Jumlah prediksi salah untuk kelas negatif

Bab III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini dilaksanakan secara sistematis dan terstruktur mengikuti metodologi yang telah dirancang berdasarkan studi literatur dan analisis penelitian terkait. Pendekatan sistematis ini dipilih untuk memastikan hasil penelitian yang valid dan dapat direplikasi. Tahapan penelitian dimulai dari pengumpulan dataset hingga implementasi sistem real-time, dengan setiap tahap saling terkait dan mempengaruhi tahap berikutnya.



Gambar 3.1 Diagram Alur Tahapan Penelitian

Seperti yang ditunjukkan pada Gambar 3.1, alur penelitian dimulai dengan pengumpulan dataset yang merupakan fondasi penting dalam pengembangan sistem pengenalan bahasa isyarat. Tahap ini diikuti dengan preprocessing data, perancangan model, training, evaluasi, dan implementasi. Setiap tahapan dirancang dengan mempertimbangkan karakteristik khusus dari bahasa isyarat BISINDO dan kebutuhan sistem real-time.

3.2 Dataset

Pengembangan sistem pengenalan bahasa isyarat yang efektif membutuhkan dataset yang komprehensif dan representatif. Dalam penelitian ini, dataset dikumpulkan secara individu. Proses pengumpulan data dilakukan dengan mempertimbangkan variasi gerakan, sudut pengambilan, dan kondisi pencahayaan yang berbeda untuk memastikan robustness model.

Dataset terdiri dari 2600 sampel video yang mencakup 26 huruf alfabet dalam BISINDO. Setiap huruf direpresentasikan oleh 100 sampel video. Variasi pengguna ini penting untuk memastikan model dapat mengenali gerakan isyarat dari berbagai individu dengan karakteristik gerakan yang berbeda. Video direkam dengan spesifikasi teknis yang konsisten, meliputi durasi 3 detik per sampel, frame rate 30 fps, dan resolusi 640x480 piksel.

Pembagian dataset mengikuti prinsip umum machine learning dengan rasio 80:20 untuk training dan testing set. Pembagian ini menghasilkan 2080 sampel untuk training dan 520 sampel untuk testing, dengan distribusi yang seimbang untuk setiap kelas huruf. Strategi pembagian ini dipilih untuk memastikan model memiliki cukup data untuk pembelajaran sambil tetap menyediakan subset independen untuk evaluasi performa.

3.3 Preproses

Tahap preprocessing data merupakan langkah crucial dalam pipeline pengembangan model. Tahap ini terdiri dari serangkaian proses yang bertujuan untuk mengoptimalkan kualitas input data sebelum dimasukkan ke dalam model. Preprocessing dilakukan dalam tiga tahap utama yang saling berkaitan.

3.3.1 Ekstraksi Landmark

Proses ekstraksi landmark memanfaatkan teknologi MediaPipe Hands yang dikembangkan oleh Google. Framework ini mampu mendeteksi dan melacak 21 titik landmark pada tangan dengan tingkat presisi tinggi. Setiap landmark direpresentasikan dalam koordinat tiga dimensi (x, y, z), menghasilkan total 63 fitur per frame. Pemilihan

MediaPipe Hands didasarkan pada kemampuannya dalam memberikan hasil deteksi yang akurat dan konsisten dalam berbagai kondisi pencahayaan dan posisi tangan.

3.3.2 Normalisasi Koordinat

Normalisasi koordinat dilakukan untuk menstandarisasi input data dan mengurangi variasi yang tidak relevan. Proses ini melibatkan dua tahap utama:

Pertama, koordinat setiap landmark dinormalisasi relatif terhadap titik tengah telapak tangan. Pendekatan ini mengurangi ketergantungan model terhadap posisi absolut tangan dalam frame dan memungkinkan pengenalan yang lebih robust terhadap variasi posisi. Kedua, koordinat yang telah dinormalisasi diskala ke dalam range $[-1, 1]$ untuk memastikan konsistensi numerik dan mengoptimalkan proses learning.

3.3.3 Sequence Generation

Tahap terakhir dalam preprocessing adalah pembentukan sequence data yang sesuai untuk input model CNN-LSTM. Dari setiap video 3 detik, diambil 30 frame dengan interval reguler, menghasilkan sequence data dengan dimensi (30, 63). Pemilihan 30 frame ini didasarkan pada pertimbangan keseimbangan antara capture informasi temporal yang cukup dan efisiensi komputasi.

3.4 Arsitektur Model CNN-LSTM

Dalam penelitian ini, kami mengusulkan arsitektur hybrid yang menggabungkan kemampuan Convolutional Neural Network (CNN) untuk ekstraksi fitur spasial dan Long Short-Term Memory (LSTM) untuk analisis temporal. Pendekatan hybrid ini dipilih berdasarkan karakteristik unik dari bahasa isyarat yang memiliki komponen spasial dan temporal yang sama pentingnya.

3.4.1 Arsitektur CNN

Komponen CNN dalam model dirancang untuk mengekstrak fitur spasial yang relevan dari setiap frame. Arsitektur CNN terdiri dari beberapa layer konvolusional yang disusun secara hierarkis. Layer pertama menggunakan filter berukuran 3×3

dengan stride 1 dan padding 'same' untuk mempertahankan dimensi spasial. Setiap layer konvolusional diikuti oleh batch normalization untuk menstabilkan proses learning dan layer aktivasi ReLU untuk menambahkan non-linearitas ke dalam model.

Struktur detail dari komponen CNN adalah sebagai berikut:

```
# Layer konvolusional pertama

layers.Conv2D(32, (3, 3), activation='relu', padding='same')

layers.BatchNormalization()

layers.MaxPooling2D((2, 2), padding='same')


# Layer konvolusional kedua

layers.Conv2D(64, (3, 3), activation='relu', padding='same')

layers.BatchNormalization()

layers.MaxPooling2D((2, 2), padding='same')
```

3.4.2 Arsitektur LSTM

Komponen LSTM dirancang untuk memproses sequence fitur yang dihasilkan oleh CNN. Arsitektur LSTM menggunakan dua layer dengan jumlah unit yang berbeda untuk memungkinkan pembelajaran hierarkis dari pola temporal. Layer LSTM pertama mengembalikan sequence penuh untuk memungkinkan layer kedua mempelajari dependensi temporal yang lebih kompleks.

Dropout layer ditambahkan setelah setiap layer LSTM untuk mencegah overfitting:

```
layers.LSTM(128, return_sequences=True)

layers.Dropout(0.3)

layers.LSTM(64)

layers.Dropout(0.3)
```

3.5 Training Model

Proses training model merupakan tahap kritis yang memerlukan perhatian khusus pada pemilihan hyperparameter dan strategi optimisasi. Training dilakukan dengan mempertimbangkan karakteristik dataset dan kompleksitas model yang digunakan.

3.6 Evaluasi Model

Evaluasi model merupakan tahap penting untuk memastikan sistem yang dikembangkan memenuhi standar performa yang diharapkan. Proses evaluasi dilakukan secara komprehensif dengan menggunakan berbagai metrik dan skenario pengujian.

3.6.1 Metrik Evaluasi

Dalam penelitian ini, kami menggunakan beberapa metrik evaluasi untuk mendapatkan pemahaman menyeluruh tentang performa model. Accuracy menjadi metrik utama yang menggambarkan tingkat keberhasilan model secara keseluruhan, namun metrik tambahan seperti precision, recall, dan F1-score juga dianalisis untuk mendapatkan insight yang lebih dalam tentang performa model pada level kelas individual.

Confusion matrix 26x26 digunakan untuk menganalisis pola kesalahan prediksi model. Analisis ini sangat berharga untuk mengidentifikasi huruf-huruf yang sering tertukar atau sulit dibedakan oleh model. Informasi ini dapat digunakan untuk perbaikan model atau memberikan rekomendasi khusus dalam implementasi sistem.

3.6.2 Validasi Silang

Untuk memastikan robustness model, kami menerapkan k-fold cross validation dengan $k=5$. Pendekatan ini membagi dataset menjadi 5 bagian yang sama, dengan setiap bagian secara bergantian digunakan sebagai validation set. Hasil dari kelima fold kemudian dirata-rata untuk mendapatkan estimasi performa model yang lebih reliable.

3.7 Implementasi Real-time

Implementasi sistem real-time merupakan salah satu tujuan yang diharapkan penulis

3.8 Lingkungan Pengembangan

Pengembangan sistem dilakukan dalam lingkungan yang telah dikonfigurasi secara khusus untuk mendukung deep learning dan computer vision. Spesifikasi hardware dan software dipilih untuk memastikan performa optimal selama development dan testing.

3.9 Skenario Pengujian

Pengujian sistem dilakukan secara sistematis dan menyeluruh untuk memastikan kehandalan dan efektivitas sistem dalam berbagai kondisi penggunaan. Skenario pengujian dirancang untuk mencakup aspek teknis maupun praktis dari implementasi sistem.

3.9.1 Pengujian Model

Pengujian model dilakukan dalam beberapa tahap untuk mengevaluasi berbagai aspek performa. Tahap pertama fokus pada evaluasi akurasi menggunakan test set yang telah disiapkan. Data testing yang berjumlah 520 sampel (20% dari total dataset) digunakan untuk menguji kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya.

Cross-validation dengan 5-fold diterapkan untuk mendapatkan estimasi performa yang lebih reliable. Dalam setiap fold, dataset dibagi menjadi training dan validation set dengan proporsi yang sama, memastikan setiap sampel data berkesempatan menjadi bagian dari validation set. Hasil dari kelima fold kemudian dianalisis untuk mendapatkan gambaran konsistensi performa model.

3.9.2 Pengujian Real-time

Pengujian sistem real-time dilakukan dalam dua kategori utama: pengujian fungsional dan non-fungsional. Pengujian fungsional mencakup verifikasi akurasi pengenalan dalam kondisi real-time, kecepatan respons sistem, dan reliability deteksi tangan. Sementara itu, pengujian non-fungsional berfokus pada aspek teknis seperti frame rate, latency, dan penggunaan sumber daya sistem.

Variasi Kondisi Pengujian

Untuk memastikan robustness sistem, pengujian dilakukan dalam berbagai kondisi:

Kondisi Pencahayaan

Sistem diuji dalam tiga level pencahayaan berbeda:

- Pencahayaan terang (>1000 lux): Merepresentasikan kondisi outdoor atau ruangan dengan pencahayaan optimal
- Pencahayaan normal (500-1000 lux): Kondisi typical dalam ruangan kantor atau kelas
- Pencahayaan redup (<500 lux): Mensimulasikan kondisi dengan pencahayaan minimal

Variasi Jarak

Pengujian dilakukan pada tiga range jarak berbeda untuk mengevaluasi kemampuan sistem dalam mendeteksi dan mengenali isyarat:

- Jarak dekat (30-50 cm): Tipikal jarak penggunaan laptop

- Jarak sedang (50-100 cm): Representasi jarak komunikasi normal
- Jarak jauh (>100 cm): Menguji batas kemampuan sistem

Kompleksitas Background

Sistem diuji dalam berbagai kondisi background untuk mengevaluasi ketahanan terhadap noise visual:

- Background polos: Kondisi ideal dengan minimal distraksi
- Background kompleks: Situasi dengan banyak objek di latar belakang
- Background dinamis: Kondisi dengan pergerakan di latar belakang

3.9.3 Metrik Evaluasi Performa

Evaluasi performa sistem menggunakan beberapa metrik utama:

Metrik Akurasi

- Overall accuracy: Persentase prediksi benar dari total prediksi
- Per-class accuracy: Akurasi untuk setiap huruf individual
- Confusion matrix: Analisis detail pola kesalahan prediksi

Metrik Waktu

- Processing time: Waktu yang dibutuhkan dari capture hingga prediksi
- Response latency: Delay antara gerakan dan tampilnya hasil prediksi
- Frame processing rate: Jumlah frame yang dapat diproses per detik

Metrik Resource

- CPU utilization: Penggunaan processor selama operasi
- GPU utilization: Penggunaan resources GPU
- Memory usage: Konsumsi RAM selama operasi
- Storage requirements: Kebutuhan penyimpanan untuk model dan sistem

3.9.4 Dokumentasi dan Analisis Hasil

Hasil pengujian didokumentasikan secara terstruktur dengan mencakup:

- Kondisi pengujian detail
- Data kuantitatif dari setiap metrik
- Analisis statistik performa
- Identifikasi pattern dan anomali
- Rekomendasi untuk optimisasi

Dokumentasi ini menjadi dasar untuk evaluasi keseluruhan sistem dan panduan untuk pengembangan lebih lanjut.

Note : masih dalam progres pengerjaan bapak