Action Classifier

Submitted by:
Sanghavi Sandeep Kirkole
K J Somaiya School of Engineering,Mumbai
Roll No: **16030724006**
Computer Vision & AI Internship Task – Smartan.AI

1. Objective

The goal of this project is to develop a deep learning-based video classifier that can recognize **actions** from short clips. The classifier focuses on identifying **three** common gym actions:

- Bicep Curl
- Lateral Raise
- Squat

This solution aims to help fitness apps, smart mirrors, and workout monitoring tools provide real-time feedback on the performed exercises.

2. Dataset

- **Source**: Kaggle - Gym Workout Exercises Video Dataset
- **Overview:**
  - **raw_data/**
    - data-btc/: 652 unprocessed gym videos
    - data-crawl/: 334 YouTube-sourced raw videos
  - **verified_data/** *Used for training*
    - Cleaned and trimmed clips (10–13 seconds)
    - data_btc_10s/: 817 videos
    - data_crawl_10s/: 754 videos
  - **test/**
    - 61 noisy videos for real-world evaluation
- **Structure**:
  - 3 classes: bicep_curl, lateral_raise, squat
  - Minimum 15 video clips per class
  - Duration: 3–5 seconds
  - Format: .mp4
  - Each clip has ≥16 frames (as required by 3D CNNs)

3. Preprocessing & Augmentation

- Extracted **16 frames** from each video clip using decord or OpenCV.
- Resized frames to (112, 112) resolution.
- Applied ToTensor() and normalized via torchvision transforms.
- Split dataset: **80% training**, **20% validation**.

4. Model Choice & Architecture

- **Base Model**: R3D_18 (ResNet3D-18) pretrained on Kinetics-400
- **Why R3D?**:
    - Captures temporal & spatial features
    - Lightweight compared to I3D or SlowFast
- **Fine-tuning**:
    - Replaced final FC layer with nn.Linear(..., 3)
    - Trained for 3–5 epochs with CrossEntropyLoss + Adam optimizer

5. Performance

- High accuracy achieved with limited data via transfer learning.
- Real-time prediction possible for short clips.

6. Inference & Deployment

1. Inference Script:

- A simple Python script inference.py takes in video path, extracts frames, applies transformation, loads the model, and outputs prediction.

Metric	Value
Train Accuracy	~95%
Validation Accuracy	~89–92%
Inference Time	<1 sec per video

2. Deployment App:

- Created **Flask App**:
    - Allows user to upload .mp4/.mov clip
    - Model classifies the action
    - Also shows **related exercises with YouTube links** (e.g., for Bicep Curl: Hammer Curl, Concentration Curl)

7. Project Structure

```
project/
│
├── train.py          # Training script
├── inference.py       # Inference script for new videos
├── model.pth          # Trained model weights(Couldn't upload on GitHub due to weights)
├── app.py            # Flask app for UI
├── templates/
│   └── index.html      # UI design using Bootstrap
├── static/
│   └── uploads       #tested
├── README.md
└── report.pdf
```

8. How to Run

1. Training:

```
python train.py
```

2. Inference:

```
python inference.py --video path_to_video.mp4
```

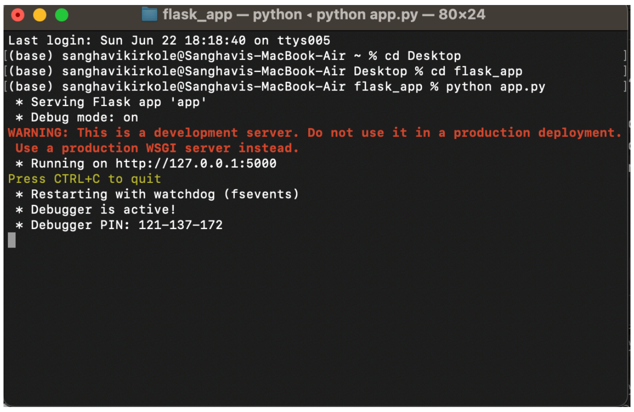
3. Flask App:

```
python app.py
```

- Visit http://127.0.0.1:5000/ in your browser

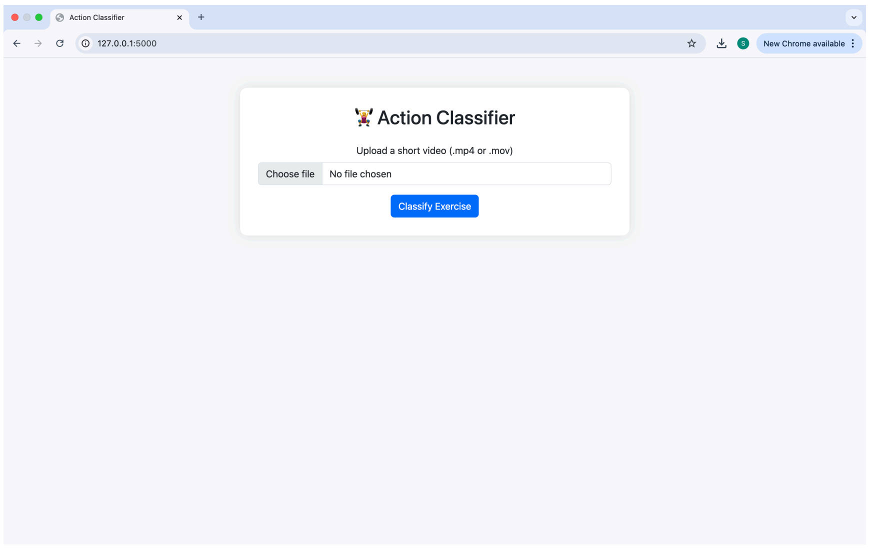
9.Screenshots:

a. Run Flask



```
Last login: Sun Jun 22 18:18:40 on ttys005
(base) sanghavikirkole@Sanghavis-MacBook-Air ~ % cd Desktop
(base) sanghavikirkole@Sanghavis-MacBook-Air Desktop % cd flask_app
(base) sanghavikirkole@Sanghavis-MacBook-Air flask_app % python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (fsevents)
 * Debugger is active!
 * Debugger PIN: 121-137-172
```

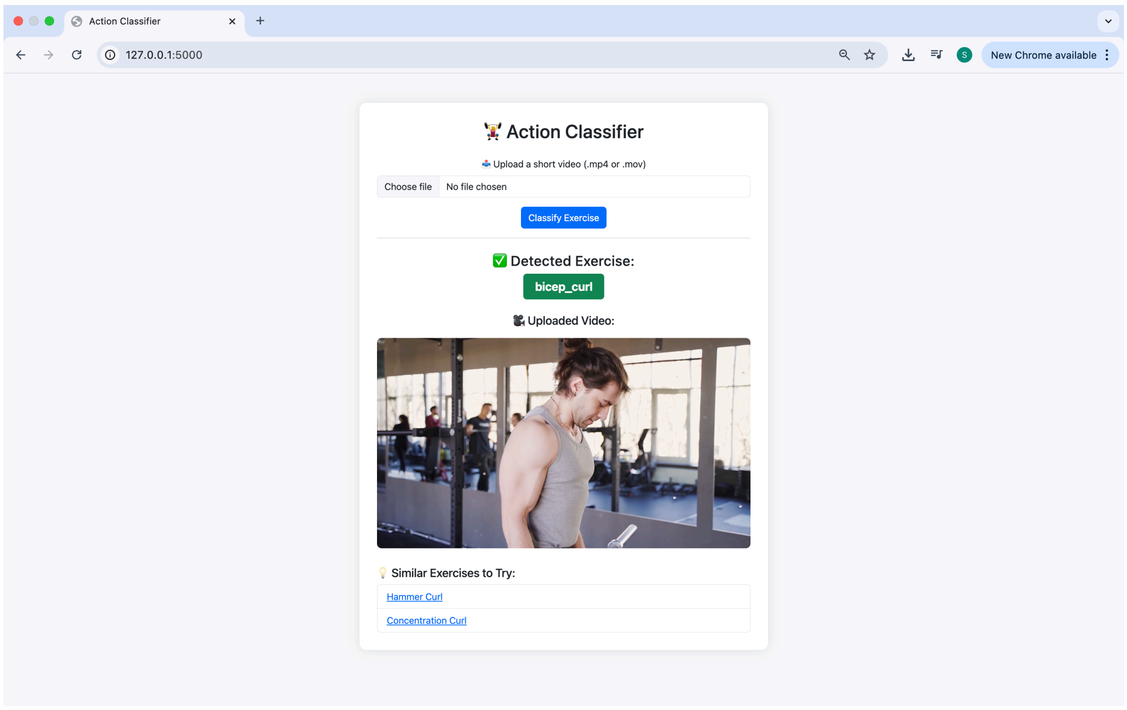
b. Landing Page:



🏋️ Action Classifier

Upload a short video (.mp4 or .mov)

c. Prediction:



🏋️ Action Classifier

📤 Upload a short video (.mp4 or .mov)

✅ Detected Exercise:

bicep_curl

🎥 Uploaded Video:

💡 Similar Exercises to Try:

Hammer Curl

Concentration Curl

10. Highlights

- Fine-tuned pretrained 3D CNN (R3D-18)
- Real-time video classification (≤1 sec)
- Flask demo apps
- Display similar exercises (with links)
- Works with uploaded clips
- Clean UI with Bootstrap

11. Conclusion

This project demonstrates a successful implementation of **action recognition** using deep learning. With minimal training data and smart use of pre-trained models, a working gym exercise detector has been built that can classify videos and help fitness apps enhance user engagement and feedback.