Reconociendo la actividad humana en videos

Maria Pilar Madariaga Lasala

Universidad Internacional de la Rioja, Logroño (España)

Fecha 15/07/2021

Resumen

Este trabajo tiene como objetivo principal, la creación de una solución software (API) para *Human Action Recognition*, un tipo de problema perteneciente a la taxonomía de *Video Human Action Understanding*, que se centra en reconocer las acciones de las personas que aparecen en los datos de tipo video. Ideamos un proyecto para reconocer acciones de un dominio específico: el baile.

Para la creación de la API se ha estudiado la línea de investigación del campo Action Recognition (AR) en el estado del arte (SOTA). Siguiendo el ciclo de vida de los proyectos de IA, se han entrenado dos modelos: (I) MobileNet y (II) CNN con filtros convoluciones 3D. Los servicios de la API realizan inferencias a estos modelos de aprendizaje profundo, que han sido entrenados con conjuntos de datos de videos, que contienen las acciones que queremos inferir más adelante en videos de YouTube.

Palabras Clave

Action Recognition API, Large-scale video datasets, Video Human Action Understanding.

I. Introducción

En la actualidad, el campo de conocimiento Semantic Video Understanding (Comprensión Semántica de los vídeos), cobra relevancia en la comunidad científica, por las aplicaciones innovadoras que podrían implementarse sí fuéramos capaces de analizar los objetos, las acciones, los eventos y los conceptos de los videos en tiempo real [2]. Por ejemplo: Sistemas de Videovigilancia, Sistemas Asistenciales que interactúen con personas, Sistemas Inteligentes que corroboren el aprendizaje correcto de una actividad, Sistemas de Traducción de Lenguaje de Signos o de Señales, Sistemas de Predicción del comportamiento de un sujeto, Sistemas de Realidad Virtual, Sistemas de Conducción Automática, Sistemas de Detección de Videos Falsos, etc.

El objetivo principal del trabajo es contribuir al desarrollo de servicios basados en modelos de Inteligencia Artificial que puedan llegar a ser capaces de emular capacidades o características propias de la inteligencia humana.

El plan de acción para desarrollar la solución software (API), sigue el ciclo de vida de los proyectos de Inteligencia Artificial, distinguiendo las siguientes fases:

En la fase inicial, analizamos el estado del arte para contextualizar el problema. En la fase de exploración y comprensión de los datos, analizamos conjuntos de datos de tipo vídeo. En la fase de modelado de la solución, identificamos cuáles son los requerimientos del proyecto y seleccionamos herramientas adecuadas para el trabajo.

Una vez completada estas fases, que nos permiten tener una visión exploratoria del problema a resolver, continuamos con la fase de implementación de la solución, en la que entrenamos la red neuronal, ajustando sus hiperparámetros. En segundo lugar, construimos la API de action recognition, creando servicios que consuman los modelos entrenados.

Finalmente, en la fase de validación de la solución, comparamos las métricas de los modelos entrenados y validamos los servicios de la API.

II. Estado del Arte

2.1 Action Understanding

En este trabajo la palabra acción es un concepto clave. Kang y Wildes [8], definen acción como un movimiento creado por el cuerpo humano que puede ser cíclico o no. Desde la perspectiva computacional, una acción, podría definirse como una secuencia de imágenes (frames), que contienen una o varias acciones [7].

Observamos, que todas las acciones no son iguales ni suceden a la misma velocidad. Por ejemplo, podemos reconocer las acciones de las personas interactuando con objetos, las acciones de las personas interactuando con otras personas o simplemente, las acciones que realiza individualmente una persona para comprender su comportamiento en un contexto.

El campo Action Understanding, se concentra en estudiar las acciones que ocurren en la escena en fuentes dinámicas (videos) [20]. Según la naturaleza de los problemas, éstos se organizan en problemas de clasificación o problemas de búsqueda o localización, generando una amplia taxonomía de problemas [5].

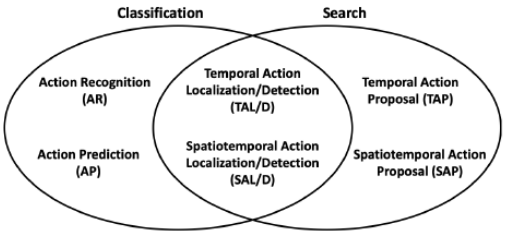

Fig1 Taxonomía de Action Understanding

Otra línea de investigación es el entendimiento de las acciones realizadas desde diferentes perspectivas. El campo Egocentric Action Undestanding (EAR), estudia las acciones desde la

perspectiva de la primera persona (perspectiva egocéntrica) con respecto a la perspectiva de cámara o vista de la tercera persona (perspectiva no egocéntrica o externa) [9].

Más aplicaciones con videos, de interés actual, en la comunidad son: Dense Captioning, que genera la descripción del video en formato texto, Action instance segmentation, que etiqueta cada instancia individual de una acción, incluso cuando existan varias acciones simultáneas, Action spotting, que localiza los instantes donde ocurre la acción y Object Tracking que detecta objetos entre los frames del video, frame a frame [5].

Es un reto entender las acciones que ocurren en escenarios realistas [7]. De hecho, un desafío clave recurrente a lo largo del tiempo, en el entrenamiento de modelos basados en redes neuronales profundas, es la falta de conjuntos de datos adecuados para la investigación de las tareas con fuentes de datos de videos, ya que estos modelos multiplican el número de parámetros de la red, en comparación con los modelos que usan las imágenes como fuente de datos. [3].

Existen más desafíos en la disciplina como reconocer las acciones en videos de duración larga [17], la definición de las anotaciones de las categorías del conjunto de datos, el sabotaje de la red neuronal (adversal attack)[22], el manejo de la información sensible de las personas que aparecen en los conjuntos de datos [8], la distribución del dataset a toda la comunidad científica o la búsqueda de arquitecturas más ligeras, para integrarlas en dispositivos portables que realicen inferencias en tiempo real[19].

2.2 Video Datasets

En esta última década, han aparecido conjuntos de datos a gran escala (large-scale datasets) [8,12,14,15,17,21], con un incremento considerable de elementos de las clases que se quieren reconocer. Son conjuntos de datos grandes, del orden de millones de muestras, elaborados para investigar las múltiples tareas de la taxonomía del Action Understanding. conteniendo muchas muestras de personas de distintas partes del mundo que realizan las acciones en distintos entornos, con distintas características de iluminación, con distinta duración de video y con distintos tipos de anotaciones (multi-label).

Algunos ejemplos de conjuntos de datos que sirven de benchmark para realizar pruebas de rendimiento en otros modelos de action understanding son: Hollywood-1, Hollywood-2, YouTube-8M, HMDB-51, UCF-50, UCF-101 (datasets), Sports-1M, ActivityNet, Charades, Multi-THUMOS, Kinetics (large-datasets) [20].

2.3 Preprocesamiento de datos de tipo video

Un video, se representa comúnmente como un volumen tridimensional (3D), con dos dimensiones espaciales y una dimensión temporal [2], cuyos frames están apilados a lo largo de la dimensión temporal. Además, si el video es a color, el volumen en realidad tiene una cuarta dimensión para tratar el canal del color RGB. Según el orden que se siga para definir el volumen, tenemos el formato channel first (frames, channels, height, widht) o channel last (frames, height, width, channels). Este orden puede permitir mejoría o degradación en el entrenamiento del modelo [5]

Algunos datasets de videos contienen los videos en bruto (raw RGB videos) y otros incluyen algún tipo de prepocesamiento previo [5]. Por ejemplo, el conjunto de datos LetsDance [15] facilita los frames de los videos en formato RGB, sin preprocesar, y a su vez, provee los flujos ópticos precalculados.

En ocasiones, resulta útil, emplear técnicas como Data cleaning [5], para eliminar datos irrelevantes en el dataset o Data Augmentation [5], para crear más elementos en el conjunto de datos. Generalmente, *color jittering*, que cambia el brillo, el contraste y la saturación del frame, así como la rotación de los frames, son técnicas de *data augmentation* que ayudan en el aprendizaje de la red neuronal, aumentando la precisión del modelo [2].

2.4 Redes Neuronales para Action Recognition

En trabajos previos, Carreira. J [20] indica que las CNNs y las RNNs han sido los tipos de redes neuronales más usadas para los problemas de Action Recognition.

Las redes neuronales convolucionales (CNNs) se componen de capas convolucionales, que realizan una extracción de características locales, capas de pooling, que agrupan características semánticamente similares en características más complejas, capas de normalización para evitar el overfitting (efecto de sobreentrenar un algoritmo de machine learning) y capas fully-connected, que conectan las neuronas de las capas entre sí. La última capa softmax funciona como un clasificador, seleccionando la clase con más probabilidad, según el conocimiento aprendido por la red neuronal.

La ventaja que aporta esta arquitectura con respecto a otras es que operan directamente sobre los datos en bruto, sin procesar y sin requerir ninguna extracción previa de características [7].

La desventaja es que sufren de exploding gradients (el gradiente se va agrandando e implica realizar actualizaciones importantes de los valores de los pesos del modelo) o vanishing gradients (el gradiente se va desvaneciendo en valores muy pequeños) [8].

Las redes neuronales recurrentes con memoria a largo y corto plazo (LSTM-RNNs) son un modelo de red recurrente, que permite almacenar y acceder a la información contextual de la secuencia temporal, donde el componente LSTM soluciona el problema del vanishing gradient principal desventaja de las arquitecturas RNNs [8]. La mayoría de los modelos basados en LSTM-RNNs, no pueden directamente trabajar sobre videos en bruto. No obstante, es frecuente que se usen las CNNs como extractores de características locales, que alimentan a la red LSTM, para fusionar la información extraída de la CNN con la de la secuencia temporal [7].

En la última década, han aparecido arquitecturas más complejas para el reconocimiento de la acción en los videos, que han ido mejorando la precisión, así como el tamaño y la latencia de la red neuronal, para integrarla en dispositivos embebidos.

Estas arquitecturas se han caracterizado por construirse con diseños *Single-Stream* o *Multi-Stream*, con bloques convolucionales de tipo 1D, 2D, 3D, diseños híbridos y técnicas de fusión para mejorar el rendimiento de las redes neuronales. Los métodos que utilizan capas convolucionales 3D suelen mejorar a los métodos más tradicionales, no obstante, supone emplear más recursos para entrenar las redes neuronales.[15]

En 2014, Simonyan y Zisserman, crearon la primera red neuronal profunda *Multi-Stream*, construida con dos redes funcionando en paralelo. El flujo espacial (Spatial-Stream), extrae las características espaciales (spatial data) del video sin preprocesar, es decir, extrae aquellas características visuales como la forma o el color. El flujo temporal (Temporal-Stream) calcula los flujos ópticos del video para extraer las características temporales (temporal data).

Las arquitecturas *Slow Fast* también usan dos flujos paralelos. El flujo o rama superior es la rama lenta, que opera con un video de baja velocidad para procesar detalladamente cada fotograma. La rama inferior, es la rama rápida, que opera con una versión de alta velocidad de cuadros temporales, de manera que se fusionan los resultados de ambas ramas en múltiples etapas.

La arquitectura *Late Fusion*, realiza un promediado en la última etapa de la red, a diferencia, de la arquitectura *Early Fusion*, que fusiona en las etapas iniciales, la dimensión temporal y la dimensión de canal RGB, generando un tensor (3T x H x W), que identifica los movimientos de píxeles locales entre los frames adyacentes.

Sin embargo, estos avances para mejorar la precisión no necesariamente hacen que las redes sean más eficientes en todos los sentidos. Un enfoque para diseñar redes neuronales más pequeñas es contraer, factorizar o comprimir redes preentrenadas con técnicas como la codificación de Huffman o la factorización de capas convolucionales, como en las arquitecturas MobileNet, que transforman las dimensiones de las capas convolucionales en dimensiones más pequeñas, simplificando tanto el tiempo de computación como el tamaño [19].

Actualmente, aunque los investigadores siguen estudiando el entendimiento de los videos, de cara a la elaboración del presente trabajo, nos basamos en dos líneas de investigación del estado del arte para la creación de la herramienta de software.

La primera línea de investigación que resaltamos es la selección de frames individuales del video para resolver el problema de clasificación de video mediante un clasificador de imágenes. La segunda línea de investigación pasa por entender el movimiento del video procesando la secuencia temporal de frames con redes LSTMs, donde a su vez, se podría combinar con características extraídas por una CNN.

2.5 Descriptores en el Action Recognition

Antes de que se considerasen las redes neuronales como algoritmos adecuados para resolver el problema del reconocimiento acciones, se han estudiado otros métodos basados en representaciones de video.

Podemos resumirlos en dos grupos: descriptores globales y locales, cuya diferencia principal es la observación de la información como un todo o como regiones independientes, respectivamente.

La finalidad de estos descriptores es (I) reconocer partes del cuerpo, (II) realizar el seguimiento del movimiento en la secuencia de frames de una o varias perspectivas de las cámaras o (III) reconocer patrones de movimientos.

Los descriptores globales, como Motion Energy Image (MEI) o Motion History Image (MHI), se basan en modelos, plantillas o siluetas para la representación de la estructura del cuerpo humano.

Los descriptores locales como Histogram of Gradient Orientations (HOG), extraen la apariencia; los Histogram of Optical Flow (HOF), extraen el movimiento y los Motion Boundary Histogram (MBH), extraen la variación del movimiento, característica que es invariante al movimiento de la cámara [19].

2.6 Action Understanding desde la perspectiva egocéntrica

En los últimos años, el reconocimiento de las acciones desde la perspectiva de la primera persona (egocentric perspective) ha despertado interés en varios aspectos.

Es un objetivo desafiante encontrar la relación que existe entre los videos con la perspectiva de la primera persona (vista egocéntrica) y con la perspectiva de la tercera persona (no egocéntrica) para aplicaciones de Realidad Aumentada, Realidad Virtual y el reconocimiento de las acciones desde la perspectiva egocéntrica (Egocentric Action Recognition) en pequeños dispositivos portátiles con cámara (wearable camera devices), como los *smartphones* y *las smart-glasses*. [2]

Sin embargo, la perspectiva egocéntrica añade desafíos nuevos a los previamente mencionados en el action recognition: movimientos rápidos de los objetos y las manos, e interacciones complejas entre la mano y los objetos, además, de las posibles oclusiones entre ambos [11].

Otro factor desafiante, es que estos dispositivos captan imágenes del entorno real y en ocasiones, habrá factores que emborronen o muevan la imagen grabada. Por ejemplo, en entornos donde llueva o haya cambios de luz, la imagen se verá peor y si la persona corre o salte abruptamente, la imagen se verá alterada con movimientos bruscos que desestabilicen la imagen.

Las redes neuronales profundas multi-stream han sido investigadas como solución para los problemas de Action Recognition (AR), así como para los problemas de Egocentric Action Recognition (EAR), usando la información captada por la cámara desde ambas perspectivas y de los sensores del dispositivo portable. Sin embargo, tales arquitecturas, requieren mucha energía y capacidad de cómputo, características que no se encuentran disponibles en estos dispositivos portables [9].

El ahorro de recursos ha motivado recientes líneas de investigación en esta área, ya que requiere comprender los movimientos de las manos y el cuerpo en la perspectiva egocéntrica y cómo interactúa el cuerpo con los objetos en entornos complejos.

En recientes publicaciones, se han utilizado flujos ópticos y señales de audio capturadas en los sensores del dispositivo, como información efectiva en los problemas EAR, así como modelos basados en métodos de atención o aprendizaje por refuerzo.

III. Objetivos y Metodología

El objetivo general del trabajo es implementar una herramienta software (API) que sea capaz de detectar la actividad humana en los videos dentro de unas métricas aceptables.

Para lograr el objetivo de este trabajo empleamos técnicas de aprendizaje profundo como la técnica de Transfer Learning y Fine Tunning, que permiten usar modelos pre-entrenados para mejorar significativamente la eficiencia del entrenamiento del nuevo modelo.

Como objetivos específicos estudiaremos mejorar las métricas de precisión o tiempos de entrenamiento de los modelos que se integren en los servicios de la API.

La metodología empleada para diseñar la solución software, sigue las siguientes fases, acorde al ciclo de vida de los proyectos de Inteligencia Artificial. En la Fase de Investigación, buscamos conjuntos de datos de videos a gran escala, adecuados para el problema que queremos resolver. En la Fase de Exploración, analizamos los conjuntos de datos tanto de perspectiva egocéntrica y no egocéntrica, identificando la duración de los clips de video, las anotaciones, la distribución de las muestras y el subconjunto de etiquetas, con objeto de seleccionar un único dataset con el que trabajar. En la Fase de Modelado, construimos el prototipo del modelo y evaluamos los resultados del prototipo para construir el modelo final. Por último, en la Fase de Análisis de Resultados, detallamos los resultados obtenidos y analizamos aquellos aspectos que puedan no haber salido como esperamos y planteamos una solución al respecto.

IV. CONTRIBUCIÓN

El objetivo práctico de la solución software es crear una API de servicios REST para resolver problemas de tipo Action Recognition, reconociendo las acciones que realizan las personas en los videos.

En este caso de uso, el dominio de las acciones a reconocer se focaliza en el dominio de la danza, es decir, en reconocer diferentes estilos de baile del dataset *Let'sDance (Learning from Online Dance Videos)* creado por la Universidad de Georgia en 2018 [15].

Las acciones de baile son acciones de grano fino y se realizan a distintas velocidades según, la persona o la música que interpretan. Por consiguiente, es un dataset interesante para trabajar los aspectos descritos en el capítulo 2.

La API, estará formada por un conjunto de servicios REST, que recibirán una URL de un video de YouTube. Tras su procesamiento, el servicio devolverá la etiqueta inferida de la acción identificada en el mismo, así como un porcentaje de confianza de la predicción. Los modelos pueden reconocer los siguientes estilos de baile: *Ballet, Break, Flamenco y Waltz*.

V. EVALUACIÓN Y RESULTADOS

Se evalúan dos modelos que se integran en dos servicios de la API y se comparan dos escenarios de evaluación: (I) Entrenamiento del modelo y (II) Pruebas de Inferencia con la herramienta software (API).

Evaluación 1: Entrenamiento de los modelos

El primer modelo se ha construido sobre el modelo MobileNet mediante la técnica de Transfer Learning, entrenando las últimas 5 capas de la red y ajustando los hiperparámetros del modelo. Para este modelo hemos utilizado el enfoque Single-Frame, donde procesamos los frames de cada video, de forma individual, entrenando el modelo como un clasificador de imágenes que aprende las características espaciales (spatial data).

El segundo modelo se ha diseñado desde cero, con una arquitectura CNN con filtros convolucionales 3D. Para este modelo hemos utilizado el enfoque que aprende las características espaciotemporales (spatiotemporal data), procesando cada video como una secuencia de frames donde los filtros convolucionales 3D son capaces de extraer dicha información temporal.

	Modelo 1	Modelo 2
Epochs	10	10
ImageDataGenerator (Batch size)	Yes/ 32	No
Función de pérdida (Loss)	Categorical	Categorical
Optimizador	Adam (learning rate: 1e-3)	Adam (learning rate: 1e-4)
Tiempo de entrenamiento:	46 minutos	29 minutos
Entorno de entrenamiento:	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
Uso de GPU	No	No
RAM	RAM 16,0 GB	RAM 16,0 GB
Métricas durante el entrenamiento (loss, accuracy, val loss, val accuracy)	loss: 0.0018 - accuracy: 1.00 val_loss: 0.45 - val_accuracy: 0.85	loss: 0.0542 - accuracy: 0.9900 - val_loss: 1.0706 - val_accuracy: 0.7103
Tiempo medio por epoch	223s	213s
Elementos por epoch	113	148
Total params del model	3,232,964	986,852
Trainable params	1,867,780	986,852
Non-trainable params	1,365,184	0

Fig 2 Métricas de los modelos en la fase de entrenamiento

	Modelo 1	Modelo 2
Métricas durante la evaluación (test loss, test accuracy)	loss: 0.556 –accuracy: 0.837	loss: 1.143 - accuracy: 0.687
Métricas de la clasificación (precision, recall, f1-score, accuracy)	precision recall f1-score 0 0.76 0.86 0.81 1 0.87 0.93 0.90 2 0.91 0.56 0.70 3 0.84 1.00 0.92	precision recall f1-score 0 0.68 0.56 0.61 1 0.80 0.64 0.71 2 0.55 0.66 0.60 3 0.75 0.89 0.82
Precisión del modelo/ ejemplos de test	0.84 / 1800	0.69 /2385
Etiquetas de clasificación (labels)	0: ballet 1: break	0: ballet 1: break

Fig 3 Métricas de los modelos en la fase de testing

Haciendo valoración de los resultados obtenidos en la fase de entrenamiento, observamos un valor para la métrica de precisión más elevado en el modelo 1 que en el modelo 2 en las mismas condiciones de entrenamiento (epochs y batch_size). Además, confirmamos que las arquitecturas CNNs con filtros 3D son más costosas que las capas convolucionales 2D o las capas factorizadas de la arquitectura mobileNet. Por otro lado, observamos el beneficio obtenido, en los problemas que emplean large-scale datasets, seleccionando un modelo pre-entrenado con imagenet, en el que aplicando la técnica de Transfer Learning entrenaremos solo las últimas capas del modelo, para identificar los nuevos ejemplos, que queremos que el modelo aprenda, en menor tiempo y coste computacional.

Evaluación 2: Validación de la herramienta

Con respecto a las pruebas de validación de la solución software (API), hemos evaluado los resultados obtenidos con 2 servicios (api/v1/predict, api/v2/predict) que consumen el modelo 1 y el modelo 2, respectivamente.

Para estas pruebas emplearemos videos de Youtube (wild videos) en el que aparecen personas bailando los estilos aprendidos por los modelos (Ballet, Waltz, Flamenco y Break Dance).

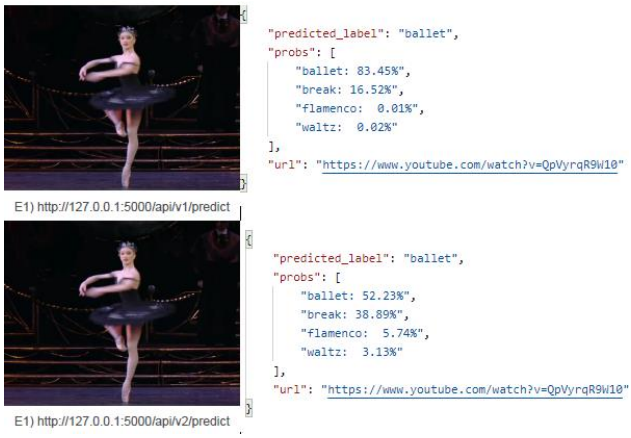

Fig 4 Ejemplo de inferencia de los servicios de la API

Hemos observado que el modelo 2 realiza una predicción ligeramente, más acertada que el modelo 1, pese haber obtenido una precisión menor. Esto puede deberse a que el modelo 2 a pesar de ser sencillo, emplea capas convolucionales 3D que son capaces de capturar las características espacio temporales, a diferencia del modelo 1 que solo aprende a clasificar los estilos de baile por sus características espaciales: forma, color, etc.

Fig 5 Ejemplo de inferencia de los servicios de la API

Parece que el modelo 2 con la información espaciotemporal sí que puede acertar en la predicción, a pesar de que algunos ejemplos mezclen algunos movimientos de manos o escenarios más típicos de otros estilos, como puede ocurrir en el ejemplo anterior donde las batallas de break dance suelen realizarse en el suelo y tal vez, el modelo 1, no ha sido de reconocer la guitarra o la vestimenta de la bailarina desde ese ángulo de cámara.

VI. Discusión

Reconocer la naturaleza de una acción (Action Recognition) en los videos se categoriza como un problema de clasificación. En ocasiones, no es suficiente con fijarse solamente en un frame del video (Single-Frame) para reconocer la acción, sino que es necesario, observar una secuencia de frames consecutiva, discriminando acciones de grano fino.

Si bien, pueden extenderse las técnicas de clasificación de imágenes en el dominio de los videos, para clasificar cada frame de forma individual, realizando un promediado del conjunto predicciones individuales para aumentar la confianza de la predicción, a la hora de reconocer una tipología de acciones de grado fino, es importante que el modelo aprenda el movimiento que se produce en la secuencia temporal, ya que en ocasiones no es suficiente con aprender las características espaciales de la imagen para distinguir dos acciones entre sí observando solo la forma, el color o contexto de un frame de forma aislada.

En este trabajo hemos aplicado filtros convolucionales 3D para extraer las características espaciotemporales de los videos, sin embargo, podríamos haber extraído características de movimiento con los flujos ópticos (OF).

VII. Conclusiones

En el presente trabajo hemos expuesto la taxonomía de problemas del Video Action Understanding que auna conocimientos de Visión por Computador y Deep Learning junto con la creación de una API de servicios REST sobre la que poder construir aplicaciones de reconocimiento de acciones.

Como trabajo futuro, podemos (I) mejorar la robustez de los modelos aumentando la precisión de los modelos de la API, (II) ampliar los servicios de la API para solventar otras modalidades de la taxonomía del Action Understanding como por ejemplo, la detección del frame concreto dónde ocurre una acción específica o la predicción del comportamiento de un sujeto, tan solo observando los primeros frames del video o (III) emplear modelos de arquitecturas multi-stream que aprovechen las características extraídas de los flujos ópticos o de la pose.

Un elemento clave para poder trabajar con dataset a gran escala en formato video, es disponer de una GPU en el entorno de trabajo, pues aumenta la capacidad de computación de las operaciones matriciales y permite ampliar el número de muestras por cada clase, así como el número de categorías del modelo

Referencias

[1] Liu, M. (2021, May 20). Egocentric Activity Recognition and Localization on a 3D Map. ArXiv.Org. https://arxiv.org/abs/2105.09544

[2] Zhu, Y. (2020, December 11). A Comprehensive Study of Deep Video Action Recognition. ArXiv.Org. https://arxiv.org/abs/2012.06567

[3] Herath, S. (2016, May 16). Going Deeper into Action Recognition: A Survey. ArXiv.Org. https://arxiv.org/abs/1605.04988

[4] HMDB: A large video database for human motion recognition. (2011, November 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/6126543

[5] Hutchinson, M. (2020, October 13). Video Action Understanding: A Tutorial. ArXiv.Org. https://arxiv.org/abs/2010.06647

[6] Kondratyuk, D. (2021, March 21). MoViNets: Mobile Video Networks for Efficient Video Recognition. ArXiv.Org. https://arxiv.org/abs/2103.11511

[7] Recent advances in video-based human action recognition using deep learning: A review. (2017, May 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/7966210

[8] Piergiovanni, A. J. (2020, July 10). AViD Dataset: Anonymized Videos from Diverse Countries. ArXiv.Org. https://arxiv.org/abs/2007.05515

[9] Egocentric Activity Recognition on a Budget. (2018, June 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8578723

[10] Sigurdsson, G. A. (2017, August 9). What Actions are Needed for Understanding Human Actions in Videos? ArXiv.Org. https://arxiv.org/abs/1708.02696

[11] Perez-Rua, J. (2020, July 3). Egocentric Action Recognition by Video Attention and Temporal Context. ArXiv.Org. https://arxiv.org/abs/2007.01883

[12] Sigurdsson, G. A. (2018, April 25). Charades-Ego: A Large-Scale Dataset of Paired Third and First Person Videos. ArXiv.Org. https://arxiv.org/abs/1804.09626

[13] Simonyan, K. (2014, June 9). Two-Stream Convolutional Networks for Action Recognition in Videos. ArXiv.Org. https://arxiv.org/abs/1406.2199.

[14] Wu, Z. (2019, June 12). Privacy-Preserving Deep Action Recognition: An Adversarial Learning Framework and A New Dataset. ArXiv.Org. https://arxiv.org/abs/1906.05675

[15] Castro, D. (2018, January 23). Let's Dance: Learning From Online Dance Videos. ArXiv.Org. https://arxiv.org/abs/1801.07388

[16] Huo, Y. (2019, August 27). Mobile Video Action Recognition. ArXiv.Org. https://arxiv.org/abs/1908.10155

[17] Huang, Q. (2020, July 21). MovieNet: A Holistic Dataset for Movie

Understanding. ArXiv.Org. https://arxiv.org/abs/2007.10937

[18] Liu, Y. (2021, May 24). FineAction: A Fined Video Dataset for Temporal Action Localization. ArXiv.Org. https://arxiv.org/abs/2105.11107

[19] Howard, A. G. (2017, April 17). MobileNets: Efficient Convolutional Neural Networks for Mobile. ArXiv.Org. https://arxiv.org/abs/1704.04861

[20] Carreira, J. (2017, May 22). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. ArXiv.Org. https://arxiv.org/abs/1705.07750

[21] Smaira, L. (2020, October 21). A Short Note on the Kinetics-700-2020 Human Action Dataset. ArXiv.Org. https://arxiv.org/abs/2010.10864

[22] Xie, C. (2019b, November 21). Adversarial Examples Improve Image Recognition. ArXiv.Org. https://arxiv.org/abs/1911.09665

[23] Wang, H. (2013, March 6). Dense Trajectories and Motion Boundary Descriptors for Action Recognition. International Journal of Computer Vision. https://doi.org/10.1007/s11263-012-0594-8

[24] Parmar, D. (2020, June 6). How to handle REST requests in Flask. TheBinaryNotes. https://thebinarynotes.com/how-to-handle-rest-requests-flask/

[25] Parmar, D. (2020b, June 6). Video Classification in Keras using ConvLSTM. TheBinaryNotes. https://thebinarynotes.com/video-classification-keras-convlstm/

[26] Rosebrock, A. (2021, April 17). Video classification with Keras and Deep Learning. PyImageSearch. https://www.pyimagesearch.com/2019/07/15/video-classification-with-keras-and-deep-learning/

[27] Rosebrock, A. (2021a, April 17). Fine-tuning with Keras and Deep Learning. PyImageSearch.https://www.pyimagesearch.com/2019/06/03/fine-tuning-with-keras-and-deep-learning/

[28] Rosebrock, A. (2021, April 17). Human Activity Recognition with OpenCV and Deep Learning. PyImageSearch. https://www.pyimagesearch.com/2019/11/25/human-activity-recognition-with-opencv-and-deep-learning/

[29] Keras with TensorFlow Course - Python Deep Learning and Neural Networks forBeginners Tutorial. (2020, June 18). YouTube. https://www.youtube.com/watch?v=qFJeN9V1ZsI

[30] Sayak, P. (2021, June 5). Keras documentation: Video Classification with a CNN-RNN Architecture. Keras. https://keras.io/examples/vision/video_classification/

[31] Anwar, T. (2021, March 8). Introduction to Video Classification and Human Activity Recognition. Learnopencv. https://learnopencv.com/introduction-to-video-classification-and-human-activity-recognition/

[32] Brownlee, J. (2019, August 14). How to Use the TimeDistributed Layer in Keras. Machine Learning Mastery. https://machinelearningmastery.com/timedistributed-layer-for-long-short-term-memory-networks-in-python/

[33] Shorten, C. (2019, January 15). Introduction to Video Classification - Towards Data Science. Medium. https://towardsdatascience.com/introduction-to-video-classification-6c6acbc57356

[34] Fernández, Y. (2020, March 5). Qué son los FPS o fotogramas por segundo, y para qué sirven en los videojuegos. Xataka. https://www.xataka.com/basics/que-fps-fotogramas-segundo-sirven-videojuegos