Automatic detection of hand washing quality using deep learning

Diana Carolina Cabrera[1]
Miguel Angel Ortiz[2] Daniel Alfonso Garavito[3]

***Abstract*—Los desarrollos en aprendizaje de máquina han contribuido abordar tópicos alcanzables a la imaginación como los de detección automática de lavado de manos, en donde, a través de la exploración de un conjunto de datos tomados de kaggle se realiza un desarrollo extenso por medio de varias revisiones de literatura. Por consiguiente, se utiliza una librería Decord para cargar videos rápidamente. Luego, se muestrea con un stride temporal el cual disminuye la resolución temporal. Así mismo, se forman clips con un tamaño especifico de 16 frames a partir de todos los posibles subclips de un vídeo. Se procede a dividir aleatoriamente 25 vídeos donde se realizan los pasos recomendados por la OMS para el lavado de manos, estableciendo los conjuntos de datos de entrenamiento, validación y prueba, con 15 vídeos, 5 vídeos, y, 5 vídeos respectivamente. Cada vídeo es separado en 12 pasos para un total de 300 videos con longitud promedio de 12.9 segundos. Por último, se aplican modelos con convoluciones espacio-temporales basados en ResNet. ResNet(2+1)D, ResNet 3D, y MC3 con convoluciones mixtas; obteniendo los mejores resultados en el conjunto de prueba con MC3 con 95% para predicciónes con 1 clip y 97% para predicciones con 2 clips post-procesadas.**

I. Video

Enlace

II. Objetivos

- Emplear métodos de reconocimiento de acciones abordados en la literatura para el tratamiento del conjunto de datos empleado.
- Implementar modelos residuales que permitan identificar los pasos del lavado de manos recomendados por la Organización Mundial de la salud (OMS), como ResNet(2+1)D, ResNet 3D (R3D), ResNet Mixed Convolution mc3.

III. Introduction

La pandemia del coronavirus ha demostrado la importancia de hábitos de aseo para la prevención de enfermedades transmisibles. El lavado de manos es el mecanismo de control más costo - eficiente para la prevención no solo del COVID-19 sino múltiples enfermedades respiratorias e intestinales.

[1]D. Cabrera, Faculty of Systems Engineering, National University, Bogota, Colombia
[2]M. Ortiz, Faculty of Systems Engineering, National University, Bogota, Colombia
[3]D. Garavito, Faculty of Systems Engineering, National University, Bogota, Colombia

Esta crisis nos enfrentó al hecho de que no dominamos una competencia básica e importante como es el lavado de manos. Es más, como organismos vivos tendemos al camino de menor resistencia, lastimosamente iremos normalizando la pandemia al punto que la pereza dictará nuestras acciones y terminaremos por volver a nuestro hábitos anteriores, a no lavarnos las manos o no hacerlo adecuadamente.

Los recientes avances en Machine Learning han permitido hacer seguimiento a la posición de partes del cuerpo, que pueden ser empleadas para el seguimiento del lavado de manos y la evaluación del mismo. El objetivo es mejorar la calidad y adherencia del lavado de manos mediante el uso de herramientas de Machine Learning y formar sólidos hábitos de higiene.

Al plantear este problema, la búsqueda de un conjunto de datos para realizar la implementación planteada, evidenció que no se cuenta con un conjunto de datos etiquetado lo suficientemente grande para entrenar modelos que permitan mejorar la calidad del lavado de manos. Por esta razón, decidimos basarnos en un conjunto de datos que contienen vídeos de las diferentes poses al practicar el lavado de manos.

IV. Antecedentes

El procesamiento de imágenes ha alcanzado un altísimo rendimiento gracias al aprendizaje profundo El procesamiento de vídeo a pesar de tratarse del apilamiento de imágenes, tiene sus propios desafíos. Desde el elevado costo computacional a la vibración de la cámara, la definición de las arquitecturas. Entre otras.

Con el objetivo de abordar las implementaciones que otros investigadores han realizado, se plantea un mapa como línea de tiempo de los métodos en el área:

En el año 2016, empezaron a sonar métodos ”Convolutional Two-Stream Network”, proponiendo una arquitectura ConvNet para la fusión espacio-temporal de fragmentos de vídeo, en donde, agregan la combinación de características abstractas en espacio-temporales, específicamente fusionan las redes espacialmente en la última capa convolución, con el fin de aumentar la precisión y el rendimiento [1].

Más tarde, en el año 2018 investigadores decidieron ahondar en métodos como ”Two-Stream Inflated 3D ConvNets (I3D)”, debido a que se presentan nuevos retos al tratar diferentes conjuntos de datos. Por esta razón, diseñan una arquitectura de clasificación de imágenes, en donde su core es inflar los filtros y núcleos de agrupación dentro una 3D [2].

En el mismo año, en paralelo un nuevo grupo de mentes brillantes deciden plantear ”Neural Graph Matching Networks (NGM)” que permite reconocer una clase de acción 3D invisible con solo unos pocas muestras. Esto le ayuda al modelo generar una buena eficiencia con pocos datos de aprendizaje [3]. De la misma manera, al visualizar el alto costo computacional al implementar estos algoritmos, otro grupo de investigadores crean un modulo de cambio temporal para una comprensión eficiente del vídeo(TSM) presentando una alta eficiencia y rendimiento en conjunto de datos de vídeos en linea de baja latencia en tiempo real[4].

Después de unos meses, se presentan redes slowfast para reconocimiento de vídeo, mezclando el desempeño a baja velocidad de frames para capturar semántica y el desempeño a alta velocidad de frames para capturar una resolución fina brindando un rendimiento tanto para la clasificación de acción como para la detección en vídeo[5].

A principios de este año, nuevas arquitecturas en expansión para reconocimiento de vídeo eficiente han sido inspiración para abordar trabajos que se han limitado por la complejidad, y gracias al planteamiento de una expansión progresiva hacia adelante seguida de una contracción hacia atrás, en donde se requiere menos parámetros y una precisión comparable con otros modelos implementados[6].

Actualmente, se siguen ahondado en pruebas exploratorias en diferentes dataset, con el fin de comprobar que al agregar un modelo como ResNet(2+1)D y ResNet 3D existe una pequeña diferencia en los resultados obtenidos de rendimiento [7].

En nuestro proyecto, se decidió abordar tres tipos de modelos [8]:

- ResNet 3D: Generaliza ResNet con convoluciones con kernels 3D que preservan la información temporal y la propagan a través de las capas de la red.
- MC3 ResNet Mixed Convolutional: Esta basado en aplicar convoluciones 2D en los grupos 1 y 2, y convoluciones 3D en el grupo 3 y grupos más profundos. En su totalidad son 18 capas, puede variar dependiendo de la implementación.
- ResNet (2+1)D: Es la descomposición de convoluciones 3D basado en una convolución 2D seguida de una convolución 1D, intercalando el modelado espacial y temporal.

V. Materiales y métodos

Para el entrenamiento usaremos el dataset HandWash de lavado de manos disponible en Kaggle.

Este dataset busca identificar cada paso en el procedimiento de lavado de manos aceptado por la OMS. El conjunto de datos consta de 300 vídeos (cada lavado de manos tiene 12 pasos), en diferentes entornos para proporcionar la mayor variabilidad posible(Iluminación, fondo, posición de la cámara, Campo de visión, individuos)

A. Estadísticas

El conjunto de datos consta de 25 vídeos por cada paso del lavado de manos, dado que son 12 pasos se obtiene un total de 300 vídeos.

Todos los vídeos son de 30 cuadros por segundo y hay 200 de ellos en resolución 720x480 y 50 en full hd (1920 x 1080)

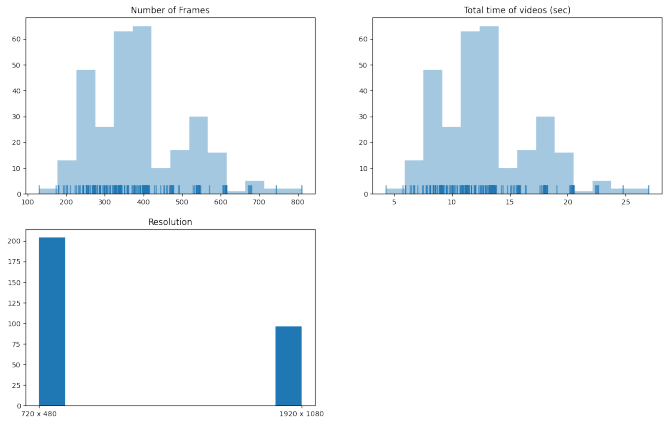

Figure 1. Frecuencia de vídeos por tiempo y calidad

La duración de los vídeos varía entre los 5 y 30 segundos aunque el promedio esta en 13 segundos con una desviación estándar de 4.1 segundos.

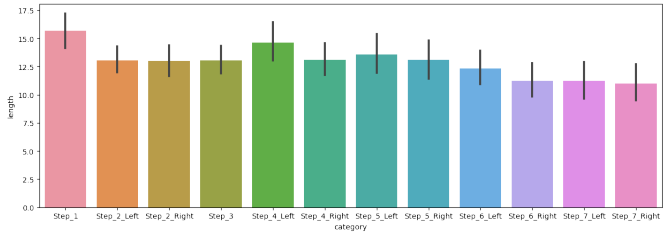

Figure 2. Duración promedia de vídeos de acuerdo al paso del lavado de manos

VI. Preprocesamiento

Para el preprocesamiento se utilizo: Decord es una biblioteca para el cargue rápido el aprendizaje profundo con un backend de c++ y fácil conexión con bibliotecas de aprendizaje profundo. Tiene la ventaja de que facilita la aleatorización y cargue.

Como se aprecia en la siguiente gráfica, la decodificación es más rápida que el con otras librerias como cv2 o el pyAV. (tomado del repositorio oficial de decord)

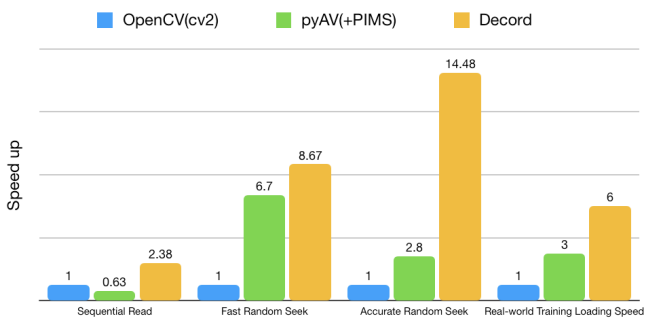

Figure 3. Comparativa de velocidad

A. *Muestreo de vídeos, preprocesamiento y creación de clases de Dataset*

Existen muchos enfoques a la hora de extraer muestras de vídeos cortos, la mayoría de ellos hacen un muestreo **denso** o uniforme. Comúnmente se muestrea con el fin de reducir los cuadros por segundo a un tamaño fijo de 16 fotogramas (tamaño fijo), que luego se introducirían en un modelo (por ejemplo, en el conjunto de datos de Jester, la longitud media del conjunto de datos de reconocimiento de gestos actualmente más grande es de 3 segundos). Algunos métodos avanzados incluyen el muestreo de cuadros por segundo adaptativo.

No obstante, el conjunto de datos empleado tiene una pequeña cantidad de vídeos pero de larga duración, 12.9 segundos en promedio. Por lo cual extraer solo 16 fotogramas cuando el promedio es de 387 fotograma significaría una perdida promedio del 95,86% de los datos.

B. *Muestreo con salto temporal τ_t*

Por otro lado, utilizar cada fotograma de un vídeo sería un desperdicio de recursos ya que los fotogramas secuenciales tienen una variación muy pequeña. En lugar de usar secuencialmente cuadro a cuadro, remuestreamos un video con paso temporal.

La resolución temporal $\frac{1}{\tau_t}$ con 1.0 siendo la tasa de fotogramas original y con $\tau_t = 2$ la mitad de la tasa de fotogramas original. Nuestro objetivo es explorar diferentes valores por τ_t.

C. *Muestreo de Clips*

Después de remuestrear un video con un τ_t especificado, formamos clips de un tamaño fijo τ_{size} (por ejemplo 16). El número total de subclips de un vídeo con fotogramas de n es de $\left\lfloor \frac{n}{\tau_t} \right\rfloor - \tau_{size} + 1$.

Para muestrear un clip tomamos una posición aleatoria inicial $\left\lfloor \frac{n}{\tau_t} \right\rfloor - \tau_{size}$ (inclusive) y seleccionamos los marcos τ_{size} en el rango $[p_0, p_0 + \tau_{size}]$.

temporal stride	frame rate	train	val	test
1.00	30.00	65229	19141	27333
1.25	24.00	51572	15109	21663
2.00	15.00	31228	9108	13199
2.50	12.00	24402	7096	10371
3.75	8.00	15337	4415	6605
4.00	7.50	14220	4089	6133
7.50	4.00	6284	1744	2841
8.00	3.75	5717	1582	2602

Table I
TABLA CON EL NÚMERO RESULTANTE DE CLIPS PARA DISTINTOS SALTOS TEMPORALES

D. *División en Entrenamiento, Validación y Prueba*

Dado que el conjunto de datos solo cuenta con 25 grabaciones, con cada una dividida en 12 pasos. Para la división entre entrenamiento, validación y prueba seleccionamos 15 grabaciones al azar y las guardamos para el entrenamiento, luego otras 5 se usan para la validación y las 5 restantes se usan para la prueba. A continuación se presentan el número resultante de clips para diferentes τ_t y una semilla aleatoria fija.

	Stride Temporal			
	1		2	
	Clip			
	1	2	1	2
R(2+1)D	0.87	0.90	0.88	0.92
R3D	0.88	0.93	0.92	0.92
MC3	0.95	**0.97**	0.95	0.95

Figure 4. Resultados de los modelos

VII. ENTRENAMIENTO Y VALIDACIÓN

Figure 5. Ejemplo de batch

A. *Entrenamiento*

1) Preprocesamiento: - Reducimos la resolución espacial de los vídeos (redimensionar a 256x256) con ffmpeg, esto acelera el entrenamiento ya que redimensionar es el paso de preprocesamiento más caro para las imágenes.

- En cada época muestreamos un clip para cada vídeo del conjunto de entrenamiento aleatoriamente , de esta manera evitamos el sesgo que puede incorporar las clases (pasos del lavado de manos) con vídeos más largos.

- Durante el entrenamiento, recortaremos aleatoriamente las imágenes de un clip a su tamaño (172 x 128). Cada fotograma se recorta igual que los otros fotogramas de un clip, pero los diferentes clips pueden recibir diferentes recortes. También realizamos un giro horizontal aleatorio a los clips durante el entrenamiento. El giro horizontal y el recorte aleatorio se emplea como mecanismo de aumento de datos.

No se recortan aleatoriamente los frames de los clips pues esto destruiría la correlación temporal. Además en la evaluación no realizamos recortes aleatorios por lo que esto cierra la disparidad tren/evaluación.

2) Optimizador: Probamos el estándar Adam y el SGD con momentum.

B. Evaluación

1) Preprocesamiento: - El redimensionamiento se hace de la misma manera que el entrenamiento. Todos los clips se recortan en el centro a su tamaño (112 x 112)

- Muestreamos un número fijo (por ejemplo, 2) de clips de cada vídeo en lugar de uno.

- Cada clip de un vídeo se procesa independientemente de los demás, pero las predicciones de cada uno de ellos se postprocesan para mejorar el rendimiento de la predicción de un vídeo (por ejemplo, promediando las puntuaciones de softmax).

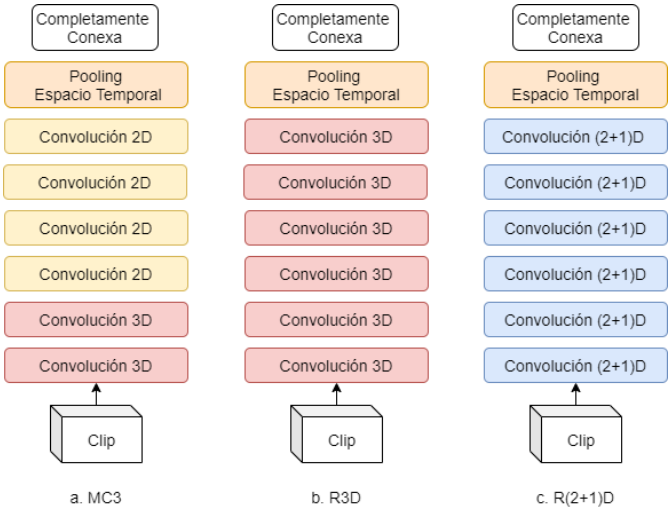

Figure 6. Arquitecturas

Las dos arquitecturas probadas en este documento corresponden a unas redes convoluciones 3D (r3d_18) y a una R(2+1) (r2plus1d_18).

La R3d_18 preservan la información temporal y la propagan a través de las capas de la red mientras que la r2plus1d aproxima la configuración espacio temporal por medio de una convolución 2D seguida de una convolución 1D, descomponiendo el modelado espacial y temporal en dos pasos independientes [8]

	Stride Temporal			
	1		2	
	Clip			
	1	2	1	2
R(2+1)D	0.87	0.90	0.88	0.92
R3D	0.88	0.93	0.92	0.92
MC3	0.95	**0.97**	0.95	0.95

Figure 7. Arquitecturas

- Los cuadros por segundo y el muestreo de cuadros son factores muy importantes para el análisis de vídeos.
- Probamos post-procesamiento de las predicciones de clips para mejorar el rendimiento en prueba

VIII. Trabajo Futuro

- Los resultados de los modelos de entrenamiento nos indican a trabajar con modelos mas livianos, como es una tendencia reciente en la creación de arquitecturas X3D [6], y, TSM [4] para el problema abordado.
- El conjunto de datos, debería extenderse para ser usado en producción, ya que, presenta pocas variaciones, siendo esté pequeño en comparación de conjuntos de datos de vídeos.

IX. Discusión

- El hecho de que la implementación más parsimoniosa mejore el desempeño sugiere que es posible simplificar aún más la arquitectura.
- Los datos presentados a los modelos son distintos en cada epoch, lo que puede ocasionar un entrenamiento caótico y resultados con alta varianza.
- los modelos generados no están enfocados a reconocimiento de gestos, sino de acciones.

X. Conclusiones

- La MC3 además de presentar un mayor desempeño tiene una tercera parte de los parámetros.
- Modelos relativamente sencillos consiguen muy buenos resultados.
- No usar características “hechas a mano” es un gran beneficio
- El conjunto de datos no presenta la variabilidad necesaria para ponerlo en producción. No obstante muestra que las arquitecturas empleadas son buenas candidatas atender en problema.

References

[1] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional Two-Stream Network Fusion for Video Action Recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, no. i, pp. 1933–1941, 2016.

[2] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 4724–4733, 2017.

[3] M. Guo, E. Chou, D. A. Huang, S. Song, S. Yeung, and L. Fei-Fei, "Neural Graph Matching Networks for Fewshot 3D Action Recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11205 LNCS, pp. 673–689, 2018.

[4] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 7082–7092, 2019.

[5] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 6201–6210, 2019.

[6] C. Feichtenhofer, "X3D: Expanding Architectures for Efficient Video Recognition," 2020.

[7] H. Kataoka, T. Wakamiya, K. Hara, and Y. Satoh, "Would Mega-scale Datasets Further Enhance Spatiotemporal 3D CNNs?," 2020.

[8] D. Tran, H. Wang, L. Torresani, J. Ray, Y. Lecun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.