

Informe entrega 2

1. Estrategia implementada para la obtención de nuevos datos

Para la obtención de datos, implementamos dos estrategias complementarias:

1.1. Generación de videos con participantes reales

- Grabamos videos de varios participantes (familiares y miembros del equipo) realizando las actividades específicas requeridas: caminar hacia la cámara, caminar de regreso, girar, sentarse y levantarse.
- Los videos fueron capturados en diferentes condiciones de iluminación y desde diferentes ángulos para aumentar la variabilidad.
- Se instruyó a los participantes a realizar las actividades de forma natural, sin movimientos excesivamente exagerados.

1.2. Técnica de ampliación de datos

- Para complementar los datos recopilados, implementamos un algoritmo de ampliación que duplicó efectivamente nuestro conjunto de datos (de 12,705 a 25,410 muestras).
- Esta técnica consiste en crear nuevas muestras sintéticas a partir de los datos originales, añadiendo pequeñas variaciones aleatorias (nivel de ruido: 0.005) a las coordenadas de los puntos clave del cuerpo (landmarks).
- La idea central es introducir variabilidad controlada en los datos de entrenamiento. Aplicamos un ruido gaussiano pequeño a las coordenadas x e y de los landmarks, lo que simula variaciones sutiles en la posición de las articulaciones.
- Al entrenar con estas variaciones, el modelo aprende a enfocarse en las características esenciales de cada actividad y se vuelve más tolerante a pequeñas diferencias en la ejecución de los movimientos.
- Esta técnica es una práctica estándar en aprendizaje automático para mejorar la generalización y evitar el sobreajuste, especialmente cuando el conjunto de

datos original es limitado.

- Como resultado, obtuvimos un modelo más robusto que puede clasificar correctamente las actividades incluso cuando hay pequeñas variaciones en la forma en que se realizan, lo que es crucial para un sistema que debe funcionar con diferentes usuarios.

2. Preparación de los datos

El proceso de preparación de datos consistió en varias etapas:

2.1. Extracción de landmarks con MediaPipe

- Los videos capturados fueron procesados frame por frame utilizando MediaPipe Pose.
- Para cada frame, se extrajeron 33 landmarks corporales, cada uno con coordenadas x, y, z y valor de visibilidad.
- Estos datos fueron convertidos a formato CSV, organizados por actividad y participante.

2.2. Normalización de coordenadas

- Implementamos la normalización de todas las coordenadas respecto a la posición de la cadera (landmark 0).
- Este paso es crucial para hacer que el modelo sea invariante a la posición de la persona en el encuadre de la cámara.
- La normalización permite que el sistema funcione correctamente independientemente de la distancia o posición lateral de la persona respecto a la cámara.

2.3. Filtrado de datos

- Eliminamos las columnas de visibilidad ('vis') y número de frame ('frame') por no ser relevantes para la clasificación.
- Aplicamos un muestreo para reducir la redundancia temporal (procesando 1 de cada 3 frames).

3. Entrenamiento de los modelos y ajuste de hiperparámetros

3.1. División de datos

- Separamos los datos en conjuntos de entrenamiento (80%) y prueba (20%) para una evaluación rigurosa.
- La división se realizó a nivel de archivos CSV (videos) para evitar contaminación entre conjuntos.

3.2. Búsqueda de hiperparámetros óptimos

- Implementamos GridSearchCV para explorar sistemáticamente 54 combinaciones de hiperparámetros.
- Parámetros evaluados:
 - n_estimators: [100, 200]
 - max_depth: [10, 15, 20]
 - min_samples_leaf: [2, 5, 10]
 - min_samples_split: [2, 5, 10]
- Usamos validación cruzada de 5 pliegues para una evaluación robusta.
- Métrica de optimización: F1-score ponderado.

3.3. Modelo seleccionado

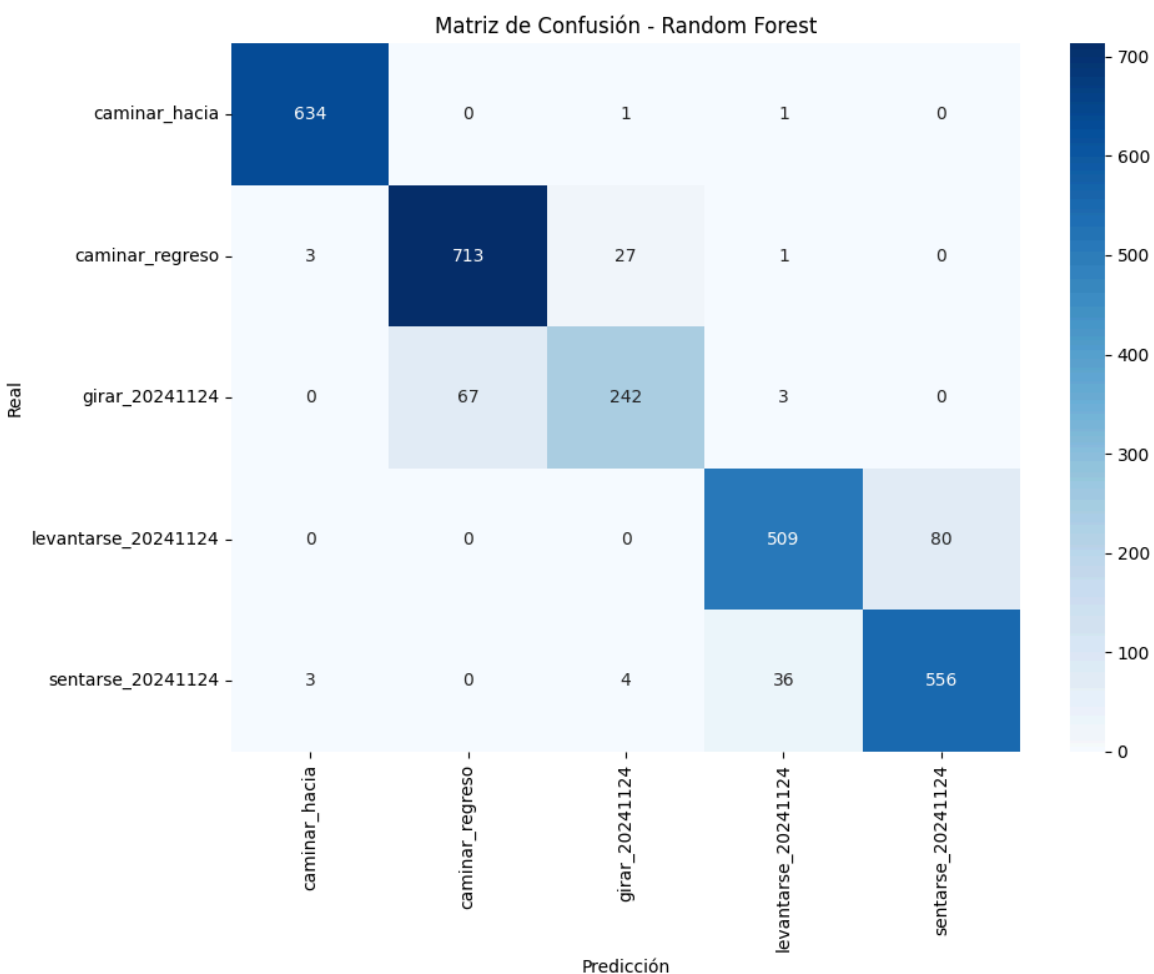
- Los mejores hiperparámetros encontrados fueron:
 - max_depth: 20
 - min_samples_leaf: 2
 - min_samples_split: 2
 - n_estimators: 200
- Estos parámetros muestran un buen equilibrio entre complejidad del modelo y capacidad predictiva.

4. Resultados obtenidos

4.1. Métricas de rendimiento

- **Accuracy:** 92.15% - Alto porcentaje de clasificaciones correctas.
- **Precision:** 92.17% - Excelente precisión en las predicciones positivas.
- **Recall:** 92.15% - Gran capacidad para identificar todas las instancias positivas.
- **F1 Score:** 92.07% - Buen equilibrio entre precisión y exhaustividad.

4.2. Análisis de la matriz de confusión



- Las actividades "caminar_hacia" y "caminar_regreso" se clasifican con alta precisión (634/636 y 713/744 respectivamente).

- Existe cierta confusión entre "girar" y "caminar_regreso" (67 casos), lo cual es comprensible dado que ambas actividades comparten movimientos de rotación.
- También hay confusión entre "levantarse" y "sentarse" (80 y 36 casos respectivamente), lo cual es lógico dado que son acciones inversas con posturas intermedias similares.

5. Plan de despliegue

Para el despliegue de nuestra solución, hemos decidido implementar un enfoque basado en ejecutable:

5.1. Conversión a aplicación ejecutable

- Utilizaremos PyInstaller para convertir nuestro script `motion_system.py` en un archivo ejecutable (.exe en Windows).
- Comando: `pyinstaller --onefile --windowed motion_system.py`
- Esta solución encapsula todas las dependencias necesarias (OpenCV, MediaPipe, scikit-learn) en un único archivo.

5.2. Requisitos del sistema

- Sistema operativo: Windows 10/11
- Webcam o cámara integrada
- Procesador: Intel Core i5 o equivalente (o superior)
- RAM: 4GB mínimo (8GB recomendado)
- No se requiere instalación de Python ni bibliotecas adicionales.

5.3. Distribución

- El ejecutable se distribuirá junto con los archivos del modelo entrenado.
- Los usuarios finales solo necesitarán ejecutar el archivo .exe para iniciar el sistema.

6. Análisis inicial de los impactos de la solución

6.1. Impacto académico

- El sistema sirve como herramienta educativa para estudiantes de visión por computadora e IA.
- Demuestra la aplicación práctica de algoritmos de aprendizaje automático y procesamiento de video.
- Los resultados pueden servir como base para investigaciones más avanzadas en análisis de movimiento humano.

6.2. Impacto en análisis de movimiento y biometría

- El sistema ofrece una alternativa accesible para la detección y análisis de actividades físicas.
- Potenciales aplicaciones incluyen:
 - Rehabilitación física: Monitoreo de ejercicios terapéuticos
 - Entrenamiento deportivo: Análisis de técnicas y posturas
 - Estudios ergonómicos: Evaluación de movimientos en entornos laborales

6.3. Impacto tecnológico

- Demuestra que es posible lograr alta precisión (>92%) en clasificación de actividades utilizando hardware estándar (webcam).
- La combinación de MediaPipe (para extracción de landmarks) y Random Forest (para clasificación) ofrece un equilibrio óptimo entre precisión y eficiencia computacional.
- Las técnicas de normalización y suavizado implementadas hacen el sistema más robusto y usable en condiciones reales.