Student name: Hamidreza Aminorroayaei

Course : CSC-4444-AI ***semester*** : Spring 2024 ***Professor*** : Mahmood Jasim

Project: rock-paper-scissors game

Introduction

The rock-paper-scissors game, traditionally played between two people, offers a simple yet fascinating challenge for AI development, particularly in the field of computer vision and machine learning. This project utilizes a convolutional neural network (CNN) to enable a computer to play rock-paper-scissors against a human opponent by recognizing hand gestures through a webcam feed. The primary motivation is to demonstrate the application of image processing and neural networks in real-time interaction systems, which can be extended to various practical applications such as gesture-driven control in gaming, virtual reality, and accessibility tools.

Problem Statement

The task of automatically recognizing hand gestures from a live video feed involves challenges related to image processing, pattern recognition, and real-time interaction. The project aims to address these challenges by developing an AI that can accurately identify three distinct gestures (rock, paper, and scissors) and make decisions based on this input to compete against human players in real-time.

Technical Background

The solution employs deep learning techniques, specifically convolutional neural networks (CNNs), known for their effectiveness in image recognition tasks. TensorFlow and Keras frameworks are used to build and train the model due to their comprehensive libraries, ease of use, and support for rapid prototyping. OpenCV (Open Source Computer Vision Library) is utilized for real-time video capture and image preprocessing, crucial for preparing the data that the CNN will process.

Methodology:

Data Collection

- Dataset Description: The dataset used for training and testing the AI model in this rock-paper-scissors game consists of images categorized into three classes: rock, paper, and scissors. These images are essential for teaching the neural network to recognize different hand gestures corresponding to the moves in the game. The dataset is divided into two main folders: one for training and one for testing.

Structure:

- Training Data: The training folder contains a substantial set of images for each gesture. These images are used to train the convolutional neural network (CNN) model to recognize and distinguish between the three possible hand gestures.

  Rock: Images showing the hand gesture for 'rock' (a closed fist).

  Paper: Images displaying the hand gesture for 'paper' (an open hand).

  Scissors: Images of the hand gesture for 'scissors' (a fist with the index and middle fingers extended, forming a V).

- Testing set of images: These images are used to evaluate the performance of the trained model, ensuring that it accurately classifies new, unseen images.

  Rock

  Paper

  Scissors

Model Evaluation:

The model was rigorously tested with a separate set of images to ensure its performance and reliability. On the test data, the model achieved a high accuracy of 0.9451, demonstrating its effectiveness in recognizing hand gestures accurately and in real time. This high level of accuracy highlights the model's potential in real-world applications and sets a benchmark for further improvements.

Data Preparation and Augmentation

To enhance the model's ability to generalize from the training data to real-world scenarios, data augmentation techniques are applied during training. These include rotations, translations, zooming, and flipping of images. Such transformations simulate different orientations and variations in hand positioning, providing a robust set of training examples.

Model Architecture

- CNN Layers: The model architecture includes several convolutional layers, which are pivotal for extracting features from the images. These layers are followed by max-pooling layers, which reduce dimensionality and computational load. Dropout layers are interspersed to prevent overfitting.
- Activation Functions: ReLU (Rectified Linear Unit) is used for the convolutional layers to introduce non-linearity, and a softmax activation function is employed in the output layer to derive probabilities for the three classes.

Model Training and Evaluation

- Training Process: The model is trained using backpropagation and a stochastic gradient descent optimizer (Adam optimizer) to minimize the categorical cross-entropy loss function.
- Performance Metrics: Accuracy and loss metrics are collected both on training and validation datasets to monitor the model's performance and ensure it does not overfit.

System Design

- Integration with Real-Time Video Feed: Using OpenCV, the system captures video feed from a webcam. Each frame is processed in real-time to extract the region of interest (ROI), which is the hand region, preprocess it, and pass it to the trained model for gesture recognition.
- Decision Logic: The system interprets the model's output to make game decisions in real-time, displaying the results on the screen alongside the live video feed.

Challenges and Solutions

- Variability in Lighting and Background: Significant preprocessing and robust data augmentation are employed to ensure the model performs well regardless of lighting conditions and background variations in the live feed.
- Real-Time Processing Needs: Efficient image processing and model inference are critical, requiring optimization to run smoothly in real-time.

Future Work

- Enhanced Gesture Recognition: Expanding the model to recognize more complex gestures and incorporating a broader range of human-computer interaction scenarios.
- User Interface Improvements: Developing a more interactive and visually appealing interface for the game.
- Adaptive AI Opponent: Implementing machine learning algorithms that adapt the AI's playing strategy based on the opponent's past choices

Conclusion

This project showcases the practical application of convolutional neural networks in interpreting and responding to human gestures in real-time, paving the way for more sophisticated interactive applications. The successful implementation of the rock-paper-scissors game serves as a proof of concept for future advancements in gesture-based user interfaces.

Access Location of Game Files:

The complete game files and the associated AI model can be accessed through the GitHub repository. This repository provides all the necessary files required to run, modify, and evaluate the game.

GitHub Repository: *Rock-Paper-Scissors Game*