

Universidad de Alicante

MÁSTER EN INTELIGENCIA ARTIFICIAL. ENTORNOS INTELIGENTES

GAIT ANALYSIS

Estimation of self-reported psychological traits from gait.

Autor:

Iñigo Zárate Rico

Mayo 2023

Índice general

1. Introducción	2
2. Metodología	3
2.1. Dataset, Skeletons	3
2.2. Normalización y preparación de los datos	4
2.2.1. Normalización dimensional	4
2.2.2. Normalización de esqueletos	5
2.2.3. Codificación de etiquetas	7
2.2.4. Preparación de los datos	7
2.3. Creación y entrenamiento del modelo	8
3. Resultados	10

Apartado 1

Introducción

El análisis de patrones al andar es un campo en el que la inteligencia artificial se convierte en una poderosa herramienta permitiendo analizar y comprender grandes conjuntos de datos y crear relaciones temporales entre las instancias del movimiento.

En este proyecto se pretende obtener un modelo clasificador capaz de evaluar la forma de caminar de una persona en un conjunto de rasgos psicológicos. Una herramienta así puede ser aplicada en multitud de campos, siendo principalmente, su uso en entornos inteligentes para la asistencia de personas mayores o con déficits de autosuficiencia.

El código desarrollado para este proyecto se puede encontrar en el siguiente repositorio de GitHub https://github.com/izarte/gait_emotion_recognition

Apartado 2

Metodología

En este apartado se va a explicar el proceso o pipeline que se ha desarrollado para este proyecto.

En primer lugar, el dataset utilizado se encuentra basado en Cosma and Radoi (2024). Debido a limitaciones técnicas sobre el hardware utilizado para el proyecto, se ha utilizado un subdataset del mencionado, constando de 10 ejemplos de diferentes personas realizando una trayectoria con 3 diferentes vistas y 2 direcciones por cada trayectoria.

De este modo, con 10 muestras, 3 cámaras y 2 direcciones se obtiene un total de 60 datos distintos. Evidentemente, no es una muestra lo suficientemente cuantiosa para obtener un modelo competente. Sin embargo, el proyecto se encuentra centrado en el proceso de preparación de los datos, organización y creación del modelo. Cabe destacar que, todas las métricas expuestas hacen referencia a este subset de datos.

2.1. Dataset, Skeletons

Para comenzar, el primer paso es entender la organización de los datos. El dataset proporciona varios formatos para relacionar el movimiento de una persona con su análisis emocional.

Para este proyecto se ha seleccionado el formato de esqueletos obtenidos mediante AlphaPose (Fang et al. (2022)). Se ha seleccionado este formato ya que, de este modo, todos los datos que recibirá el modelo son relevantes ya que, cada uno de los datos representa un punto relevante para conocer la posición y pose de una persona en un instante de tiempo.

Se concluyó este formato debido a que, las otras opciones, involucran imágenes con gran parte del contenido sin información relevante para el análisis, como puede ser el propio fondo de la escena o las dimensiones de la persona en cada caso.

2.2. Normalización y preparación de los datos

2.2.1. Normalización dimensional

Tal y como se ha comentado previamente, para cada sujeto, se encuentran 6 sucesiones de instancias de tiempo (por los diferentes ángulos y direcciones) con una lista de puntos creando el esqueleto para cada una. En cada experimento de un sujeto y un ángulo, se encuentra un número variable de instancias.

En la figura 2.1 se muestra un gráfico mostrando, por un lado, el histograma de longitudes para cada experimento, y, por otro lado, la distribución que representan dichas longitudes

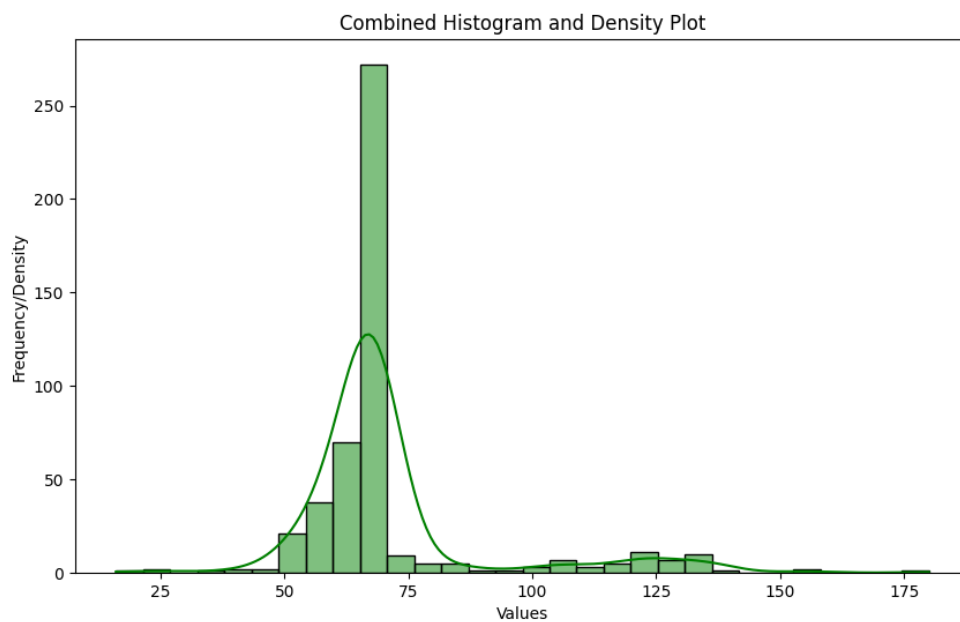


Figura 2.1: Longitud de los "videos" en cada experimento

Como se puede observar, los datos representan una distribución normal definida por sus paráme-

tros como: $\mu = 71,36, \sigma = 19,76$.

Gracias a esta distribución, se puede asumir que los experimentos con una cantidad de datos inferior al 25 % del conjunto no presentan suficiente información (relativa al resto de datos) por lo que son descartados.

Por otro lado, se observa una concentración de la mayoría de experimentos en un reducido rango de longitudes, con mayores longitudes con menor peso relativo. Para solventar la diferencia y normalizar la longitud de cada conjunto de datos (cada experimento en un ángulo), se ha decidido reducir todas las longitudes a una sola. Como valor estándar se ha seleccionado el mínimo valor que representa una carga suficiente relativa para ser considerado. Este valor ha sido decidido basándose en el valor del percentil 25, en este caso 65.

Por tanto, se ha creado una lógica para redimensionar cada experimento. Si la longitud del experimento supera el valor de 50, se realiza un *downsample* eliminando valores internos de la sucesión con la mayor distancia posible entre cada dato eliminado. De este modo, se reduce la dimensionalidad del experimento, manteniendo instantes relevantes del mismo reduciendo lo máximo posible los saltos de información en el conjunto total del experimento.

Otra aproximación posible sería reducir los valores superiores a la media y aumentar los valores inferiores (mayores o no a un valor mínimo) mediante interpolación de los puntos entre instancias. En la imagen 2.2 se muestra un histograma de las longitudes de los experimentos tras la normalización de la misma, de este modod se obtiene una distribución definida por: $\mu = 65, \sigma = 0$.

2.2.2. Normalización de esqueletos

En primer lugar, se debe leer la lista de puntos que define el esqueleto y convertirla en un esqueleto. Cada esqueleto se representa como uno tipo de dato de la librería **torch** de forma *Data*. En esta estructura de datos se almacenan 2 matrices, una con las posiciones XY de los puntos del esqueleto y en la otra las aristas que unen dichos puntos para generar el esqueleto.

La matriz de aristas define las conexiones entre los diferentes puntos de la lista de puntos. Como los datos han sido calculados utilizando la red AlphaPose, y, este modelo predice los puntos con una lista de aristas constante, se pueden obtener las relaciones entre puntos de este modo.

En la figura 2.3 se muestra la representación del esqueleto generado con puntos en 3 dimensiones, por otro lado, en la figura 2.4 se muestra la proyección de dichos puntos en 2 dimensiones.

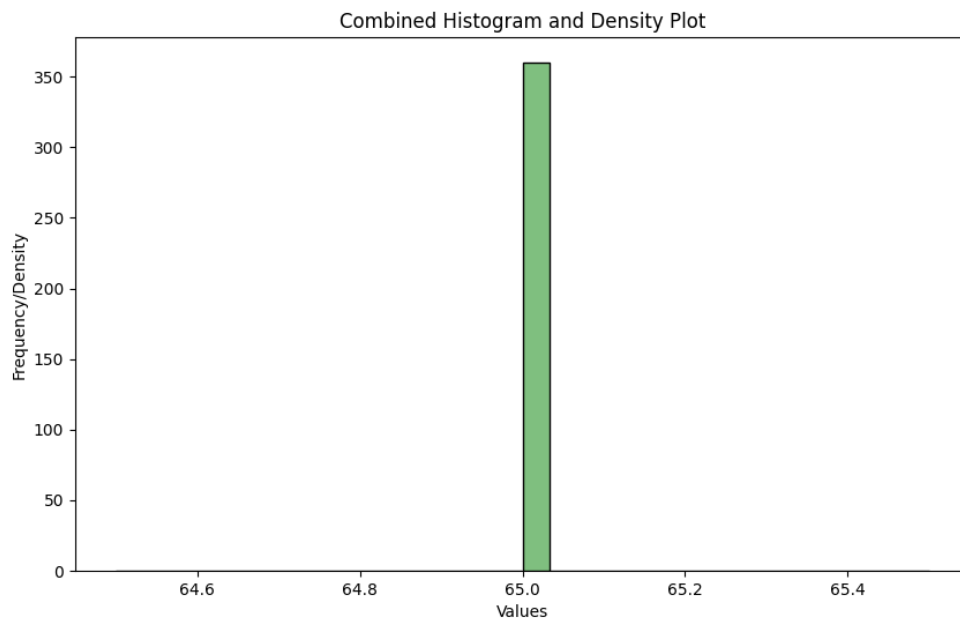


Figura 2.2: Histograma de longitudes normalizadas

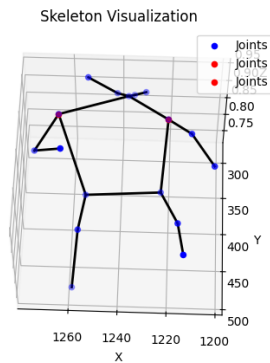


Figura 2.3: Esqueleto 3D

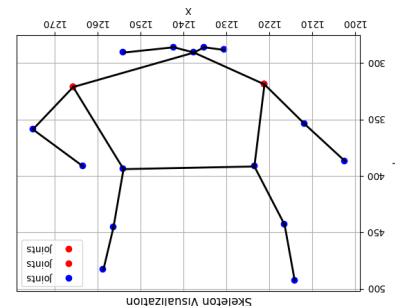


Figura 2.4: Proyección a esqueleto 2D

Una vez creado el esqueleto y confirmado visualmente la correcta disposición de las aristas, se deben normalizar los esqueletos.

Para abordar esta tarea, en primer lugar, se centra el esqueleto, para ello, se calcula la distancia del origen de coordenadas con un punto de referencia (debe ser constante a todos los esqueletos), en

este caso, el punto de referencia es el 0 que representa la posición de la cabeza. Se resta la distancia calculada a todos los puntos de modo que, todos los esqueletos se encuentran centrados en la cabeza.

Tras el centrado del esqueleto se procede a la normalización de sus dimensiones. Para ello, se toman 2 puntos de referencia (también constantes a todos los esqueletos), en este caso se han seleccionado los puntos que representan los hombros (puntos rojos en la Figura 2.4). Se calcula la distancia entre dichos puntos y se obtiene el factor de escalado que genera dicha distancia a 1. Se multiplican todos los puntos por dicho factor de modo que se asegura la longitud de 1 entre los hombros, manteniendo proporcional la distancia relativa con el resto de puntos creando un esqueleto escalado y constante para todos los esqueletos.

2.2.3. Codificación de etiquetas

En el dataset, se presenta un análisis emocional para cada sujeto. Cada uno fue evaluado en un test con 17 categorías distintas, para cada una se asigna un estado o clase en referente a los resultados del test.

Para poder procesar estos datos, se crea un mapeo que establece relaciones entre el *string* dado como clase y un valor numérico que lo representa. Este proceso se podría realizar de forma automática, sin embargo, se ha decidido crear este mapeo manual para obtener una relación numérica entre las clases significativa. Por ejemplo, la clasificación "Low" se representa con un 0, la clasificación "Normal / Low" con un 1, "Normal" con un 2 y así sucesivamente.

Con esta codificación se obtienen valores numéricos para cada clase ordenados ascendentemente según su significado real.

2.2.4. Preparación de los datos

Con el objetivo de crear una estructura de datos para alimentar secuencialmente a la red con los datos correspondientes durante su entrenamiento, se ha decidido crear un **dataloader** (estructura de la librería torch para cumplir esta función) propio personalizado.

Este **dataloader**, almacena en una lista las secuencias de matrices de cada experimento, de modo que cada elemento de la lista consta de otra lista con las matrices (en formato *Data*) de cada instancia del experimento, por tanto, cada elemento de la lista almacenada representa la sucesión de movimientos de un experimento con un ángulo concreto.


```
SkeletonGNN(
  (conv1): GraphConv(2, 64)
  (transformer_conv): TransformerConv(64, 64, heads=4)
  (lin1): Linear(in_features=256, out_features=64, bias=True)
  (fc): Linear(in_features=64, out_features=17, bias=True)
  (layers): ModuleList(
    (0-1): 2 x GraphConv(64, 64)
  )
  (skip): Linear(in_features=64, out_features=256, bias=False)
)
```

Figura 2.5: Capas del modelo propuesto

Cada experimento se identifica mediante un valor numérico o **id** para relacionar la sucesión de esqueletos con su etiqueta correspondiente, uno por cada sujeto.

Por otro lado, se lee el archivo que contiene las etiquetas de cada sujeto, se eliminan las columnas irrelevantes, como puede ser el género o la edad y se almacena.

De este modo, cada vez que se tenga que alimentar con datos en el entrenamiento, se obtiene un conjunto de experimentos organizados con sus correspondientes etiquetas.

2.3. Creación y entrenamiento del modelo

Para realizar la tarea de clasificación múltiple en base a un conjunto de elementos, se ha decidido utilizar una red basada en grafos con convolución transformer para obtener las relaciones temporales relevantes en el conjunto de datos.

Como se puede observar en la Figura 2.5 el modelo comienza con una capa convolucional basada en grafo con 64 nodos (y 2 por cada uno, uno por cada coordenada del punto). Posteriormente se introduce una capa de convolución transformer 64x64 con 4 cabezales. Seguidamente, una capa lineal, y 2 capas convolucionales basadas en grafos de dimension 64x64. Se incorpora una capa conexión skip para la convolución transformer.

Finalmente, la última capa produce 17 salidas, una por cada categoría a clasificar.

Con esta estructura, se pretende crear un conjunto de pesos capaz de analizar una sucesión de grafos (esqueletos) con memoria temporal en el conjunto de éstos y clasificar cada categoría

independientemente tras comprender el movimiento de cada punto del esqueleto teniendo en cuenta el conjunto de los mismos.

En este apartado, la única lógica a destacar es la creación de una función "collate" para alimentar a la red a partir del **dataloader** creado con la misma estructura que espera el modelo propuesto.

La división de los datos se realiza en la proporción 80 20, es decir, el 80 % de los datos destinados al entrenamiento y el 20 % restante a evaluación para comprobar el éxito del modelo en cada época y al finalizar.

Apartado 3

Resultados

En este apartado se van a exponer los resultados obtenidos con el sistema y modelo planteado.

Cabe destacar que estos resultados no son concluyentes para distinguir un buen sistema y modelo de uno malo. Ya que el conjunto de datos inicia con 10 sujetos, con un total de 60 experimentos, con la normalización se reduce este número, quedando un conjunto de datos para comprobar el correcto funcionamiento del sistema pero no suficiente para evaluar su rendimiento.

El loss del modelo comienza en $2,57 \times 10^3$ y, tras 100 épocas se reduce a $2,5 \times 10^3$.

La evaluación del modelo entrenado con 100 épocas y los pocos datos mencionados, se puede encontrar en la Tabla 3.1. El modelo ha sido evaluado según la pérdida obtenida, según la precisión y según la puntuación F1.

Loss	Accuracy	F1 score
92.1778	0	0.0538

Tabla 3.1: Resultados del modelo propuesto

Bibliografía

- Cosma, A. and Radoi, E. (2024). Psymo: A dataset for estimating self-reported psychological traits from gait. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4603–4613.
- Fang, H.-S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.-L., and Lu, C. (2022). Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.