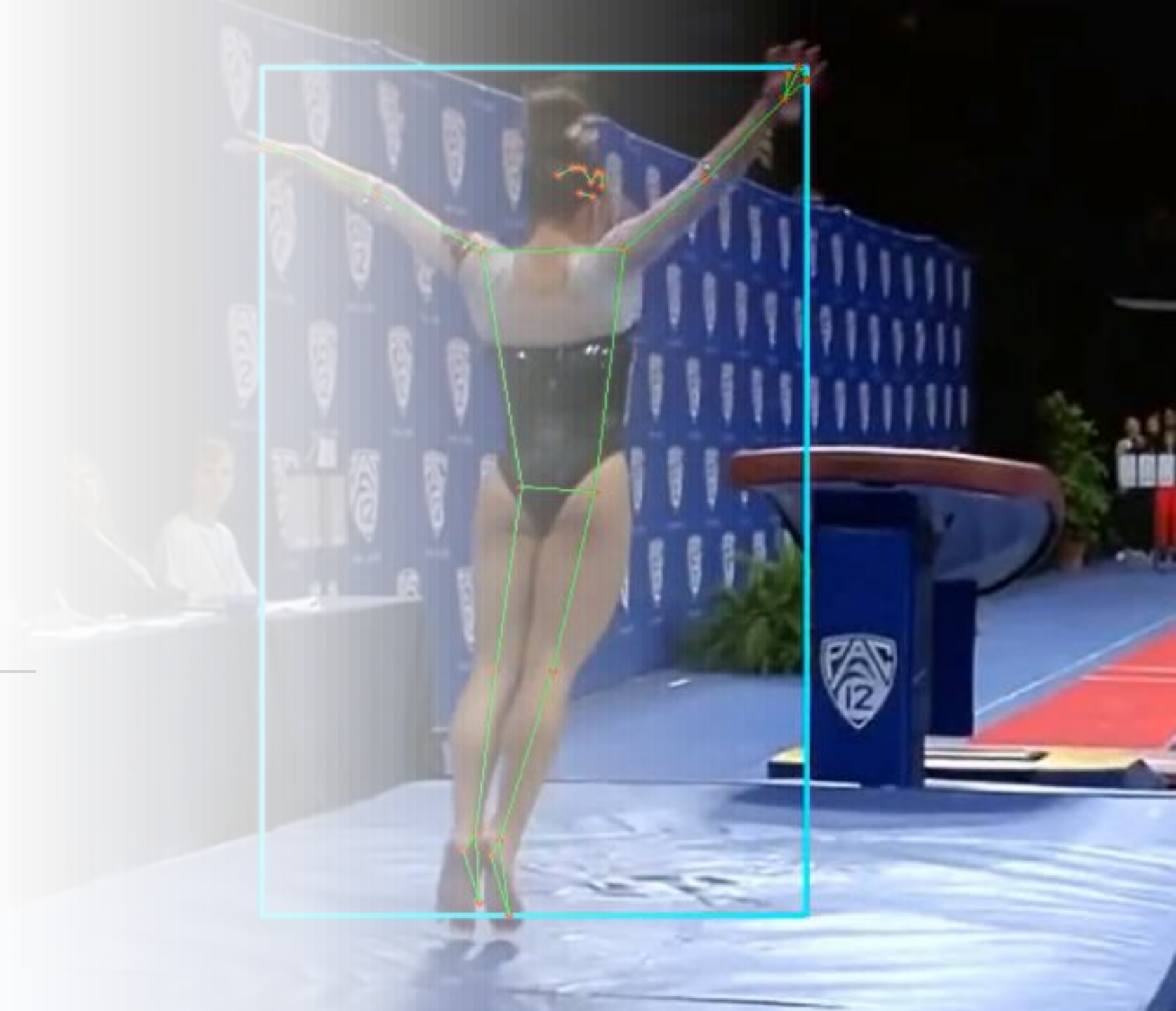


Анализ позы спортсмена с помощью MediaPipe

Автор: Крикунов Дмитрий
Геннадьевич



Задачи

- 1) Изучить входные данные - датасет неразмеченных видео.
- 2) Разработать пайплайн для поиска спортсмена и только спортсмена в видео с использованием модели MediaPipe.
- 3) Вычисление метрик стабильности, плавности движений и аномалий.
- 4) Сравнить результаты с моделями семейства YOLO.

Модель MediaPipe

Архитектура:

- Основана на **BlazePose** - легковесной CNN, оптимизированной для реального времени.
- Определяет **33 ключевые точки** тела (лицо, руки, туловище, ноги).

Выходные данные:

- 3D-координаты (x, y, z) и видимость каждой точки.
- Углы между суставами.
- Визуализация скелета (POSE_CONNECTIONS).

Преимущества модели

- **Высокая производительность:**
 - 30+ FPS на CPU
 - Минимальные требования к ресурсам
- **Точность в сложных условиях:**
 - Устойчивость к перекрытиям (self-occlusion)
 - Работа при частичной видимости тела
 - Коррекция дрожания точек (фильтр Калмана)

Архитектура решения

- 1) Загрузка данных (видеоархив).
- 2) Установка зависимостей (MediaPipe).
- 3) Инициализация модели позы.
- 4) Обработка видео и трекинг ключевых точек.
- 5) Расчет метрик и визуализация результатов.

```
def init_detector():
    mp_pose = mp.solutions.pose
    pose = mp_pose.Pose(static_image_mode=False,
                        model_complexity=1,
                        enable_segmentation=False,
                        min_detection_confidence=0.5,
                        min_tracking_confidence=0.5)

    return pose, mp_pose
```

```
kp_array = np.zeros((33, 2))
if results.pose_landmarks:
    h, w, _ = frame.shape
    for idx, lm in enumerate(results.pose_landmarks.landmark):
        kp_array[idx] = [lm.x * w, lm.y * h]
    all_keypoints.append(kp_array)

    # Визуализация скелета и соединений
    for x, y in kp_array:
        cv2.circle(frame, (int(x), int(y)), 2, (0,0,255), -1)
    for start_idx, end_idx in mp_pose.POSE_CONNECTIONS:
        x_start, y_start = kp_array[start_idx]
        x_end, y_end = kp_array[end_idx]
        cv2.line(frame,
                 (int(x_start), int(y_start)),
                 (int(x_end), int(y_end)),
                 (0,255,0), 1)
else:
    all_keypoints.append(np.zeros((33, 2)))

out.write(frame)
```

```
# Собираем ключевые точки
kp_array = np.zeros((33, 2))
if results.pose_landmarks:
    h, w, _ = frame.shape
    for idx, lm in enumerate(results.pose_landmarks.landmark):
        kp_array[idx] = [lm.x * w, lm.y * h]
    all_keypoints.append(kp_array)

    # Визуализация bounding box
    x_coords = kp_array[:, 0]
    y_coords = kp_array[:, 1]
    x1, y1 = int(np.min(x_coords)), int(np.min(y_coords))
    x2, y2 = int(np.max(x_coords)), int(np.max(y_coords))
    cv2.rectangle(frame, (x1, y1), (x2, y2), (255,255,0), 2)
```

Используемые метрики

- 1) Pose stability: средняя дисперсия координат ключевых точек по времени — показывает стабильность трекинга.
- 2) Smoothness: среднее ускорение ключевых точек — характеризует плавность движения.
- 3) Avg inf time: среднее время инференса (мс) — производительность модели.
- 4) Anomalous frames: количество кадров со скачками позы выше порога ($\text{mean} + 2 \cdot \text{std}$) — указывает на возможные артефакты.

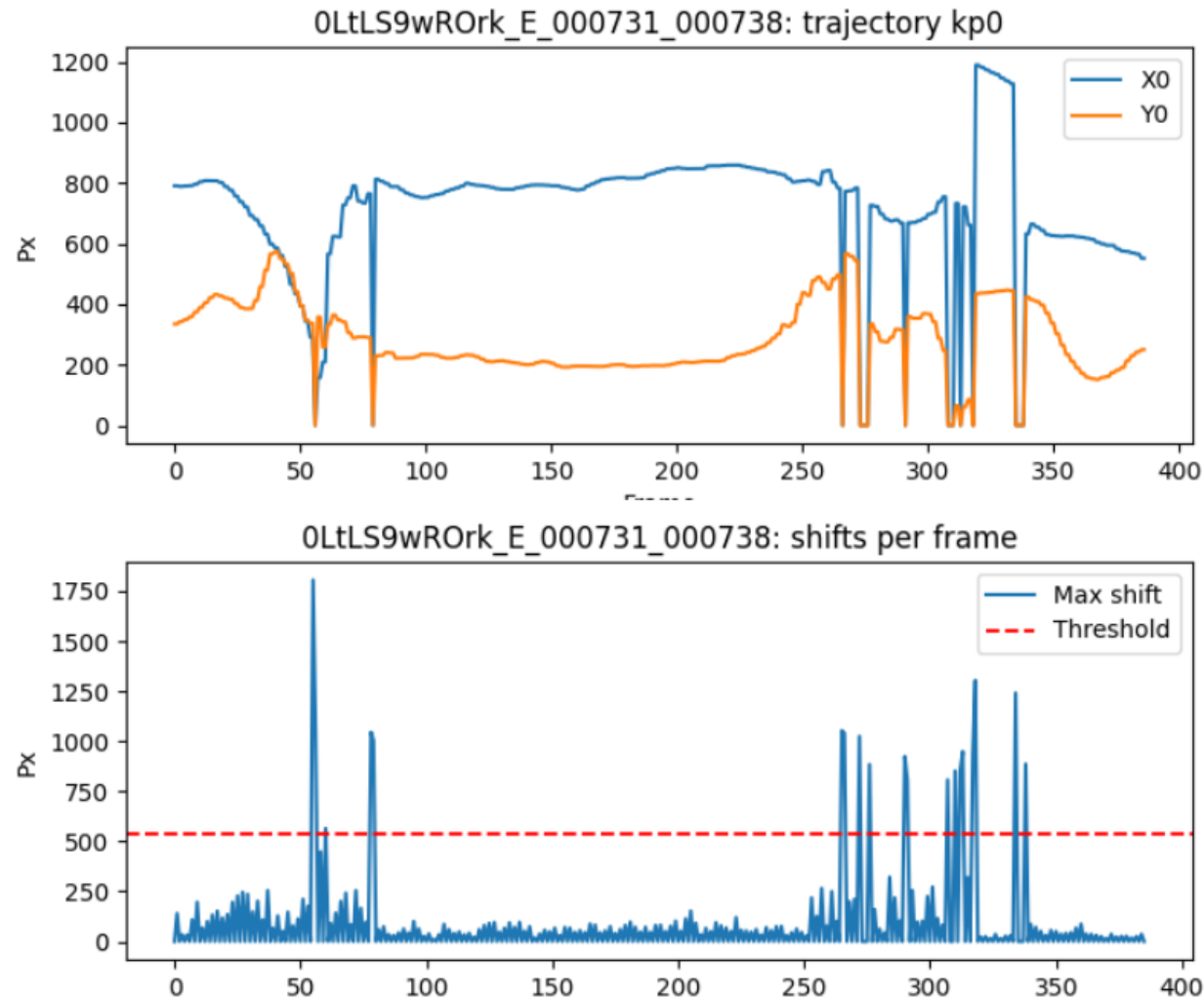
Результаты

```
pose_module, mp_pose = init_detector()
os.makedirs(output_dir, exist_ok=True)
results = []
for fname in os.listdir(input_dir):
    if not fname.lower().endswith('.mp4'):
        continue
    print(f'Processing {fname}...')
    res = process_video(pose_module, mp_pose,
                        os.path.join(input_dir, fname),
                        output_dir)
    results.append(res)

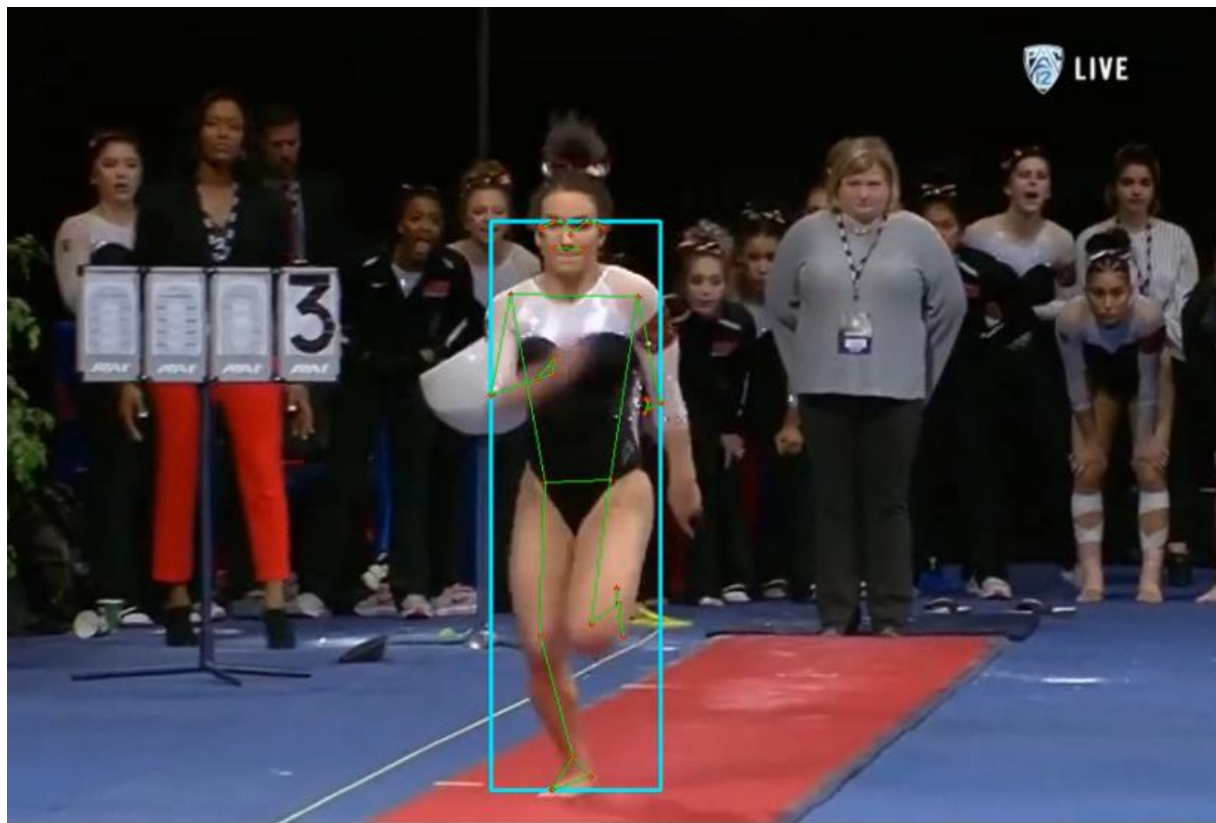
table = [[r['video'],
          f"{r['pose_stability']:.2f}",
          f"{r['smoothness']:.2f}",
          f"{r['avg_inf_time']:.1f}",
          r['num_anomalies']]
         for r in results]
print(tabulate(table, headers=['Video', 'Stability', 'Smooth', 'Inf ms', 'Anomalies'])
```

Pose stability	122832
Smoothness	107.97
Avg inf time (ms)	31.1
Anomalous frames	19
Processing 0LtLS9wR0rk_E_000519_000594.mp4...	
Metric	Value
-----	-----
Pose stability	37241.5
Smoothness	20.95
Avg inf time (ms)	30.1
Anomalous frames	66
Processing 0LtLS9wR0rk_E_000322_000417.mp4...	
Metric	Value
-----	-----
Pose stability	60625.6
Smoothness	38.51
Avg inf time (ms)	30.8

Визуализация метрик для одного из видео



Кадры из результирующего видео



Сравнение с моделями YOLO

MediaPipe

Processing 0LtLS9wR0rk_E_000731_000738.mp4...				
Metric	Value			
-----	-----			
Pose stability	122832			
Smoothness	107.97			
Avg inf time (ms)	40.8			
Anomalous frames	19			
Video	Stability	Smooth	Inf ms	Anoms
-----	-----	-----	-----	-----
0LtLS9wR0rk_E_000731_000738	122832	107.97	40.8	19

YOLO

Comparative Results:					
+-----+-----+-----+-----+-----+					
Video	Model	Stability	Smooth	Inf ms	Anoms
+-----+-----+-----+-----+-----+					
0LtLS9wR0rk_E_000731_000738	yolov8n-pose	138370	233.9	168.2	17
+-----+-----+-----+-----+-----+					
0LtLS9wR0rk_E_000731_000738	yolov8s-pose	109765	219.92	449.4	18
+-----+-----+-----+-----+-----+					
0LtLS9wR0rk_E_000731_000738	yolov8m-pose	142934	225.79	1056.8	22
+-----+-----+-----+-----+-----+					

