

# IDENTIFIKASI CITRA POSE TEGAK UNTUK PEMILIHAN ANGGOTA BARIS BERBARIS MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DI SMK MUHAMMADIYAH MUNGKID

Baetoha<sup>1\*</sup>, Supatman<sup>2</sup>

<sup>1,2</sup>Universitas Mercubuana Yogyakarta; Jl. Jembatan Merah No. 84C Gejayan; 0274-584922

## Riwayat artikel:

Received: 30 November 2022

Accepted: 29 Desember 2023

Published: 1 Januari 2024

## Keywords:

Image Identification;

Body Pose;

Upright and Less Upright;

Convolutional Neural Network.

## Correspondent Email:

201130219@student.mercubuana-yogya.ac.id

**Abstrak.** Penelitian ini membahas bagaimana penerapan metode CNN dapat digunakan untuk menciptakan sebuah sistem yang secara otomatis dan akurat mengidentifikasi pose berdiri tegak untuk baris berbaris, dengan menggunakan metode Convolutional Neural Network (CNN) yang merupakan metode klasifikasi yang sangat baik untuk menganalisis gambar digital. Desain CNN dibuat khusus untuk menganalisis objek dalam gambar, sehingga sangat berguna untuk klasifikasi gambar. Diharapkan dengan menggunakan metode ini bersama dengan data yang relevan, hasilnya adalah tingkat ketajaman dan efisiensi yang lebih tinggi. Penelitian ini menghasilkan sebuah sistem yang dapat mengidentifikasi pose tubuh kurang tegak dengan akurasi sebesar 59% dengan confidence 58,54% dan mengidentifikasi pose tegak dengan akurasi sebesar 80% dengan confidence 79,36% berdasarkan gambar yang di upload.

**Abstract.** This research discusses how the implementation of the CNN method can be utilized to create a system that automatically and accurately identifies an upright standing posture for marching in a line. By employing the Convolutional Neural Network (CNN) method, which is an excellent classification technique for analyzing digital images, the design of CNN is specifically crafted for object analysis in images, making it highly useful for image classification. It is anticipated that, by using this method in conjunction with relevant data, the results will yield a higher level of precision and efficiency. The research yields a system capable of identifying a less upright body posture with an accuracy of 59% and a confidence level of 58.54%. Additionally, it identifies an upright posture with an accuracy of 80% and a confidence level of 79.36% based on the uploaded images.

## 1. PENDAHULUAN

Pose tubuh yang tegak pada baris berbaris memiliki kepentingan yang signifikan. Selain mencerminkan disiplin dan ketertiban, postur yang tegak juga memberikan tampilan bersih

dan rapi pada barisan, menciptakan kesan kekompakan dan profesionalisme. Penting untuk keamanan dan kesehatan, karena memegang pose yang benar dapat mencegah cedera dan masalah kesehatan jangka panjang.

Keseragaman dalam pose tubuh memastikan barisan terlihat seragam dan sejajar, menciptakan estetika yang kuat. Pose yang tegak juga mendukung komunikasi efektif antara pemimpin dan anggota baris berbaris, sambil meningkatkan fokus mental dan membentuk karakter melalui nilai-nilai seperti kedisiplinan dan tanggung jawab. Dengan demikian, pose tubuh yang tegak bukan hanya menciptakan penampilan yang baik secara visual, tetapi juga mendukung tujuan pembentukan karakter dan kepemimpinan.

Salah satu pendekatan yang kuat dan efektif dalam pengenalan pola pada citra adalah Convolutional Neural Network (CNN). CNN telah terbukti sangat sukses dalam berbagai tugas pengenalan pola, termasuk dalam klasifikasi citra dan pengenalan objek. Kemampuan CNN untuk secara otomatis mengekstraksi fitur-fitur penting dari data citra, tanpa memerlukan fitur manual yang rumit, menjadikannya pilihan yang menjanjikan untuk tugas klasifikasi pose tubuh yang tegak.

Convolutional Neural Network (CNN) adalah metode klasifikasi yang sangat baik untuk menganalisis gambar digital. Desain CNN dibuat khusus untuk menganalisis objek dalam gambar, sehingga sangat berguna untuk klasifikasi gambar. Diharapkan dengan menggunakan metode ini bersama dengan data yang relevan, hasilnya adalah tingkat ketajaman dan efisiensi yang lebih tinggi [2]. CNN merupakan pengembangan dari Multilayer Perceptron (MLP) yang dirancang untuk menampilkan data dalam dua dimensi. CNN diklasifikasikan sebagai Deep Neural Network karena memiliki jumlah koneksi yang signifikan dan biasanya digunakan untuk memproses data gambar. Meskipun secara teoritis klasifikasi gambar dapat dilakukan dengan menggunakan MLP, praktik ini kurang ideal karena dapat meningkatkan jumlah informasi spasial dalam data gambar secara signifikan. Metode ini melibatkan penerapan setiap foto sebagai fungsi kustom, sehingga menghasilkan performa yang tidak optimal [1].

Pada penelitian pertama yang dilakukan oleh [1] yang berjudul "Identifikasi Citra Tulisan Tangan Digital Menggunakan Convolutional Neural Network (CNN)" menghasilkan dengan menggunakan metode CNN menunjukkan akurasi sebesar 98,6 % untuk dataset MNIST dan sebesar 88% untuk

data koresponden. Hal ini dapat disimpulkan bahwa metode CNN mampu mengenali tulisan angka dengan baik.

Selanjutnya pada penelitian kedua yang dilakukan oleh [2] yang berjudul "Identifikasi dan Prediksi Umur serta Jenis Kelamin Berdasarkan Citra Wajah Menggunakan Algoritma Convolutional Neural Network (CNN)" menghasilkan bahwa metode yang digunakan ini efektif dalam mendeteksi umur dan jenis kelamin pada citra foto yang diberikan. Berdasarkan pada uji program yang telah dilakukan, diberikan 7 sampel foto dari beberapa umur dan kelamin yang berbeda. Pada keseluruhan foto sampel yang diberikan menghasilkan deteksi yang sesuai pada jenis kelaminnya, namun masih terdapat beberapa kesalahan pada prediksi umur apabila dibandingkan dengan umur aslinya. Hal tersebut disebabkan oleh kualitas foto yang kurang jelas (blur).

Pada penelitian ketiga yang dilakukan oleh [3] yang berjudul "Identifikasi Objek Cagar Budaya Candi Mahligai Berbasis Citra Digital Menggunakan Mask R-CNN" menghasilkan Penerapannya dilakukan pada gambar Candi Mahligai. menggunakan Confusion Matrix dan diperoleh akurasi sebesar 82.0%, presisi 78.57% dan recall 88.0%. Identifikasi Kelas Tunggal berhasil dilakukan pada nilai ambang batas 0,776 dengan persentase kesalahan sebesar 18%.

Melalui penelitian ini, penulis berharap bahwa penerapan CNN dapat digunakan untuk menciptakan sebuah sistem yang secara otomatis dan akurat mengidentifikasi pose berdiri tegak untuk baris berbaris.

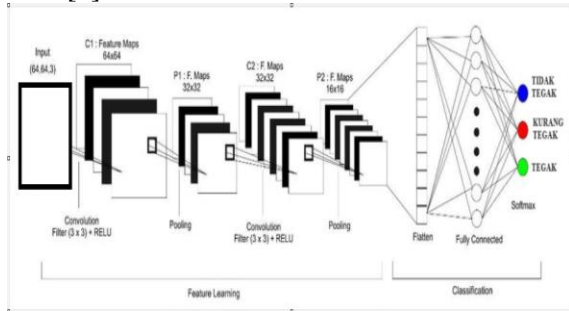
## 2. TINJAUAN PUSTAKA

Pada bagian ini akan menjelaskan tentang teori-teori dan referensi dari buku-buku yang relevan dengan topik atau permasalahan yang sedang dibahas.

### 2.1. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu jenis deep feed-forward artificial neural network yang sering digunakan dalam analisis citra. Struktur CNN terdiri dari satu lapisan masukan (input layer), satu lapisan keluaran (output layer), dan beberapa lapisan tersembunyi (hidden layers). Lapisan

tersembunyi ini umumnya terdiri dari convolutional layers, pooling layers, normalization layers, ReLu layer, fully connected layers, dan loss layer. Semua lapisan dalam CNN disusun secara hierarkis dan ditumpuk berurutan. Berbeda dengan arsitektur dua dimensi pada MLP, CNN menggunakan arsitektur tiga dimensi yang mencakup lebar, tinggi, dan kedalaman dalam pemrosesan data citra [2]



Gambar 1. Arsitektur CNN

## 2.2. Deep Learning

*Deep Learning* merupakan konsep yang mampu meningkatkan performa kecerdasan buatan itu sendiri, sebagai contoh adalah model pengembangan *Deep Learning*. *Deep Learning* dapat mengubah representasi gambar sederhana menjadi konsep yang lebih kompleks secara otomatis, tanpa mengandalkan aturan kode atau pengetahuan manusia dalam domain tersebut. Oleh karena itu, model *Deep Learning* dapat mempercepat dan menyederhanakan tugas-tugas seperti deteksi objek, pengenalan suara, terjemahan bahasa, dan berbagai pekerjaan kecerdasan buatan lainnya. Salah satu metode *Deep Learning* adalah *Convolutional Neural Network* (CNN), yang merupakan turunan dari metode *Artificial Neural Network* (ANN). Meskipun CNN dan ANN memiliki arsitektur dan model yang serupa, namun ada perbedaan mencolok antara keduanya. Pada metode ANN, setiap node dalam jaringan berdiri sendiri, sementara pada CNN, node-node tersebut saling terhubung. Kelebihan ini membuat metode CNN lebih efisien dalam komputasi dibandingkan dengan ANN, sehingga CNN lebih unggul dalam memindai bagian terkecil dari gambar [3]

## 2.3. Tensorflow

TensorFlow adalah sebuah Pustaka perangkat lunak yang bersifat sumber terbuka (open source) yang digunakan untuk mengembangkan, melatih, mengevaluasi, dan

menerapkan model pembelajaran mesin. Banyak pengembang menggunakan Bahasa pemrograman Python sebagai basis untuk membangun sistem dengan menggunakan TensorFlow. Selain itu, ada juga KERAS, yang merupakan salah satu dari beberapa antarmuka (API) tingkat tinggi dari TensorFlow, yang sangat membantu dalam pembuatan dan pelatihan jaringan neural yang mendalam [4], [5]

## 2.4. HyperParameter Optimization

Hyperparameter optimization pada Convolutional Neural Network (CNN) adalah proses mencari dan menentukan nilai optimal untuk hyperparameter pada arsitektur dan pelatihan model CNN. Hyperparameter adalah parameter yang ditentukan di luar model itu sendiri dan berpengaruh terhadap cara model belajar dan berperilaku.[6] Beberapa contoh hyperparameter pada CNN meliputi:

### a. Learning Rate

Learning Rate adalah parameter yang menentukan seberapa besar langkah yang diambil saat mengoptimasi model selama proses pelatihan. Jika learning rate terlalu besar, model dapat melewati titik minimum dan kesulitan untuk konvergensi, dan jika terlalu kecil, proses pelatihan akan menjadi lambat.

### b. Ukuran Batch

Batch Size adalah jumlah sampel datayang diproses dalam satu iterasi selama pelatihan. Batch size yang lebih besar bisa mempercepat pelatihan, tetapi membutuhkan lebih banyak memori GPU.

### c. Dropout Rate

Dropout adalah teknik regularisasi yang digunakan untuk mengurangi overfitting. Dropout rate menentukan seberapa banyak neuron yang di-drop selama pelatihan

### d. Jumlah dan Ukuran Filter

Filter adalah salah satu komponen penting dalam konvolusi. Jumlah dan ukuran filter yang dipilih dapat mempengaruhi ekstraksi fitur dari citra. Jumlah filter menentukan berapa banyak fitur yang diekstraksi dalam setiap layer.

### e. Fungsi Aktivasi

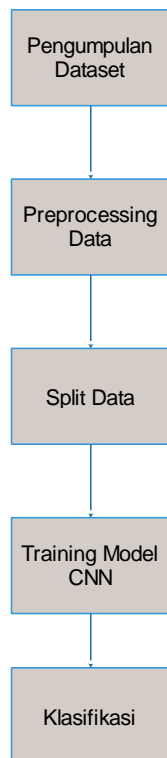
Pilihan fungsi aktivasi (misalnya ReLU, Sigmoid, atau Tanh) pada setiap layer adalah hyperparameter yang harus diatur.

### 3. METODE PENELITIAN

#### 3.1. Tahapan Penelitian

Dalam upaya menyelesaikan penelitian ini, terdapat beberapa langkah penting yang perlu dijalani.

Proses penelitian ini melibatkan serangkaian tahap yang mencakup implementasi dan uji coba metode, bertujuan untuk mencapai hasil yang diinginkan dan mencerminkan gambaran yang diharapkan. Untuk memvisualisasikan langkah-langkah tersebut. Ditujukan pada gambar 2



Gambar 2. Flowchart Penelitian

#### 3.2. Pengumpulan Dataset

Pengumpulan dataset adalah proses mengumpulkan sejumlah data atau sampel yang akan digunakan untuk melatih, menguji, atau menguji kinerja dari suatu model atau algoritma. Dataset yang baik merupakan kunci kesuksesan dalam pelatihan model *machine learning* dan *deep learning*, karena kualitas dan jumlah data yang cukup akan berpengaruh pada akurasi dan generalisasi model. Penelitian ini menggunakan dataset yang berasal dari situs publik, yaitu Kaggle dan disimpan pada *cloud storage* yaitu google drive [7]

#### 3.3. Preprocessing Data

*Preprocessing* merupakan tahapan yang dilakukan sebelum data citra asli diolah oleh

*Convolutional Neural Network* (CNN). Dalam tahapan ini, diterapkan fungsi *resize* citra yang berguna untuk menyesuaikan ukuran citra. *Resize* dilakukan untuk menghilangkan *noise* dengan cara memotong bagian citra yang tidak relevan. Jika citra mengandung *noise* yang berlebihan, citra tersebut akan diabaikan. Tujuan dari proses ini adalah untuk mempercepat dan mempermudah pemrosesan data *training*.

#### 3.4. Split Data

Pada tahap pemisahan data, dataset dibagi menjadi tiga bagian, yaitu data pelatihan, data pengujian, dan data validasi. Data pelatihan digunakan sebagai referensi untuk membangun model klasifikasi, sementara data pengujian digunakan untuk menguji performa dari model algoritma klasifikasi tersebut. Data validasi berfungsi untuk melakukan validasi model dan mencegah terjadinya kondisi di mana akurasi pada data pelatihan tinggi tetapi akurasi pada data pengujian rendah [3], [8].

#### 3.5. Training Model CNN

Dalam tahap pelatihan, digunakan untuk melatih model arsitektur CNN guna mengenali jenis olahraga. Tahap ini memanfaatkan 20% dari total dataset yang ada, dengan ukuran *batch size* sebesar 64 dan ukuran citra 150x150 piksel. Hasil dari pelatihan pada setiap model arsitektur disimpan dalam format ekstensi HDF5. Hasil penyimpanan dari tahap pelatihan mencakup model, bobot pelatihan, dan informasi yang akan digunakan untuk tahap pengujian[10]. Model Arsitektur CNN yang diterapkan yaitu DenseNet121

##### a. DenseNet121

DenseNet 121 adalah jenis *Convolutional Neural Network* (CNN) di mana setiap lapisan terhubung dengan lapisan non-berikutnya yang lebih dalam dalam struktur, artinya, lapisan utama terhubung dengan lapisan kedua, ketiga, keempat, dan seterusnya, lapisan berikutnya terhubung dengan lapisan ketiga, keempat, kelima, dan seterusnya. Hal ini dilakukan untuk memaksimalkan aliran data antara lapisanlapisan dalam struktur tersebut [9], [10]

#### 3.6. Klasifikasi

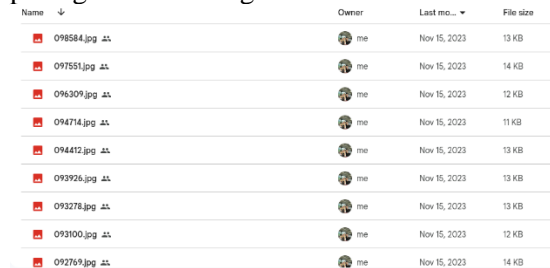
Klasifikasi merupakan sebuah proses untuk mengembangkan model atau fungsi yang menggambarkan konsep dari suatu data. Dalam proses klasifikasi, data-data tertentu dikumpulkan dan kemudian dikelompokkan ke

dalam kategori-kategori yang disebut sebagai kelas. Kelas yang terbentuk ini dikenal sebagai dataset pelatihan, yang akan digunakan untuk dilatih oleh algoritma klasifikasi. Proses ini dikenal sebagai pelatihan. [11], [12] Dengan pelatihan, akan dihasilkan sebuah model yang mampu melakukan klasifikasi pada data-data baru:

## 4. HASIL DAN PEMBAHASAN

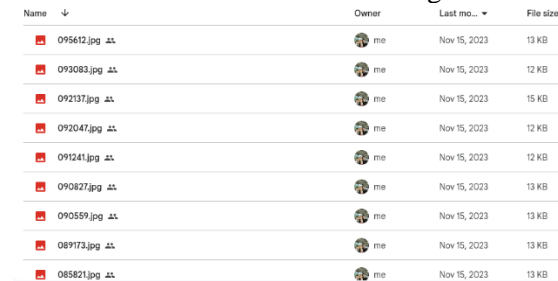
### 4.1. Pengumpulan Dataset

Dataset yang diterapkan dalam penelitian ini terdiri dari gambar-gambar yang merepresentasikan pose tubuh berdiri tegak dan tidak, dengan total keseluruhan mencapai 2936 citra. Langkah awal dalam penelitian ini melibatkan proses preprocessing, dimana citra-citra tersebut mengalami transformasi ukuran dengan menggunakan fungsi *resize*. Dalam langkah ini, seluruh citra diubah ukurannya menjadi 150x150 piksel, tujuannya adalah untuk meningkatkan efisiensi dalam proses data *training* dan untuk memperoleh hasil terbaik dari model yang dihitung. Setelah tahap *resize* citra selesai, dilakukan pembagian dataset, dimana terdiri dari dataset pose tegak dan dataset pose kurang tegak yang ditunjukkan pada gambar 3 dan gambar 4 di bawah ini.



Name	Owner	Last modified	File size
098584.jpg	me	Nov 15, 2023	13 KB
097551.jpg	me	Nov 15, 2023	14 KB
096309.jpg	me	Nov 15, 2023	12 KB
094714.jpg	me	Nov 15, 2023	11 KB
094412.jpg	me	Nov 15, 2023	13 KB
093926.jpg	me	Nov 15, 2023	13 KB
093278.jpg	me	Nov 15, 2023	13 KB
093100.jpg	me	Nov 15, 2023	12 KB
092769.jpg	me	Nov 15, 2023	14 KB

Gambar 3. Dataset Pose Tegak



Name	Owner	Last modified	File size
095612.jpg	me	Nov 15, 2023	13 KB
093083.jpg	me	Nov 15, 2023	12 KB
092137.jpg	me	Nov 15, 2023	15 KB
092047.jpg	me	Nov 15, 2023	12 KB
091241.jpg	me	Nov 15, 2023	12 KB
090827.jpg	me	Nov 15, 2023	13 KB
090559.jpg	me	Nov 15, 2023	13 KB
089173.jpg	me	Nov 15, 2023	13 KB
088821.jpg	me	Nov 15, 2023	13 KB

Gambar 4. Dataset Pose Kurang Tegak

### 4.2. Preprocessing Data

Pada tahap ini dilakukan import modul *os* yang menyediakan fungsi-fungsi untuk berinteraksi dengan sistem operasi, seperti

manipulasi path file, pembuatan direktori, dan lainnya. Lalu dilakukan mengimpor modul *shutil* yang menyediakan utilitas operasi file dan direktori yang lebih tinggi dari modul *os*. Contohnya termasuk fungsi untuk mengcopy, menghapus, dan menggeser file/direktori. Selanjutnya mengimpor fungsi *train\_test\_split* dari modul *model\_selection* di library *sklearn*. Fungsi ini digunakan untuk membagi dataset menjadi subset pelatihan dan pengujian. Ini sangat berguna dalam pengembangan model *machine learning* untuk menguji kinerja model pada data yang tidak dilihat selama pelatihan. Selengkapnya dapat pada gambar 5.

```
base_dir = '/content/gdrive/My Drive/datasetfoto'

import os
import shutil
from sklearn.model_selection import train_test_split

# Tentukan path ke direktori dataset
dataset_path = os.path.join(base_dir, 'dataset')

# Tentukan path untuk direktori train dan validation
train_path = os.path.join(base_dir, 'train')
val_path = os.path.join(base_dir, 'val')

# Tentukan perbandingan pembagian dataset (train: 70%, validation: 30%)
validation_split = 0.3

# List semua kelas dalam direktori dataset
classes = os.listdir(dataset_path)
```

Gambar 5. Preprocessing Data

### 4.3. Split Data

Pada tahap split data ini dilakukan proses pemisahan dataset yang akan dipindahkan ke dalam direktori train data untuk lebih jelasnya dapat dilihat pada gambar 6 di bawah ini



```

# Loop melalui setiap kelas
for class_name in classes:
    class_path = os.path.join(dataset_path, class_name)

    # List semua file dalam setiap kelas
    all_files = os.listdir(class_path)

    # Split setiap kelas menjadi train set dan validation set
    train_files, val_files = train_test_split(all_files, test_size=validation_split, random

    # Buat direktori train dan validation untuk setiap kelas jika belum ada
    os.makedirs(os.path.join(train_path, class_name), exist_ok=True)
    os.makedirs(os.path.join(val_path, class_name), exist_ok=True)

    # Pindahkan file ke dalam direktori train untuk setiap kelas
    for file in train_files:
        source = os.path.join(class_path, file)
        destination = os.path.join(train_path, class_name, file)
        shutil.copy(source, destination)

    # Pindahkan file ke dalam direktori validation untuk setiap kelas
    for file in val_files:
        source = os.path.join(class_path, file)
        destination = os.path.join(val_path, class_name, file)
        shutil.copy(source, destination)

print("Pembagian dataset selesai.")

```

Gambar 6. Split Data

#### 4.4. Training Model CNN

Pada tahapan ini dilakukan proses train data atau proses data latih dengan parameter target size 150 x 150 dan menggunakan parameter class\_mode binary untuk mengklasifikasikan dua buah kelas. Untuk detail lebih lanjut di tunjukan pada gambar 7 di bawah ini

```

Code + Text All changes saved

] train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    horizontal_flip=True,
    shear_range = 0.2,
    fill_mode = 'nearest')

test_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    horizontal_flip=True,
    shear_range = 0.2,
    fill_mode = 'nearest')

] train_generator = train_datagen.flow_from_directory(
    train_dir, # direktori data latih
    target_size=(150, 150), # mengubah resolusi seluruh gambar menjadi 150x150 piksel
    batch_size=8,
    # karena ini merupakan masalah klasifikasi 2 kelas maka menggunakan class_mode = 'binary'
    class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
    validation_dir, # direktori data validasi
    target_size=(150, 150), # mengubah resolusi seluruh gambar menjadi 150x150 piksel
    batch_size=8, # karena ini merupakan masalah klasifikasi 2 kelas maka menggunakan class_mode = 'binary'
    class_mode='binary')

Found 224 images belonging to 2 classes.
Found 96 images belonging to 2 classes.

] model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(512, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

] model.compile(loss='binary_crossentropy',
    optimizer='adam',

```

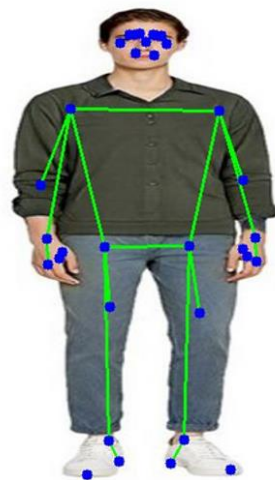
Gambar 7. Training Model CNN

#### 4.5. Hasil Identifikasi

Melalui implementasi metode Convolutional Neural Network (CNN) pada identifikasi pose tubuh, hasil penelitian menunjukkan bahwa dalam mengenali pose tubuh kurang tegak, model mencapai tingkat akurasi sebesar 59%, dengan tingkat kepercayaan (confidence) sebesar 58,54%. Ditunjukan pada gambar 8. Meskipun akurasi ini mencerminkan tingkat ketepatan model dalam mengklasifikasikan pose tubuh kurang tegak, kepercayaan yang relatif tinggi menunjukkan tingkat keyakinan model terhadap prediksinya. Sebaliknya, ketika mengidentifikasi pose tubuh tegak, model menunjukkan kinerja yang lebih baik dengan akurasi mencapai 80%, dan tingkat kepercayaan sebesar 79,36%. Ditunjukan pada gambar 9. Keakuratan yang lebih tinggi dalam mengenali pose tubuh tegak menunjukkan kemampuan model dalam mengklasifikasikan dengan tepat, sementara tingkat kepercayaan yang signifikan memberikan indikasi bahwa model memiliki

keyakinan yang kuat terhadap hasil prediksinya dalam konteks ini.

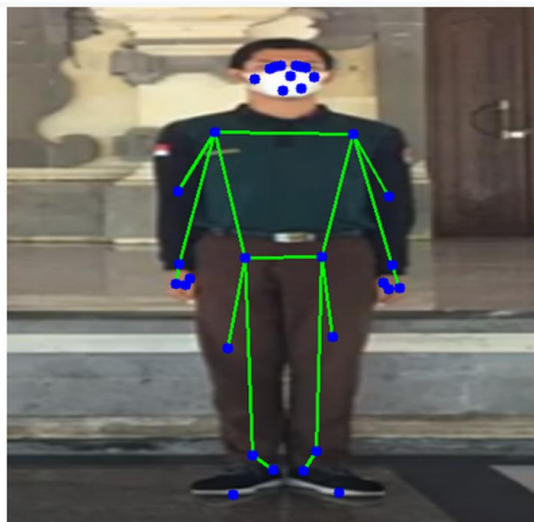
akurasi sebesar 80% dengan confidence 79,36%



**Kurang Tegak !**

Confidence: 58.54%  
Accuracy: 59.00%

Gambar 8. Identifikasi Pose Kurang Tegak



**Tegak**

Confidence/Tingkat Keyakinan: 79.36%  
Accuracy: 80.00%

Gambar 9. Identifikasi Pose Tegak

## 5. KESIMPULAN

Berdasarkan keseluruhan tahap yang sudah dilakukan. Maka hasil penelitian ini dapat mengidentifikasi pose tubuh kurang tegak dengan akurasi sebesar 59% dengan confidence 58,54% dan mengidentifikasi pose tegak dengan

## DAFTAR PUSTAKA

- [1] N. Khunafa Qudsi *et al.*, "Identifikasi Citra Tulisan Tangan Digital Menggunakan Convolutional Neural Network (CNN)," *Semin. Inform. Apl. Polinema*, pp. 48–53, 2019, Accessed: Nov. 27, 2023. [Online]. Available: <http://jurnalti.polinema.ac.id/index.php/SIAP/article/view/344>.
- [2] A. Arifandi, "Identifikasi dan Prediksi Umur serta Jenis Kelamin Berdasarkan Citra Wajah Menggunakan Algoritma Convolutional Neural Network (CNN)," *RAINSTEK J. Terap. Sains Teknol.*, vol. 4, no. 2, pp. 89–96, Jun. 2022, doi: 10.21067/JTST.V4I2.6985.
- [3] Y. Rizki, R. Hayami, and E. Rahmadani, "Identifikasi Objek Cagar Budaya Candi Mahligai Berbasis Citra Digital Menggunakan Mask R-CNN," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 3, pp. 309–314, Dec. 2022, doi: 10.37859/COSCITECH.V3I3.3939.
- [4] S. A. E. ALBAKIA and R. A. Saputra, "Identifikasi Jenis Daun Tanaman Obat Menggunakan Metode Convolutional Neural Network (CNN) Dengan Model VGG16," *J. Inform. Polinema*, vol. 9, no. 4, pp. 451–460, Aug. 2023, doi: 10.33795/JIP.V9I4.1420.
- [5] J. V. P. Putra, F. Ayu, and B. Julianto, "Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN," *Stain. (SEMINAR Nas. Teknol. SAINS)*, vol. 2, no. 1, pp. 155–162, Jan. 2023, Accessed: Nov. 29, 2023. [Online]. Available: <https://proceeding.unpkediri.ac.id/index.php/stains/article/view/2888>.
- [6] N. Hanun, M. Sarosa, and R. A. Asmara, "Pemanfaatan Algoritma Faster R-CNN ResNet-101 Untuk Deteksi Potongan Tubuh Manusia," *J. Elektron. dan Otomasi Ind.*, vol. 10, no. 1, pp. 94–103, May 2023, doi: 10.33795/ELKOLIND.V10I1.2754.
- [7] F. H. Hawari, F. Fadillah, M. R. Alviandi, and T. Arifin, "Klasifikasi Penyakit Tanaman Padi Menggunakan Algoritma

- Cnn (Convolutional Neural Network),” *J. Responsif Ris. Sains dan Inform.*, vol. 4, no. 2, pp. 184–189, Aug. 2022, doi: 10.51977/JTI.V4I2.856.
- [8] M. Rahman and A. Pambudi, “Implementasi Algoritma Convolutional Neural Network (Cnn) Untuk Identifikasi Kualitas Tanaman Selada Berdasarkan Citra Daun,” *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3s1, pp. 2830–7062, Sep. 2023, doi: 10.23960/JITET.V11I3S1.3368.
- [9] I. Jenis, R.-R. Menggunakan, M. Cnn, and B. Android, “Identifikasi Jenis Rempah-Rempah Menggunakan Metode CNN Berbasis Android,” *Jurasik (Jurnal Ris. Sist. Inf. dan Tek. Inform.*, vol. 8, no. 1, pp. 223–232, Feb. 2023, doi: 10.30645/JURASIK.V8I1.558.
- [10] S. Faisal, T. F. M Butarbutar, P. Sirait, and J. SIFO Mikroskil, “Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun,” *J. SIFO Mikroskil*, vol. 20, no. 2, pp. 117–134, Oct. 2019, doi: 10.55601/JSM.V20I2.670.
- [11] S. N. Ria, M. Walid, B. A. Umam, F. Teknik, and U. Islam, “Pengolahan Citra Digital Untuk Identifikasi Jenis Penyakit Kulit Menggunakan Metode Convolutional Neural Network (CNN),” *Energy J. Ilm. Ilmu-Ilmu Tek.*, vol. 12, no. 2, pp. 62–67, Dec. 2022, doi: 10.51747/ENERGY.V12I2.1118.
- [12] R. Pujiati and N. Rochmawati, “Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode Convolutional Neural Network (CNN),” *J. Informatics Comput. Sci.*, vol. 3, no. 03, pp. 351–357, Jan. 2022, doi: 10.26740/JINACS.V3N03.P351-357.