Video Action Recognition Using MediaPipe and LSTM

Introduction

In the exciting world of AI, Video Action Recognition is a key area where we blend complex tech to achieve real results. At the core of my project, I've combined two powerful tools: MediaPipe for pulling out key features from videos, and Long Short-Term Memory (LSTM) networks for understanding the sequence of actions over time. This mix is not just efficient – it's also more accurate than older methods.

Why MediaPipe is Great for Getting Features:

MediaPipe is a standout for grabbing important parts from video frames quickly and in detail. It’s like having a sharp eye that spots all the important movements in a video. This detailed view is essential for our system to know what's happening in each frame.

LSTM: The Smart Way to Understand Action Over Time:

LSTMs are very important for making sense of how actions unfold in a video. Unlike basic methods, LSTMs are great at remembering patterns over long videos. They catch the little details in how people move, which helps us recognize actions super accurately.

Putting MediaPipe and LSTMs together gets us the best of both worlds: excellent feature spotting and smart sequence understanding. This combo means my system doesn’t just work well – it's also one of the most accurate out there for figuring out what actions are happening in a video.

In this report, I will dive into how these technologies team up to make a cutting-edge system for recognizing actions in videos, really pushing the boundaries of what's possible in video processing.

Methodology

- **Data Exploration and Class Selection**:
  - My journey began with a deep dive into the UCF50 dataset. I explored and visualized the data, focusing on selecting four specific classes for my study. A key aspect of this stage was ensuring a balanced dataset. I carefully examined the number of samples in each class to prevent any bias that could arise from one class dominating over the others.
- **Frame Extraction from Videos**:
  - Next, I tackled the challenge of extracting frames from the videos. To achieve this efficiently, I employed a 'skip frames window' technique. This method allowed me to evenly sample frames across the entire length of each video, ensuring a representative and consistent set of frames for analysis.
- **Feature Extraction Using MediaPipe Holistic Model**:
  - After frame extraction, I utilized the MediaPipe Holistic model for feature extraction and pose detection. This powerful tool helped me capture key points from the frames, translating complex video data into a more analyzable form.
- **Dataset Creation and Preprocessing:**
  - The third phase involved creating the actual dataset for the model. I extracted features from the frames and paired them with corresponding one-hot encoded labels. To streamline the process for future experiments, I saved these features and labels in a .pkl file using pickle. This step was crucial for saving processing time for any subsequent methods applied to the same data.
- **Data Splitting:**
  - With the dataset ready, I divided it into training (65%), testing (15%), and validation (20%) sets. This split was designed to ensure a comprehensive training process while leaving more data for validation and testing, vital for assessing the model's performance.
- **Model Evaluation and Real-World Testing:**
  - Finally, I saved the best-performing model and tested it on new video data sourced from the internet. This was a crucial step to ensure that the model generalized well beyond the dataset it was trained on. Evaluating the model on external videos provided a realistic assessment of its practical applicability and robustness.

Data Collection

For this project, the dataset played a crucial role in training and validating the model. It comprised a total of 497 videos, a sizeable amount that contributed significantly to the model's ability to learn and make accurate predictions. This dataset was not just about quantity, the quality and diversity of the videos were key in ensuring robust model training.

The focus of the dataset was narrowed down to four distinct classes of physical activities: Jumping Jack, Pull Ups, PushUps, and JumpRope. By concentrating on these specific activities, This selective approach was instrumental in fine-tuning the model's ability to distinguish between different types of physical movements.

Also, a critical aspect of the data collection process was not only to train and validate the model effectively but also to ensure its capability to generalize to new, unseen data. To test this, I collected additional video data from the internet, which served as a new and independent dataset. This step was essential in evaluating the model's real-world applicability and its performance outside the confines of the original training dataset.

Implementation

- **Constructing the Model:**
    - The construction of the model was a crucial step in the project. This phase involved carefully piecing together the architecture that would be capable of accurately recognizing and classifying various physical activities in videos. Attention was paid to every detail, from the structure of the neural network layers to the specific parameters that would drive the model's learning process.

- **Role of Keras Tuner in Optimization:**
    - A key aspect of this phase was optimization, where Keras Tuner played an indispensable role. With Keras Tuner, I had the flexibility to experiment with a range of configurations. I tested different activation functions, numbers of hidden units, and learning rate settings. This experimentation was not just about enhancing performance but also

about understanding how different configurations impacted the model's learning and generalization capabilities.

- **Balancing Model Complexity and Efficiency:**
  - The result of this process was a model that embodied both efficiency and complexity, consisting of 201,924 parameters. This number represented a carefully considered balance with enough complexity to accurately learn and predict from the video data yet streamlined enough to maintain computational efficiency.
- **Training Parameters and Evaluation Criteria:**
  - I trained the model over 20 epochs, setting a limit of 5 max trials to refine and determine the best possible model configuration. Throughout the training process, validation loss was the primary metric for monitoring performance. This focus on validation loss was crucial for ensuring that the model was not just fitting well to the training data but also generalizing effectively to new, unseen data.

Results and Evaluation Metrics

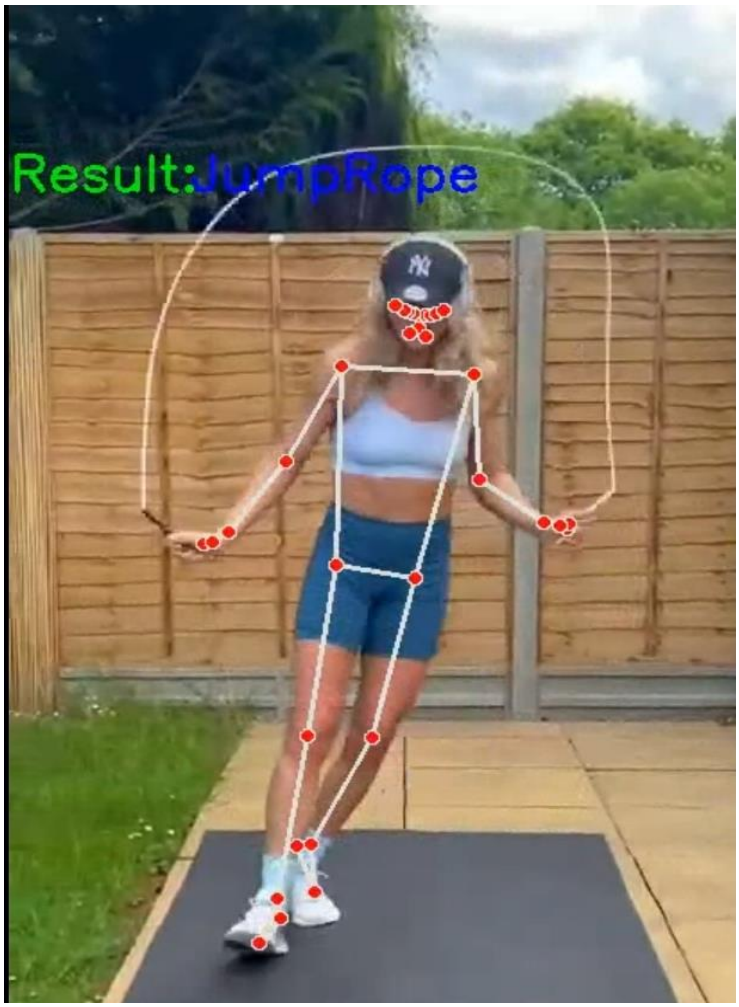
I am pleased to report that the model achieved an impressive 97.3% accuracy on the testing data. This high level of accuracy is indicative of the model's effectiveness in correctly identifying and classifying the various actions within the videos and equally important is the model's loss score, which came in at a remarkably low 0.0885 on the testing dataset. This low loss score signifies that the model's predictions were very close to the actual labels, further underscoring its reliability and precision.

- **Key Metrics: Classification Accuracy and Loss:**
  - Throughout the evaluation phase, the primary metrics used were classification accuracy and loss. These metrics provided a comprehensive understanding of the model's performance. Classification accuracy offered insights into how often the model correctly identified actions, while the loss score revealed the average error between the predicted and actual labels. The combination of these two metrics gave a well-rounded view of the model's overall effectiveness.

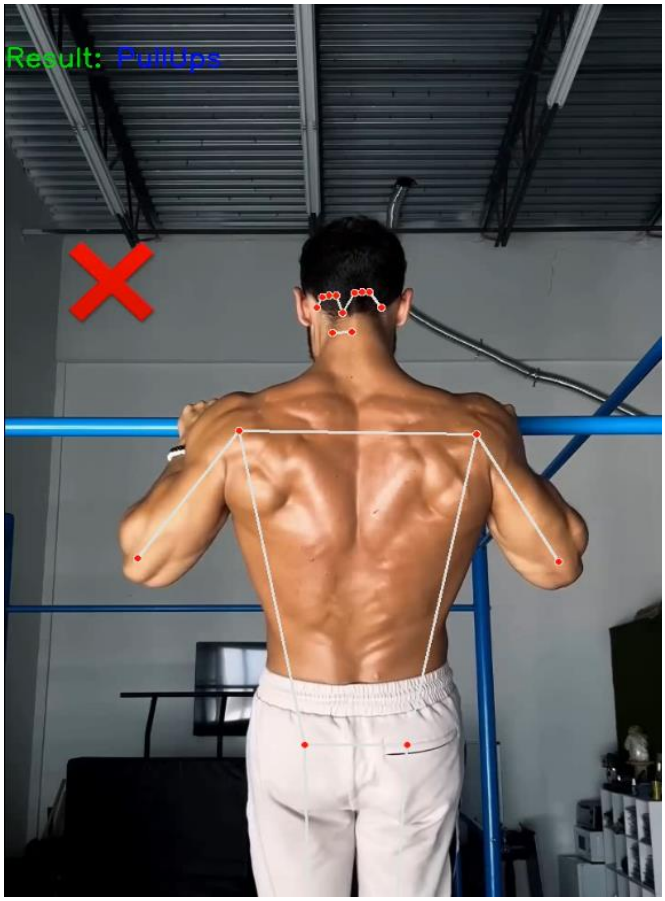
Demo

Now let’s look at the output of the model where we take frames from result videos:

1) Jumping Rope

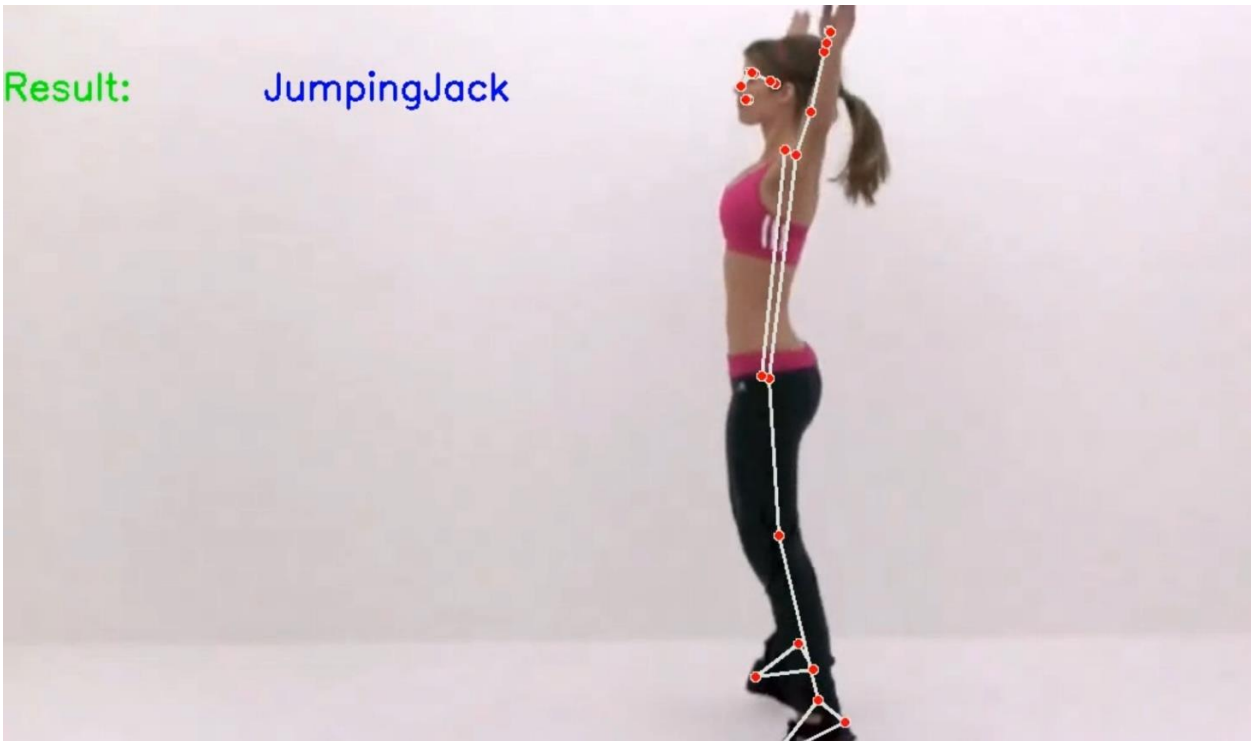


2) Pull Ups



3) Push Ups

4) **Jumping Jacks**



You can see the full videos from here.