UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis automático del uso de espacios a través del reconocimiento de personas en videos

Marcelo Ortega

Santiago Gómez Siri

Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Diciembre de 2018

UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis automático del uso de espacios a través del reconocimiento de personas en videos

Marcelo Ortega

Santiago Gómez Siri

Tesis de Grado presentada a la Facultad de Ingeniería de la Universidad de la República, como parte de los requisitos necesarios para la obtención del título de Ingeniero en Computación.

Director:
Dr. Ing. Prof. Pablo Rodriguez Bocca

Director académico:
Dr. Ing. Prof. Pablo Rodriguez Bocca

Montevideo – Uruguay
Diciembre de 2018

Ortega, Marcelo
Gómez Siri, Santiago

Análisis automático del uso de espacios a través del reconocimiento de personas en videos / Marcelo Ortega. - Montevideo: Universidad de la República, Facultad de Ingeniería, 2018.

VII, 80 p. 29, 7cm.

Director:

Pablo Rodriguez Bocca

Director académico:

Pablo Rodriguez Bocca

Tesis de Grado – Universidad de la República, Programa de Ingeniería en Computación, 2018.

Referencias bibliográficas: p. 76 – 80.

I. Rodriguez Bocca, Pablo, . II. Universidad de la República, Programa de Posgrado de Ingeniería en Computación. III. Título.

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS

Dra. Ing. Prof. Libertad Tansini

Dr. Ing. Prof. Pedro Piñeyro

Dra. Ing. Prof. Regina Motz

Montevideo – Uruguay
Diciembre de 2018

RESUMEN

En esta tesis se estudia la viabilidad de implementar un sistema que, apoyado en las técnicas del estado del arte para la detección y seguimiento de personas en videos, analice cuantitativa y cualitativamente el uso de un espacio físico. La condición del sistema es que funcione únicamente con los videos capturados por las cámaras ya instaladas en el local, generalmente de videovigilancia. Con esto se logra obtener una información de valor a partir de horas de grabaciones que en la mayoría de los casos son solamente utilizadas como evidencia ante algún siniestro o hurto, utilizándolas para determinar el comportamiento de las personas que visitan un lugar. Esto además minimiza el soporte de hardware adicional requerido, y por consiguiente su implantación, y amplía sus posibilidades de adopción por parte de potenciales interesados. El sistema debe además ofrecer funcionalidades con un alto grado de generalidad para ser aplicado a un espacio independientemente de sus características.

Para evaluar la viabilidad de este tipo de soluciones, en este trabajo se realizó un prototipo para analizar el comportamiento de los clientes de una cafetería ubicada en la zona céntrica de Montevideo. El prototipo utiliza para la detección de poses humanas la herramienta OpenPose, y realiza un seguimiento de ellas a través del video con el algoritmo DeepSort. Los usuarios puden realizar consultas sobre la permanencia de personas en áreas de la cafetería a través de una plataforma Web. Si bien la solución ayuda a determinar patrones de comportamiento del público en la cafetería, el error cometido al obtener otras métricas, como la permanencia promedio de los visitantes, es significativo. La causa de las impresiciones se debe en gran parte a que el algoritmo de seguimiento, en presencia de una única cámara y en condiciones no controladas, no es lo suficientemente robusto. Por tanto, en búsqueda de una solución con alto grado de generalidad, se necesitan algunos agregados para realizar un análisis preciso de la utilización de un espacio.

Tabla de contenidos

Capítulo 1

Introducción

1.1. Motivación

La visión artificial, o visión por computador, es una rama de la ingeniería que trata el problema de cómo pueden las computadoras, u otros sistemas, emular la percepción visual humana. Si bien es un área que involucra múltiples disciplinas, su principal objetivo puede resumirse como el de construir sistemas artificiales que sean capaces de lograr una interpretación de alto nivel en imágenes o videos, y que dichos sistemas puedan utilizarse para automatizar tareas que la visión humana es capaz de realizar.

Si bien los orígenes de la visión artificial se remontan a 1960 [44], en los últimos años ha ganado especial atención, gracias en gran parte al éxito logrado al combinar su aplicación con modelos de aprendizaje automático. El punto de quiebre puede considerarse el año 2012, cuando Alex Krizhevsky [25] introdujo el uso de redes neuronales convolucionales para la clasificación de imágenes, superando ampliamente el rendimiento de los modelos basados en técnicas tradicionales que conformaban el estado del arte del momento. Desde entonces, el crecimiento de los modelos de aprendizaje profundo llevó a que desplazaran rápidamente a los métodos tradicionales, basados en técnicas como la extracción manual de características, logrando mejorar año a año el desempeño en las competencias de reconocimiento visual hasta llegar a superar el rendimiento humano en dicha tarea [16].

La utilización del aprendizaje profundo en problemas de visión artificial ha sido adoptada exitosamente, no solo para la clasificación de imágenes, sino también para tareas como la detección de objetos (es decir, localizar qué partes

de una imagen corresponden a objetos de un determinado tipo), segmentación semántica (etiquetar cada pixel de una imagen) y en generación de leyendas a partir de una imagen (producir un texto descriptivo de su contenido) [40]. En la industria, sus más notables aplicaciones podrían resumirse en el desarrollo de vehículos de conducción automática, así como también en el análisis automático de radiografías médicas. De forma esperable, los emprendimientos relacionados a solucionar problemas a través de visión por computadora atraen cada año más financiamiento, teniendo una valoración promedio de $5,2 millones de dólares en el mercado estadounidense [3].

Entre las múltiples aplicaciones que la visión por computadora combinada con el aprendizaje profundo pueden tener, de las cuales solo hemos mencionado un subconjunto, nos resulta de particular interés centrarnos en los problemas de detección de personas en imágenes. Delimitando la problemática con aún más precisión, lo que nos interesa abordar es la localización de personas, y sus puntos corporales, a lo largo de una grabación de video. Un sistema que combine la detección de una persona con el seguimiento de la misma, lograría registrar los movimientos de todas las personas presentes en el campo de visión de una cámara a lo largo de una secuencia de video.

La mayoría de las soluciones propuestas para esto implican soporte de hardware adicional, como sensores infrarrojos o cámaras termales, lo que complejiza su implementación e implantación. La posibilidad de detectar personas en videos, sin mayor soporte que el de las cámaras que los registran, incrementa considerablemente sus posibilidades de adopción, y abre además un gran abanico de potenciales aplicaciones. Simplemente por mencionar algunos ejemplos, un sistema con estas características podría utilizarse para disparar alarmas ante intrusiones, realizar un conteo de clientes en una góndola de un supermercado, o integrarse con funciones de domótica en el hogar para disparar una acción cuando una persona ingresa a una habitación, o simplemente realiza un gesto de interés.

Si bien la mayoría de tiendas comerciales y espacios públicos en general tienen ya instalado un sistema de videocámaras de seguridad que guarda un registro de lo ocurrido en el lugar, la utilidad que se le da a esta información recabada está limitada solamente a funciones de vigilancia. Normalmente, los registros solo son consultados como evidencia cuando ocurre algún siniestro o hurto, y son descartados en el resto de situaciones cotidianas. Sin embargo, esta información de uso del lugar registrada guarda potencial no explorado

para comprender mejor la utilización que el público le da un espacio físico y cuales son sus patrones de comportamiento.

1.2. Propuesta

En este trabajo se propone entonces investigar la posibilidad de desarrollar un sistema que, aplicando técnicas recientes para la detección y seguimiento de personas, permita analizar un conjunto de secuencias de video tomadas por cámaras de seguridad para estudiar el comportamiento del público en el lugar. Así, en lugar de desperdiciar la información recabada constantemente por las videocámaras ya instaladas, se explota para el estudio de comportamientos que sean de interés para, por ejemplo, el dueño del establecimiento.

La variedad de aplicaciones incluso en este escenario delimitado son grandes. Podría por ejemplo resultar de interés contar el tiempo de espera de los pacientes en una sala de espera antes de ser atendidos, la cantidad de personas en la fila de una caja en una sucursal bancaria según el momento del día, el uso de los pasillos de una estación de trenes, etc.

Un sistema que ofrezca un conjunto de funcionalidades con un alto grado de generalidad, que permitan realizar un análisis del uso de un lugar independientemente de sus características, resultaría interesante. Un sistema así debería ofrecer una serie de herramientas con las que se puedan cubrir diferentes casos de uso, sin tener que desarrollar para cada uno una funcionalidad específica.

Nos interesa puntualmente investigar la viabilidad de un sistema con estas características, utilizando las herramientas de visión artificial disponibles hoy en día. Para esto evaluamos el rendimiento alcanzado por los métodos últimamente propuestos para la detección y el seguimiento de personas en condiciones no controladas. Los sistemas presentados en publicaciones académicas son, en su gran mayoría, evaluados en conjuntos de datos preexistentes, con el fin de poder comparar directamente su rendimiento frente a otros. Por el contrario, en este trabajo nos interesa evaluar el rendimiento alcanzado por el estado de arte en una situación no controlada, en la que las condiciones del ambiente, como la ubicación de las videocámaras o la disposición del mobiliario, no puedan ser modificadas a conveniencia.

1.3. Organización del documento

Este documento expone la investigación realizada, la solución propuesta y los resultados obtenidos por el prototipo realizado a lo largo de cinco capítulos, que comienzan en esta introducción.

En el Capítulo 2 se describen los fundamentos teóricos de los problemas de detección y seguimiento de personas y se presentan los métodos del estado del arte que serán utilizados en el sistema propuesto. La presentación de estos dos problemas que serán los pilares del sistema propuesto, es acompañada por un breve estudio del problema de análisis del público en espacios, sus utilidades y otras soluciones propuestas.

En el Capítulo 3 se presenta la solución propuesta para la implementación de un sistema que permita el análisis del comportamiento de personas en secuencias de video. Se describe su arquitectura general, cómo se integran en esta los métodos de detección y seguimiento utilizados, y se presentan sus funcionalidades y cómo estas pueden ser adaptadas a una variedad de escenarios posibles. Para evaluar la calidad de la solución propuesta, se realiza un prototipo de sus funcionalidades mínimas, que es presentado al final de la sección.

Los resultados obtenidos en la evaluación del prototipo son presentados en el Capítulo 4. En él se describe el conjunto de datos utilizado, sus características, y el procedimiento para su evaluación.

Finalmente, en el Capítulo 5 se presentan las conclusiones del trabajo, evaluando la viabilidad, y oportunidades de mejora del sistema a partir de los resultados obtenidos.

Capítulo 2

Fundamentos teóricos

2.1. Análisis de público en espacios

Si bien la popularidad del comercio online crece continuamente, las tiendas físicas aún dominan el mercado, abarcando la mayor parte de las ventas realizadas hoy en día. Estudios realizados muestran que el 91 % de las ventas son realizadas en tiendas físicas [21], y que el 82 % de los pertenecientes a la generación *Millenial* prefieren realizar sus compras en tiendas físicas antes que en tiendas online [10]. Aún así, el comercio online evoluciona rápidamente y la tendencia de su uso es al alza, por lo que es crucial para las tiendas físicas comprender las demandas de sus consumidores para poder ofrecerles un servicio personalizado y de calidad a las necesidades que diferentes perfiles puedan tener.

Para competir con las compras online, las tiendas físicas deben utilizar técnicas de marketing digital, con anuncios personalizados a cada cliente de manera de ofrecerle productos que sean de su interés. Estudios realizados indican que las interacciones digitales influencian el 49 % de las compras en sitio [30]. Pero para que esta publicidad dirigida resulte efectiva, necesita conocimiento de los hábitos de compra de los clientes y de los productos en los que han demostrado interés a lo largo del tiempo. Los sitios web cuentan con una gran ventaja en este sentido, ya que es posible registrar con alto nivel de detalle y precisión todas las interacciones del usuario con el sitio, seguir su comportamiento a través del histórico de páginas visitadas, el tiempo que se mantuvo en cada página, los horarios más frecuentes de visita, etc. Para poder desarro-

llar algo similar en tiendas, es necesario realizar una identificación precisa de cada cliente y mantener un seguimiento de su recorrido a lo largo de su estadía dentro de la tienda.

Para poder aprender los intereses de sus clientes, las tiendas[1] necesitan un sistema que permita el seguimiento e identificación de los clientes con gran precisión y en tiempo real. Es necesario ubicar a la persona con una precisión de metros para poder distinguir en que porción del espacio físico se encuentra. Además, ya que las personas normalmente permanecen en determinadas zonas solamente por unos segundos, es necesario también que los cambios en la ubicación sean detectados rápidamente por el sistema.

Para abordar este problema se han desarrollado sistemas basados principalmente en sensores, tales como localización por WiFi [9], Bluetooth [17] [43], cámaras estéreo [7] [50] y sensores térmicos [18]. El problema con este tipo de sistemas es que requieren un hardware especial de alto costo, no accesible a todas las tiendas, y que no justifica la inversión por la nula utilidad que la tienda puede darle más que para este sistema. En el caso de los sistemas que no requieren sensores especiales, como los de localización por WiFi y Bluetooth, el rango de error que manejan para el seguimiento los hace inadecuados para este tipo de escenarios.

Recientemente ha surgido otra tendencia que aborda el problema a resolver utilizando hardware ya existente en la gran mayoría de las tiendas: las cámaras de videovigilancia. El seguimiento de clientes en tiendas físicas en tiempo real a través del análisis de las secuencias de video tomadas por videocámaras ha empezado a vislumbrarse como una solución prometedora y viable. Compañías como Amazon, a través de su producto Amazon Go [1] y Standard Cognition [42] utilizan algoritmos privados para identificar clientes y realizar un seguimiento de sus acciones en la tienda a través de grabaciones recogidas por cámaras convenientemente colocadas en el lugar. Este tipo de servicios ofrecen calcular la cantidad de clientes y la distribución espacio-temporal de los mismos, pero requieren que el usuario descargue una aplicación propietaria.

Las soluciones de seguimiento basadas en cámaras de video se pueden cate-

[1] Con *tienda* en realidad nos referimos a cualquier local que reciba clientes, ya sea un supermercado, un restaurante, una tienda de ropa, etc.

gorizar en dos grandes grupos. Por un lado las que se basan en el reconocimiento facial, y por otro las que efectúan el seguimiento únicamente a través de la detección de la pose de las personas. El problema con los métodos basados en reconocimiento facial para el escenario de las tiendas, es que se enfrenta a algunos desafíos de privacidad, disponibilidad, y precisión. Por un lado porque los rostros de las personas son considerados información sensible y privada. Además, para entrenar un modelo con estas características sería necesario requerir a los clientes que voluntariamente ofrezcan una toma de su rostro a efectos de un posterior reconocimiento, lo cual puede no contar con la aceptación esperada. Por otro lado, el reconocimiento facial a través de cámaras de videovigilancia puede no funcionar correctamente. Principalmente debido al ángulo con el que estas estén posicionadas, y porque la calidad de la misma puede no ser la óptima, que representa una de las principales vulnerabilidades de este tipo de soluciones.

Las soluciones que intentan resolver este problema solamente mediante la detección de la pose de las personas a través de cámaras de videovigilancia parecen ser entonces un enfoque adecuado para las dos partes: los dueños de las tiendas, que no requieren de inversión adicional en hardware, y los clientes, que no tienen que prestar ninguna información adicional a los efectos de su identificación. Aunque también presentan una desventaja con respecto a los métodos de reconocimiento facial, y es que, como no guardan información sobre la apariencia de los clientes, no son capaces de reconocerlos en visitas sucesivas. Sin embargo esto podría subsanarse en algunos casos, si las personas realizan alguna transacción en la tienda que las identifique, como por ejemplo efectuar una compra.

Independientemente de esta limitante, será el enfoque utilizado en nuestro trabajo debido a su simplicidad de adopción. En las siguientes secciones de este capítulo nos centraremos en este tipo de soluciones, detallando su base teórica, presentando ejemplos de métodos que resuelven este tipo de problemas, y los que se eligieron en este trabajo.

2.2. Detectores de personas

La detección y el reconocimiento de figuras en imágenes es objeto de estudio actual, y se ubica entre los desafíos más importantes en el campo de la Visión por Computadora. Puntualmente el problema de detectar objetos de

una determinada categoría, como por ejemplo personas, o automóviles, presenta múltiples dificultades ya que estos objetos pueden variar considerablemente en apariencia. Estas variaciones pueden provenir desde fuentes tan diversas como la iluminación en la imagen, o características visuales propias de cada tipo de objeto. Por ejemplo las personas pueden vestir de formas muy diferentes, o tomar poses diversas, mientras que los automóviles pueden presentarse en muchas formas y colores.

En esta sección introduciremos los conceptos teóricos de base para nuestro proyecto con respecto a la detección de personas. Primero introduciremos sobre el modelo de partes deformables, una de las primeras metodologías con las que se abordó este problema, y que sirvió de base y motivación para desarrollar técnicas más sofisticadas. Luego detallaremos el funcionamiento de OpenPose, la librería utilizada en nuestro trabajo, que como veremos a continuación, se encarga de reconocer poses humanas a través de la detección de puntos anatómicos del cuerpo.

2.2.1. Deformable Parts Model

La base de este tipo de modelos parte de un marco de trabajo conocido como estructuras descriptivas (Pictorial Structures en inglés), introducida por Fischler y Elschlager [15] casi 50 años atrás.

En este framework se representa cada objeto como una colección de partes, estructuradas en una configuración deformable. Un objeto se modela utilizando varias partes que capturan las características locales de un subconjunto del objeto, mientras que la estructura deformable se representa mediante vínculos extensibles (la forma más intuitiva de pensarlo es imaginando dichos vínculos como resortes), entre determinados pares de partes.

Al momento de buscar un determinado objeto en una imagen, se ubica el objeto sobre todas las posiciones posibles en la imagen, contrayendo y extendiendo los vínculos entre las partes, hasta obtener el mejor alineamiento entre el objeto buscado y la imagen de referencia. Luego se evalúa cada posicionamiento según cuan buena la correspondencia haya sido, y cuánto se haya tenido que deformar la configuración interna del objeto (es decir, extender o contraer los vínculos entre sus partes), para lograrlo.

Formalmente el algoritmo utilizado en este modelo se define como sigue. Suponiendo que un objeto está compuesto por p partes, definimos la variable

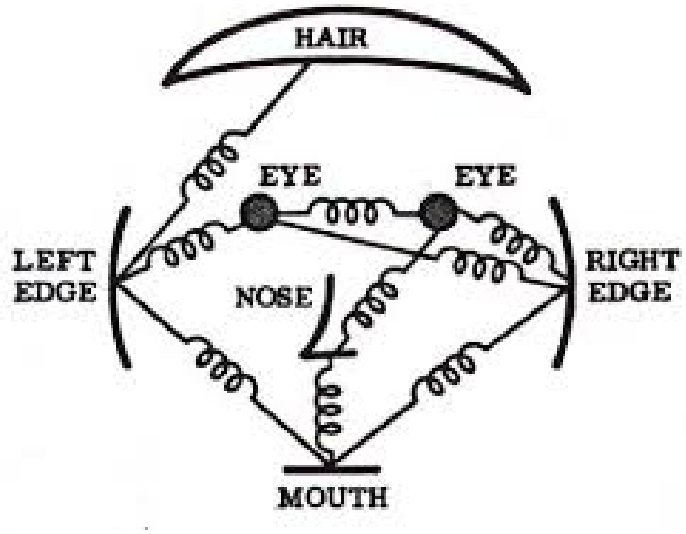

Figura 2.1: Imagen ilustrativa incluida en el artículo original Fischler y Elschlager [15]

x_i para $1 \leq i \leq p$ que puede tomar valores entre todas las posiciones posibles de la imagen de referencia. Es decir, x_i es la posición de la i-ésima componente. Suponemos también que existe un mecanismo, que a los efectos de presentar el modelo tomaremos como caja negra, que para la posición x_i de la componente i-ésima devuelve un valor numérico $l_i(x_i)$ que indica que tan bien se ajusta la componente i-ésima en la posición x_i de la imagen de referencia. A menor valor de $l_i(x_i)$, mejor es la correspondencia.

La idea es que $l_i(x_i)$ asigne valores independiente de cualquier otro conocimiento que se pueda tener sobre la ubicación del resto de las partes en la imagen de referencia. Es decir que se espera que $l_i(x_i)$ sea puramente local.

Cada ubicación de la imagen de referencia puede ser asociada con un vector de dos dimensiones (por ejemplo los componentes pueden ser fila y columna de la ubicación en la imagen). En este caso, $x_i - x_j$ es un vector que apunta desde x_j a x_i. Se puede entonces definir $g_{ij}(x_i, x_j) = g_{ij}(x_i - x_j)$ como el costo asociado con el vínculo que une las partes i y j. Si no hay un vínculo que una estas componentes, entonces su costo asociado, g_{ij} es 0. Si fijamos $g_{ij}(x_i, x_j) = l_i(x_i)$ cuando $i = j$, y decimos que $X_i = x_1, x_2, ..., xi$, entonces el costo total de ubicar p componentes en ubicaciones X_p es $G(X_p)$.

$$G(X_p) = \sum_{i=1}^{p} \sum_{j=1}^{i} g_{ij}(x_i, x_j) \tag{2.1}$$

Esta misma expresión puede ser escrita como

$$G(X_p) = \sum_{i=1}^{p} h_i(X_i) \tag{2.2}$$

donde $h_i(X_i) = \sum_{j=1}^{i} g_{ij}(x_i, x_j)$ que puede ser interpretado como el costo de ubicar la i-ésima componente en la ubicación x_i, dado que las $i-1$ componentes anteriores hayan sido ubicadas en las ubicaciones especificadas por X_i.

Los algoritmos computacionales que resuelven este problema están basados mayoritariamente en programación dinámica, y continúan siendo objeto de estudio. En particular, presentaremos como ejemplo el trabajo realizado por el autor P. Felzenszwalb [35], en el año 2013, que combinó este modelo con técnicas de aprendizaje automático para obtener resultados alineados (y en algunos casos superiores) al estado del arte. Como resultado de la clasificación, el modelo envuelve cada objeto reconocido en cajas delimitantes rectangulares etiquetadas según la clase del objeto. El modelo se evaluó utilizando el dataset PASCAL VOC 2010, ampliamente reconocido como un desafío complejo en el campo de la detección de objetos en imágenes [29]. El dataset contiene cajas delimitantes para 20 clases de objetos, que ofician de ground-truth para las pruebas. El autor aclara que una caja delimitante para un determinado objeto reconocido, como resultado del modelo evaluado, es considerada correcta si se superpone en más de un 50 % con la caja provista como ground truth.

	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV
Base[a]	47.2	50.8	8.6	12.2	32.2	48.9	44.4	28.1	13.6	22.7	11.3	17.4	40.4	47.7	44.4	7.6	30.0	17.3	38.5	34.3
BB[b]	48.7	52.0	8.9	12.9	32.9	51.4	47.1	29.0	13.8	23.0	11.1	17.6	42.1	49.3	45.2	7.4	30.8	17.1	40.6	35.1
Context[c]	52.4	54.3	13.0	15.9	35.1	54.2	49.1	31.8	15.5	26.2	13.5	21.5	45.4	51.6	47.5	9.1	35.1	19.4	46.6	38.0
Best[d]	58.4	55.3	19.2	21.0	**35.1**	55.5	**49.1**	47.7	20.0	31.5	27.7	37.2	51.9	56.3	**47.5**	13.0	37.8	33.0	50.3	41.9

[a]Score of our base system.
[b]The system with bounding box prediction.
[c]The final system with context rescoring.
[d]The highest score over all systems entered into the 2010 competition (bolded numbers indicate that our system obtained the highest score).

Figura 2.2: Resultados del trabajo de P. Felzenswalb sobre Pascal VOC 2010.[35]

2.2.2. OpenPose

La estimación de la pose humana en dos dimensiones, es decir el problema de localizar los puntos anatómicos (a los que nos referiremos como "partes") de una persona, se ha abordado históricamente ubicando las partes corporales de cada individuo. Pero inferir la pose de múltiples personas en imágenes con-

lleva un conjunto más amplio de desafíos. En primer lugar, cada imagen puede contener un número desconocido de personas que pueden presentarse en cualquier posición y tamaño. En segundo lugar, las interacciones entre las personas inducen interferencias espaciales complejas debido al contacto, la oclusión y la articulación de las extremidades, haciendo que la asociación entre las partes resulte muy complicada. Y en tercer lugar, el tiempo de ejecución tiende a crecer con el número de personas en la imagen, haciendo que la performance en tiempo real se vuelva todo un desafío.

Un enfoque utilizado para esto, como por ejemplo el del autor P. F. Felzenszwalb basado en estructuras descriptivas [14], es emplear un detector de personas, y estimar la pose de cada individuo a partir de las detecciones. Si bien estos enfoques lograron buenos rendimientos, sufrían de un problema de base: si el detector de personas falla (escenario posible debido a las complejidades mencionadas en el párrafo anterior), no se cuenta con recursos para recuperarse. Además, el tiempo de ejecución de estos enfoques top-down, es proporcional al número de personas: para cada detección se ejecuta un estimador de pose individual, y cuántas más personas haya, mayor será el costo computacional.

A diferencia de esto, el método detrás de OpenPose implica un enfoque bottom-up. Como se describirá a continuación, resulta más atractivo al ofrecer más robustez ante fallas en las detecciones, y presentar el potencial de desacoplar la complejidad en el tiempo de ejecución del número de personas en la imagen.

A continuación se describirá el método utilizado por OpenPose, y el resultado obtenido al ser evaluado en COCO 2016 Key-Points Challenge y MPII Multi-Person benchmark.

El texto que sigue está basado en la publicación de 2017 de los autores Zhe Cao, Tomas Simon, Shih-En y Wei Yaser Sheikh [8] de la Carnegie Mellon University que, sumado a Gines Hidalgo y Hanbyul Joo, son los autores de OpenPose. [1]

Método

El sistema toma, como entrada, una imagen a color de tamaño w x h y produce, como salida, las ubicaciones en dos dimensiones de los puntos anatómicos

[1] El código y más información sobre OpenPose pueden encontrarse en su sitio en GitHub - https://github.com/CMU-Perceptual-Computing-Lab/openpose

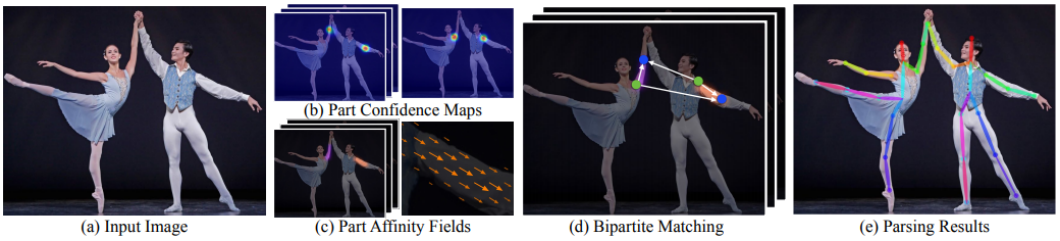

Figura 2.3: Resumen de la metodología detrás de OpenPose [8]

de cada persona en la imagen. Para lograr esto, en primer lugar una red neuronal feed-forward predice simultáneamente un conjunto de mapas de confianza **S** de dos dimensiones para cada parte de cada cuerpo, y un conjunto **L** de campos vectoriales de dos dimensiones que representan la afinidad de las partes, que codifica la asociación entre ellas. El conjunto $\mathbf{S} = (S_1, S_2, ..., S_J)$ contiene J mapas de confianza, uno para cada parte, donde $S_j \in \Re^{wxh}$. El conjunto **L** = (L_1,L_2,...,L_C) contiene C campos vectoriales, uno por extremidad[1], donde el conjunto $\mathbf{S} = (L_1, L_2, ..., L_C) \in \Re^{wxhx2}$. Cada ubicación de la imagen en L_c genera un vector de dos dimensiones. Finalmente, los mapas de confianza y los campos de afinidad son procesados por un algoritmo de tipo Greedy, para devolver los puntos anatómicos en dos dimensiones para todas las personas en la imagen. Un resumen ilustrativo puede verse en la Figura 2.3.

Detección y Asociación simultáneas

La arquitectura de OpenPose está ilustrada en la siguiente figura.

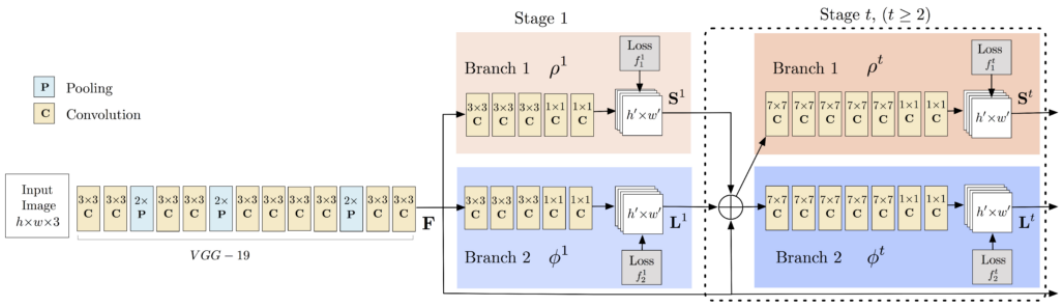

Figura 2.4: Arquitectura de OpenPose [8]

La red está dividida en dos ramas: la rama superior, en color beige, predice los mapas de confianza, mientras que la rama inferior, en azul, es la que predice

[1]Nos referimos a pares de partes como extremidades, más allá de que algunos pares no se correspondan con extremidades del cuerpo humano

los campos vectoriales de afinidad entre las partes. Cada rama lleva a cabo un método de predicción iterativa, que sigue lo presentado por Wei et al. [39], y refina la predicción en etapas sucesivas $t \in \{1, ..., T\}$, con supervisión inmediata tras cada etapa.

La imagen es primero analizada por una red neuronal convolucional, diseñada de acuerdo al modelo VGG-19 [20], que generan un mapa de características $\mathbf{F}$ que se reciben como entrada para la primera etapa de cada rama. En la primera etapa, la produce un conjunto de mapas de confianza $\mathbf{S}^1 = \rho^1(\mathbf{F})$ y un conjunto de campos de afinidad de partes $\mathbf{L}^1 = \phi^1(\mathbf{F})$ donde $\rho^1 y \phi^1$ representan las CNNs en la etapa 1. En cada etapa siguiente, las predicciones de cada rama para la etapa anterior, más el mapa de características original de la imagen $\mathbf{F}$, son concatenados y utilizados para producir predicciones refinadas:

$$\mathbf{S}^t = \rho^t(\mathbf{F}, \mathbf{S}^{t-1}, \mathbf{L}^{t-1}), \forall t \geq 2 \tag{2.3}$$

$$\mathbf{L}^t = \phi^t(\mathbf{F}, \mathbf{S}^{t-1}, \mathbf{L}^{t-1}), \forall t \geq 2 \tag{2.4}$$

donde ρ^t y ϕ^t son las CNNs en la etapa t.

Para guiar el refinamiento iterativo de las predicciones, se aplican dos funciones de pérdida, tipo L2[1], entre las predicciones estimadas y los datos anotados.

Las funciones de pérdida se definen como:

$$\mathbf{f}_S^t = \sum_{j=1}^{J} \sum_{\mathbf{P}} \mathbf{W}(\mathbf{p}).\|\mathbf{S}_j^t(p) - \mathbf{S}_j^*(p)\|_2^2 \tag{2.5}$$

$$\mathbf{f}_L^t = \sum_{c=1}^{C} \sum_{\mathbf{P}} \mathbf{W}(\mathbf{p}).\|\mathbf{L}_c^t(p) - \mathbf{L}_c^*(p)\|_2^2 \tag{2.6}$$

donde $\mathbf{S}_j^*$ y $\mathbf{L}_c^*$ son, respectivamente, el mapa de confianza y el campo vectorial de afinidad de partes generados a partir de los datos etiquetados, y $\mathbf{W}$ es una máscara binaria tal que $\mathbf{W}(\mathbf{p}) = 0$ si no se cuenta con una etiqueta para la posición p de la imagen. Globalmente, queda definida como:

$$f = \sum_{t=1}^{T} (\mathbf{f}_S^t + \mathbf{f}_L^t) \tag{2.7}$$

[1]En el campo de Aprendizaje Automático se define una función de pérdida de tipo L2 a la función que cuantifica el error cuadrático cometido durante la clasificación de un objeto

Mapas de confianza para la detección de partes

Para evaluar f_S en la ecuación (2.7), se generan mapas de confianza $\mathbf{S}^*$ a partir de los datos anatómicos etiquetados. Cada mapa de confianza es una representación en dos dimensiones de la convicción de que una determinada parte del cuerpo esté ubicada en cada pixel de la imagen. Idealmente, si una única persona está presente en la imagen, entonces debería existir un único pico para cada mapa de confianza (si dicha parte está visible); en caso que la imagen contenga más de una persona, entonces debería haber un pico para cada parte visible j para cada persona k.

Se generan primero individualmente cada mapa de confianza $S^*_{j,k}$ para cada persona k. Si $x_{j,k} \in \Re^2$ es la ubicación etiquetada en la imagen de la parte j para la persona k, entonces el valor de $S^*_{j,k}$ para la ubicación $\mathbf{p} \in \Re^2$ queda definida como:

$$\mathbf{S}^*_{j,k}(p) = exp(-\frac{\|\mathbf{p} - \mathbf{x}_{j,k}\|^2_2)}{\sigma^2}) \tag{2.8}$$

donde σ controla la extensión del pico. Luego el mapa de confianza final se generan maximizando los mapas individuales:

$$\mathbf{S}^*_j(p) = \max_k \mathbf{S}^*_{j,k}(p) \tag{2.9}$$

Campos de afinidad para la asociación entre partes

Dado un conjunto de detecciones de partes del cuerpo, la pregunta que surge es: ¿cómo es posible ensamblarlas para formar el cuerpo completo de un número desconocido de personas? Para esto, los autores de OpenPose introducen una funcionalidad innovadora que bautizaron como *campos de afinidad entre partes* (abreviado como **PAFs**, Part Affinity Fields en los textos originales)

Como se ilustra en la figura anterior, en un campo de afinidad entre partes se representa, para cada pixel perteneciente a cada extremidad, un vector que codifica la dirección desde una parte de la extremidad hacia la otra. Cada tipo de extremidad tiene un campo de afinidad que une sus dos partes del cuerpo asociadas.

Como ejemplo ilustrativo, en la imagen anterior se visualiza una extremidad. Los puntos $x_{j_1,k}$ y $x_{j_2,k}$ son las posiciones de las partes del cuerpo j_1 y j_2 de la extremidad c de la persona k en una imagen. Si un punto $\mathbf{p}$ está posicionado sobre una extremidad, entonces el valor de $\mathbf{L}^*_{c,k}(\mathbf{p})$ es un vector unitario

Figura 2.5: Part Affinity Fields [8]

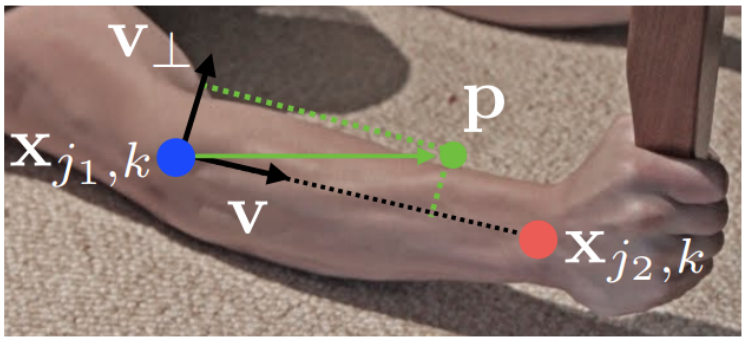

Figura 2.6: Ejemplo de la asociación entre partes aplicada a un brazo [8]

que apunta desde j_1 a j_2. Para el resto de los puntos (que no recaen sobre la extremidad), el vector toma valor 0.

Para evaluar $f_{\mathbf{L}}$ en la ecuación (2.7) se define, durante la etapa de entrenamiento, el campo de afinidad entre partes según los datos etiquetados $\mathbf{L}^*_{c,k}(\mathbf{p})$ como:

$$\mathbf{L}^*_{c,k}(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{si } \mathbf{p} \text{ está en la extremidad c,k} \\ \mathbf{0} & \text{en otro caso} \end{cases} \tag{2.10}$$

En la ecuación anterior, $\mathbf{v} = \frac{x_{j_2,k} - x_{j_1,k}}{\|x_{j_2,k} - x_{j_1,k}\|_2}$ es el vector unitario en la dirección de la extremidad. El conjunto de puntos que pertenecen a la extremidad comprende aquellos puntos que estén entre un umbral de distancia con respecto al segmento de línea, es decir, los puntos $\mathbf{p}$ tal que:

$$0 \leq \mathbf{v} \cdot (\mathbf{p} - \mathbf{x}_{j_1,k}) \leq l_{c,k} \quad y \quad |\mathbf{v}_\perp \cdot (\mathbf{p} - \mathbf{x}_{j_1,k})| \leq \sigma_1$$

donde σ_1 es el ancho de la extremidad, medida como distancia en pixeles, el largo de la extremidad es $l_{c,k} = \|x_{j_2,k} - x_{j_1,k}\|_2$, y $\mathbf{v}_\perp$ es el vector perpendicular

a $\mathbf{v}$.

El campo de afinidad entre partes según los datos etiquetados (*grountruth*) se genera promediando los campos de todas las personas en la imagen:

$$\mathbf{L}_c^*(\mathbf{p}) = \frac{1}{n_c(\mathbf{p})} \sum_k \mathbf{L}_{c,k}^*(\mathbf{p}) \tag{2.11}$$

donde $n_c(\mathbf{p})$ es el número de vectores distintos de zero en el punto $\mathbf{p}$ para las k personas, es decir el promedio de pixeles donde se superponen extremidades de diferentes personas.

Para hallar los campos de afinidad predichos, en la instancia de testing, se mide la asociación entre las detecciones de las partes candidatas a través del cálculo de la integral de línea sobre el campo vectorial correspondiente, a lo largo del segmento de recta que conecta las partes candidatas. En otras palabras, se mide el alineamiento entre el campo de afinidad de partes predicho y la extremidad que se tiene como candidata para ser formada al conectar las partes del cuerpo detectadas. En términos matemáticos, para dos partes candidatas d_{j_1} y d_{j_2}, se muestrea el campo de afinidad predicho L_c a lo largo del segmento de recta para medir el nivel de confianza en la asociación:

$$E = \int_{u=0}^{u=1} \mathbf{L}_c(\mathbf{p}(u)) \cdot \frac{d_{j2} - d_{j1}}{\|d_{j2} - d_{j1}\|_2} du \tag{2.12}$$

donde $\mathbf{p}$(u) interpola la posición de las dos partes d_{j_1} y d_{j_2}.

$$\mathbf{p}(u) = (1-u)\mathbf{d}_{j_1} + u\mathbf{d}_{j_2} \tag{2.13}$$

Parsing multi-persona utilizando campos de afinidad entre partes

Definidos los mapas de confianza y los campos de afinidad entre partes, resta ver como se unen los conceptos. La última parte del algoritmo ejecuta una supresión no máxima sobre los mapas de confianza para obtener un conjunto discreto de posiciones candidatas para las partes. Para una misma parte se pueden tener varios candidatos, ya sea porque más de una persona está presente en la imagen o por los falsos positivos detectados. Se pondera cada extremidad candidata a través de la integral definida en la ecuación (2.12). El problema de encontrar el parsing óptimo se corresponde con un problema de emparejamiento en K dimensiones calificado como NP-Hard [47]. Para abordar este problema, los autores presentaron una relajación greedy que produce

en forma constante emparejamientos de alta calidad.

Formalmente se obtiene primero un conjunto de partes candidatas $D_J = \{\mathbf{d}_j^m : con \quad j \in \{1...J\}, m \in \{1...N_J\}\}$, siendo N_j el número de candidatos para la parte j, y $\mathbf{d}_j^m \in \Re^2$ la posición de la detección candidata m-ésima de la parte j. Estas detecciones candidatas de partes tienen que ser todavía asociadas con otras partes de la misma persona. Es decir, se necesita encontrar los pares de partes que formen efectivamente extremidades. Para eso se define una variable $z_{j_1 j_2}^{mn} \in \{0, 1\}$ que indica si dos detecciones candidatas $d_{j_1}^m$ y $d_{j_2}^m$ están conectadas, con el objetivo de encontrar la asignación óptima para todo el conjunto de posibles conexiones: $Z = \{z_{j_1 j_2}^{mn} \quad : \quad con j_1, j_2 \in \{1...J\}, m \in \{1...N_{j_1}\}, n \in \{1...N_{j_2}\}\}$.

Si consideramos un par individual de partes j_1 y j_2 para la c-ésima extremidad, hallar la asociación óptima se reduce a un problema de emparejamiento de grafos bipartitos con peso máximo [47]. En este caso, los nodos del grafo son detecciones candidatas de partes del cuerpo D_{j_1} y D_{j_2}, y los enlaces son todas las posibles conexiones entre pares de detecciones candidatas. Los pesos de cada enlace son asignados según la ecuación (2.12). Un emparejamiento en un grafo bipartito es un subconjunto de enlaces seleccionados de tal forma que ningún par de enlaces compartan un vértice. El objetivo es entonces encontrar un emparejamiento de peso máximo:

$$\max_{Z_c} E_c = \max_{Z_c} \sum_{m \in D_{j_1}} \sum_{n \in D_{j_2}} E_{mn} \cdot z_{j_1 j_2}^{mn} \tag{2.14}$$

$$s.t. \quad \forall m \in D_{j_1}, \sum_{n \in D_{j_2}} z_{j_1 j_2}^{mn} \leq 1 \tag{2.15}$$

$$\forall n \in D_{j_2}, \sum_{m \in D_{j_1}} z_{j_1 j_2}^{mn} \leq 1 \tag{2.16}$$

donde E_c es el peso global del emparejamiento desde la extremidad tipo c, Z_c es el subconjunto de Z para la extremidad tipo c, E_{mn} es la afinidad entre las partes $\mathbf{d}_{j_1}^m$ y $\mathbf{d}_{j_2}^n$ definidas en la ecuación (2.12). Las restricciones (2.15) y (2.16) imponen que dos enlaces no compartan un vértice, es decir, que dos extremidades del mismo tipo (por ejemplo antebrazo izquierdo) compartan una parte.

Como se mencionó anteriormente, determinar Z es un problema NP Hard para el cual existen múltiples relajaciones [47]. Los autores utilizan dos, espe-

cializadas en el dominio de la detección de objetos. Primero se elige una cantidad minimal de enlaces para obtener un árbol de recubrimiento que simboliza el esqueleto de una pose humana, en lugar de utilizar el grafo completo. En segundo lugar, se descompone el problema de emparejamiento en un conjunto de sub-problemas de emparejamiento bipartito y se determina el emparejamiento en vértices adyacentes en forma independiente.

Aplicando estas dos relajaciones, la optimización se descompone como:

$$\max_{Z} E = \sum_{c=1}^{C} \max_{Z_c} E_c \tag{2.17}$$

Con esto se obtienen entonces las conexiones entre extremidades para cada tipo, en forma independiente utilizando las ecuaciones (2.12), (2.15) y (2.16). Con esto es posible ensamblar las conexiones que comparten las mismas detecciones de partes candidatas para formar poses completas de múltiples personas. Como se detallará en la siguiente sección, los autores logran demostrar que el método aproxima correctamente la solución global y su costo computacional es órdenes más rápido que los métodos basados en optimización utilizando un grafo completo.

Evaluación del método al momento de su lanzamiento

Para evaluar la eficacia del método se utilizaron dos de los datasets que ofician de punto de referencia en la estimación de la pose humana: MPII multi-person benchmark [2] y COCO 2016 keypoints challenge [45]. Estos dos datasets compilan miles de imágenes en diversos escenarios del mundo real, como ser espacios densamente poblados, personas en diferentes escalas, oclusiones y contactos. Como se verá a continuación, el resultado obtenido en estos desafíos fue realmente bueno. OpenPose definió el estado del arte en el COCO 2016 keypoints challenge, mientras que en el MPII multi-person benchmark superó ampliamente el estado del arte establecido previamente.

Para el **MPII multi-person benchmark**, la Figura 2.7 resume el resultado obtenido en comparación al resto de los métodos en el momento de la evaluación[1].

[1]Los resultados vigentes del desafío, que incluye nuevos métodos presentados posteriormente pueden consultarse en http://human-pose.mpi-inf.mpg.de/#results

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	**mAP**	s/image
Subset of 288 images									
Deepcut	73.4	71.8	57.9	39.9	56.7	44.0	32.0	54.1	57995
Iqbal et al.	70.0	65.2	56.4	46.1	52.7	47.9	44.5	54.7	10
DeeperCut	87.9	84.0	71.9	63.9	68.8	63.8	58.1	71.2	230
Ours	**93.7**	**91.4**	**81.4**	**72.5**	**77.7**	**73.0**	**68.1**	**79.7**	**0.005**
Full testing set									
DeeperCut	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Iqbal et al.	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
Ours (one scale)	89.0	84.9	74.9	64.2	71.0	65.6	58.1	72.5	0.005
Ours	**91.2**	**87.6**	**77.7**	**66.8**	**75.4**	**68.9**	**61.7**	**75.6**	**0.005**

Figura 2.7: Resultados de OpenPose enl MPII Multi-Person benchmark [8]

Los resultados están agrupados en dos conjuntos: los obtenidos sobre el dataset completo (tabla inferior) y los obtenidos sobre un subconjunto de 288 imágenes (tabla superior).

De acuerdo a los criterios establecidas por los creadores del dataset [2], una detección se considera acertada si la localización de sus puntos coincide en más de un determinado umbral con respecto al ground-truth. El umbral se definió como el 50 % de la longitud del segmento de la cabeza. Las columnas de cada tabla indican el porcentaje de acierto en las detecciones de las siguientes partes del cuerpo respectivamente: cara, hombro, codo, muñeca, cadera, rodilla y tobillo. El promedio del acierto en las detecciones (mean Average Precision, **mAP**) se muestra en la última columna.

El dataset **COCO** está compuesto por más de 100 mil instancias de personas etiquetadas con más de 1 millón de partes del cuerpo etiquetadas (keypoints). El conjunto de pruebas contiene tres subconjuntos: *test-challenge*, *test-dev* y *test-standard*, que contienen aproximadamente 20 mil imágenes cada uno.

La evaluación del COCO challenge está basado en una métrica llamada Object Keypoint Similarity (**OKS**), introducida por los creadores del dataset[1]. La precisión en las detecciones de los algoritmos se miden con base en ésta métrica, que vendría a ser un análogo a la intersección sobre la unión (IoU, Intersection Over Union) para los detectores de objetos. Más adelante en el documento nos expandimos más sobre este último concepto.

Los resultados obtenidos para este desafío se pueden ver en la Figura 2.8, obtenida también del artículo de los autores de OpenPose:

[1]Se pueden encontrar más detalles sobre las métricas de evaluación de COCO Keypoints Challenge en http://cocodataset.org/#keypoints-eval

Team	**AP**	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Test-challenge					
Ours	**60.5**	**83.4**	**66.4**	55.1	**68.1**
G-RMI [19]	59.8	81.0	65.1	**56.7**	66.7
DL-61	53.3	75.1	48.5	55.5	54.8
R4D	49.7	74.3	54.5	45.6	55.6
Test-dev					
Ours	**61.8**	**84.9**	**67.5**	57.1	**68.2**
G-RMI [19]	60.5	82.2	66.2	**57.6**	66.6
DL-61	54.4	75.3	50.9	58.3	54.3
R4D	51.4	75.0	55.9	47.4	56.7

Figura 2.8: Resultados de OpenPose en desafío COCO 2016 [8]

2.2.3. Estado del arte

En lo que respecta a la detección de la pose de múltiples personas en imágenes, los enfoques vigentes pueden ser clasificados a grandes rasgos en dos grupos:

- **Top-down**, que comprende a los métodos que primero realizan la detección de las personas, y luego aplican un estimador de pose individual para cada persona
- **Bottom-up**, que incluye a los métodos que realizan primero una detección de partes en toda la imagen, y luego agrupan esas partes para formar la pose de una persona

OpenPose, como se describió anteriormente en este capítulo, pertenece al segundo grupo. Sin embargo, ambos presentan ventajas y desventajas, por lo que la tendencia de los métodos desarrollados recientemente no recae solamente sobre uno de ellos. A continuación vamos a realizar una breve mención a otras alternativas que respaldan esta afirmación. En particular, vamos a hacer un repaso sobre los tres métodos que se presentaron al desafío MPII multi-person benchmark [2], y obtuvieron resultados superiores a los de OpenPose. Los resultados completos pueden verse en la imagen debajo:

Associative embedding

A. Newell, Z. Huang, y J. Deng [33] (*Newell et al* en la Figura 2.9) desarrollaron un método que realiza la detección de partes y el agrupado de las mismas en una sola etapa. El alcance es más general que el de la estimación

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	mAP
Iqbal&Gall, ECCVw'16	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1
Insafutdinov et al., ECCV'16*	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5
Insafutdinov et al., arXiv'16a*	89.4	84.5	70.4	59.3	68.9	62.7	54.6	70.0
Levinkov et al., CVPR'17	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6
Varadarajan et al., arXiv'17	92.1	85.9	72.9	61.7	72.0	64.6	56.6	72.2
Insafutdinov et al., CVPR'17	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3
Cao et al., CVPR'17	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6
Fang et al., arXiv'16	88.4	86.5	78.6	**70.4**	74.4	**73.0**	**65.8**	76.7
Newell et al., NIPS'17	**92.1**	89.3	78.9	69.8	76.2	71.6	64.7	**77.5**
Fieraru et al., CVPRw'18	91.8	**89.5**	**80.4**	69.6	**77.3**	71.7	65.5	**78.0**

Figura 2.9: Resultados del desafío MPII multi-person benchmark, modalidad conjunto completo. *Cao et al* identifica a OpenPose. **Fuente**: Sitio web de MPII multi-person benchmark[2]

de poses de personas, ya que los autores también lo aplican a la segmentación de instancias (u objetos). El método, llamado *associative embedding* (incrustamiento asociativo, en su traducción literal al español), consiste en introducir una etiqueta para cada parte detectada que sirve de identificador del grupo al que pertenece. Para la estimación de poses, el método se basa en una red neuronal convolucional especial (bautizada como modelo *reloj de arena apilado* - *Stacked Hourglass*), que produce un mapa de calor de detecciones y otro de etiquetas, y luego agrupa las partes con misma etiqueta formando personas. En esencia es un método *bottom-up*, con la sutil diferencia de que la detección y el agrupado se realizan en simultáneo. En su evaluación en el MPII (modalidad conjunto completo), el método obtuvo una puntuación de 77.5 mAP, que definió el estado del arte.

RMPE: Regional Multi-Person Pose Estimation

Un ejemplo de un enfoque top-down es el *Regional Multi-Person Pose Estimation (RMPE)* [13], desarrollado por H. Fang, S. Xie, Y. Tai y C. Lu (*Fang et al* en la Figura 2.9). El método consiste en detectar y generar poses precisas incluso en presencia de detecciones de personas (en realidad en forma de cajas delimitantes) defectuosas. El corazón del método está formado también por una red neuronal convolucional especial (*SSTN* por sus iniciales en inglés). En términos generales, esta recibe una imagen con cajas delimitantes propuestas para las personas presentes, delimita individualmente a las personas con mayor precisión, y luego genera las poses de cada una de ellas. El método ofrece además un marco de trabajo genérico e integrable con distintos detectores de

personas y generadores de poses individuales. En su evaluación en el MPII obtuvieron una puntuación de 76.7 mAP, superando ampliamente el estado del arte.

PoseRefiner

El método desarrollado por M. Fieraru, A. Khoreva, L. Pischulin y B. Schiele [31] es el más reciente del listado, y es, en cambio, un refinador de poses (de ahí su nombre, *PoseRefiner*). Motivados por el aún alto porcentaje de error cometido por los métodos vigentes al estimar las poses en casos desafiantes como oclusión o proximidad entre personas de aspectos similares, los autores desarrollaron un método que toma una imagen RGB y una estimación de las poses en ella, para devolver un resultado refinado. Esto implica que el método está diseñado para trabajar como pos-procesamiento al aplicarse sobre un estimador de poses, por lo que representa un complemento para los enfoques presentados previamente. Independientemente, como se ve en la Figura 2.9 (identificado como *Fieraru et al*), y en el resto de los desafíos en los que fue evaluado el método, presenta resultados que expanden los límites del estado del arte en la estimación de poses.

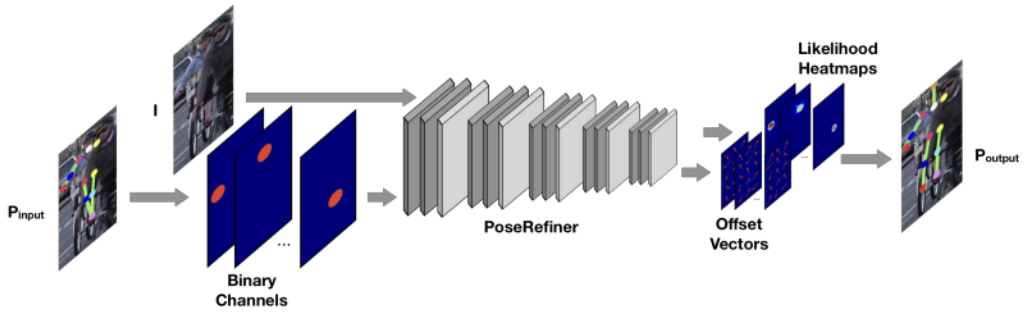

Figura 2.10: Resumen del método aplicado por PoseRefiner. **Fuente**: Learning to refine Human Pose estimation [31]

Como se ve en la Figura 2.10, el método toma una imagen I y una estimación de pose P_{input}. La pose es codificada en n canales binarios, siendo n la cantidad de partes, que son apilados con la imagen para formar la entrada de una red neuronal convolucional. Ésta red tiene la capacidad de predecir mapas de calor de similaridad (*Likelihood heatmaps*)para cada tipo de parte, en conjunto con vectores de desplazamiento (*Offset vectors*) para alinear la posición

de las partes entre los mapas de calor y la imagen I. Como salida, retorna una pose P_{output} que representa una refinación de P_{input}.

2.3. Seguimiento de objetos

2.3.1. Definición del problema

El problema de seguir la ubicación de múltiples objetos a través de un video es llamado Seguimiento de Objetos Múltiples (MOT por las siglas de su nombre inglés, *Multiple Object Tracking*). En adición a la detección de objetos, es necesario conectar las detecciones obtenidas en distintos cuadros del video asignándoles un etiquetado consistente (ID) que mantenga las identidades de cada una. El seguimiento realizado debe ser capaz de resolver desafíos que pueden aparecer cuando los objetos son ocluidos parcial o totalmente, o cuando dejan temporalmente el campo de visión y luego reingresan, escenarios en los que idealmente deberían mantener sus IDs originales. Situaciones en las que la trayectoria de dos objetos se intersecten son también adversas, ya que podrían confundir al modelo y causar un intercambio erróneo en sus IDs.

Existen distintos escenarios de aplicación de MOT que dan lugar a diferentes tipos de modelos. La principal distinción es entre modelos *online* y *offline*. Un modelo online recibe su entrada de datos de un video y lo procesa secuencialmente cuadro a cuadro, dando una salida para cada cuadro. En cada cuadro, la única información con la que se cuenta es la del cuadro actual junto con la de los vistos anteriormente. Los modelos offline trabajan con toda la secuencia de video a la vez, por lo que en todo momento cuentan también con la información de los futuros cuadros. Al tener acceso a mayor información, es de esperar que el seguimiento offline logre mejores resultados, al costo de ser más restrictivo en los escenarios a los que puede ser aplicado. Este tipo de seguimiento puede ser visto como un problema de optimización en el que el objetivo es encontrar una serie de trayectorias que minimicen una función de pérdida global determinada. Con este planteamiento, el problema ha sido resuelto aplicando programación lineal y enrutamiento en k caminos mínimos.

En el seguimiento online, la secuencia de imágenes es procesada de forma secuencial, contando para el procesamiento de un cuadro solamente con la información de los cuadros pasados. Debido a esta limitación en la información disponible, sufre una mayor dificultad en determinar a cuáles seguimientos

previos asociar detecciones de objeto ruidosas del cuadro de video actual. Como contrapartida, el seguimiento online puede ser aplicabale en escenarios en los que sea necesario un seguimiento en tiempo real, como navegación de robots o manejo autónomo de vehículos.

También es posible categorizar los modelos de MOT en dos grandes grupos según cómo sean inicializados los objetos a seguir: Seguimiento Basado en Detecciones (DBT por las siglas de su nombre en inglés *Detection-Based Tracking*) y Seguimiento Libre de Detecciones (DFT por *Detection-Free Tracking*). En el primero, también referido usualmente como seguimiento por detección, los objetos son primero detectados y después vinculados a trayectorias, mientras que el otro se trata de modelos que parten de una inicialización de los objetos a los que se quiere seguir en el primer cuadro, y después localiza esos objetos en los cuadros siguientes.

Al estar ligado al uso de un detector previamente entrenado, la mayoría de los modelos DBT se focalizan en una categoría específica de objetivos como pueden ser peatones, automóviles o caras, mientras que los modelos DFT al no requerir de un detector pueden ser potencialmente aplicados a una mayor cantidad de tipos de objetos diferentes. Sin embargo, DFT se focaliza específicamente en seguir unos pocos objetos individualmente sin encontrar una solución global que los abarque a todos, y su complejidad aumenta enormemente con el número de objetos. Tampoco es posible en este tipo de modelos expandir el seguimiento a nuevos objetos que aparezcan en el video, el seguimiento se limita solamente a los etiquetados manualmente al inicio. Debido a estas limitaciones, el uso de un modelo DBT es preferible en escenarios dinámicos donde la entrada y salida de objetos en el campo de vision ocurra con frecuencia y sobretodo, se disponga de un detector capaz de identificar en la imagen la categoría de objetos buscada. A continuación vamos a profundizar en este tipo de modelos, que es el elegido para nuestro trabajo.

2.3.2. Seguimiento por detección (DBT)

Una característica fundamental de los modelos DBT es mantener una clara distinción entre las actividades de detección de objetos y su seguimiento. Generalmente, para cada cuadro de la secuencia se aplica primero el detector para localizar los objetos, y después se asocian estas detecciones a través de los diferentes cuadros usando características tales como su ubicación, su veloci-

dad o su apariencia. Yu et al. [52] remarcan que el rendimiento del seguimiento por detección es fuertemente dependiente de la precisión del detector usado, pudiendo obtener resultados cercanos al estado del arte en el problema MOT incluso con algoritmos de seguimiento simples si se cuenta con una detección eficiente. Sin embargo no se debe perder de vista que una mayor precisión en la detección implicará generalmente un mayor costo computacional, lo que puede reducir la velocidad del seguimiento y comprometer su utilidad en escenarios de tiempo real.

La metodología más comúnmente utilizada por los modelos DBT consta de dos fases: predicción de la ubicación de los objetos que se están siguiendo, y asociación entre estas predicciones y las detecciones observadas [6]. Para cada frame procesado, el algoritmo de seguimiento realiza (1) una *Detección* de los objetos de interés (2) *Predicción* de las nuevas ubicaciones esperadas de los objetos vistos en cuadros anteriores y (3) *Asociación* entre los objetos del cuadro actual y los objetos vistos en cuadros anteriores. La mayoría de los métodos comparten esta metodología aunque existen algunos que difieren, por ejemplo manteniendo múltiples hipótesis activas simultáneamente [23].

Detección

El primer paso corresponde a la detección, en la que la imagen del cuadro a procesar es analizada por el detector usado, que extrae la ubicación de todos los objetos de las clases buscadas. Esta detección, es decir la salida del detector, es representada generalmente por un conjunto de cajas delimitadoras, pudiéndose representar cada una de ellas por las coordenadas de los píxeles de sus esquinas superior izquierda e inferior derecha, o en otros casos por las coordenadas del centro de la caja junto con su ancho y largo. Dado que los detectores no funcionan de forma binaria sino que manejan para cada detección un valor de la confianza en la correctitud de esa detección, cada caja puede ser asociada también a este valor. Este valor de la confianza que se tiene en cada observación puede ser utilizado por el algoritmo de seguimiento para ponderar el costo de su asociación en el paso correspondiente.

Los primeros algoritmos de seguimiento por detección utilizaban principalmente máquinas de vectores de soporte (*Support Vector Machine, SVM*) como detectores. Avidan [4] fue uno de los primeros en trabajar este tipo de modelos, utilizando un SVM entrenado para detectar vehículos y ecuaciones de flujo

óptico para conectar las detecciones a través de los cuadros de la secuencia de video. A partir de que Krizhevsky et al. [25] ganasen por amplio margen en el 2012 la *ILSVRC* (ImageNet Large-Scale Visual Recognition Challenge), una competencia que consiste en medir el rendimiento de modelos de análisis de imágenes, comenzó a haber un viraje hacia métodos basados en redes neuronales convolucionales, como también sucedió en todos los otros problemas de visión por computadora. Hoy en día los métodos de detección que utilizan redes neuronales conforman el estado del arte, y superan ampliamente los resultados obtenidos por los métodos previos basados en SVM.

Predicción

En el paso de predicción el seguidor (*tracker* en inglés, término que usaremos para referenciar al método de seguimiento) genera una serie de ubicaciones donde espera encontrar en el cuadro actual a los objetos vistos anteriormente. Ya que la nueva ubicación es el resultado de la ubicación anterior junto con el movimiento que haya tenido el objeto en el intervalo de tiempo entre los cuadros, el problema puede ser visto como uno de filtrado. En un problema de este tipo, el objetivo es establecer a través de un conjunto de observaciones ruidosas la mejor estimación para el verdadero valor del estado interno de un sistema. Puntualmente en el seguimiento de un objeto, las observaciones ruidosas son las detecciones vistas a lo largo de la secuencia, y el estado interno que se intenta determinar a partir de ellas es su velocidad. Otras técnicas aplicadas en la predicción han sido el uso de flujos ópticos [53], redes neuronales recurrentes [32] [34] o filtros de partículas [41].

Se busca entonces modelar la velocidad de un objeto teniendo acceso solamente a una observación de su posición en tiempos discretos, como un problema de filtrado. En cada instante de tiempo t, el objetivo es determinar el siguiente estado x_t dado un conjunto de observaciones $z_{1:t}$, potencialmente ruidosas.

En el caso del seguimiento por detección, las observaciones con las que se cuenta son las cajas delimitadoras retornadas por el detector. El estado interno x contiene información sobre la posición de la caja, sus dimensiones y su velocidad, y también puede agregársele información adicional como aceleración o deformación de la forma de la caja al costo de un mayor costo computacional. El objetivo de determinar la posición en el cuadro $k+1$ equivale a predecir el valor de z_{k+1}. Sabiendo la diferencia de tiempo entre el cuadro k y el $k+1$,

la nueva posición z_{k+1} es estimada sumando la velocidad x_k a la observación previa z_k. El problema de filtrado es precisamente el de resolver cuál es la mejor estimación para esta velocidad x_k.

El filtro de **Kalman** es una forma óptima de estimar el estado de un sistema dinámico lineal. Fue presentado por Kalman en 1960 [19], quien demostró que la estimación lograda minimiza el error cuadrático medio bajo la hipótesis de que el ruido sigue una distribución gaussiana normal.

Dado un estado $x \in \mathbb{R}^n$ y una señal $u \in \mathbb{R}^l$, el estado es gobernado por la ecuación

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{2.18}$$

donde A es la matriz de transición de estados, B es la matriz de control y w_k es el ruido del proceso. La matriz A modela la predicción del siguiente estado dado solamente el estado interno actual, mientras que la matriz de control B modela los cambios que no están relacionados al estado interno mismo sino que son atribuibles a causas externas como el ambiente exterior.

Dado que solo se cuenta con mediciones de instrumentos que manejan cierta incertidumbre, el estado x no es directamente observable sino que se relaciona a la variable observada $z \in \mathbb{R}^m$ de la siguiente manera

$$z_k = Hx_k + v_k \tag{2.19}$$

donde H es la matriz de observación, que modela la incertidumbre de los sensores. Obsérvese que $x \in \mathbb{R}^n$ mientras que $z \in \mathbb{R}^m$, pudiendo ser m potencialmente diferente de n. Las unidades y escalas de la lectura pueden no ser las mismas que las del estado que se esta siguiendo, y esta conversión es también manejada por la matriz de observación H.

Las variables aleatorias w_k y v_k que modelan el ruido del proceso y de la medición se asumen independientes y distribuidas normalmente con media cero y covarianza Q y R respectivamente. El filtro de Kalman computa una estimación óptima $\hat{x}$ combinando recursivamente las estimaciones previas junto con las nuevas observaciones obtenidas. Consiste de un primer paso de *predicción*, en la que el estado óptimo $\hat{x}_k^{(-)}$ es computado previo a la observación z_k y un paso de *actualización*, en la que un nuevo estado óptimo a posteriori $\hat{x}_k$ es computado después de haber observado z_k. Adicionalmente, se calcula en cada paso la covarianza del error a priori y a posteriori, $P_k^{(-)} = E[e_k^{(-)} e_k^{(-)T}]$ donde

$e_k^{(-)} = x_k^(-) - \hat{x}_k^{(-)}$. El paso de predicción es:

$$\hat{x}_k^{(-)} = A\hat{x}_{k-1} + Bu_{k-1}, \tag{2.20}$$

$$P_k^{(-)} = AP_{k-1}A^T + Q \tag{2.21}$$

y el paso de actualización

$$K_k = P_k^{(-)}H^T(HP_k^{(-)}H^T + R)^{-1} \tag{2.22}$$

$$\hat{x}_k = \hat{x}_k^{(-)} + K_k(z_k - H\hat{x}_k^{(-)}) \tag{2.23}$$

$$P_k = (I - K_kH)P_k^{(-)} \tag{2.24}$$

donde I es la matriz de identidad, Q es la covarianza del ruido del proceso y R es la covarianza del ruido de la medición. La matriz K_k es llamada la ganancia de Kalman e influencia cuánto la observación impacta en la estimación del estado.

El filtro de Kalman puede ser aplicado entonces a cualquier sistema dinámico lineal sometido a ruido blanco aditivo. Las matrices A, B, Q, H y R deben de especificarse según el problema que se esté tratando, así como también es necesario dar estimaciones iniciales de $\hat{x}$ y P [46]. De esta aplicación del filtro de Kalman, la salida del paso de predicción del seguimiento serán las ubicaciones donde se cree más probable que estén los objetos observados en cuadros anteriores para los que se mantiene el seguimiento.

Asociación

El tercer paso en el proceso de seguimiento por detección corresponde a la tarea de asociación, en la que se determina para cada detección observada a qué objeto seguido corresponde basado en su predicción del paso anterior, o alternativamente si la detección representa un nuevo objeto no visto anteriormente.

El problema de asignar un conjunto de detecciones a objetos seguidos puede verse, si la cardinalidad de ambos conjuntos es igual, como un problema de asignación, en el que el objetivo es encontrar una asociación óptima entre dos conjuntos de elementos. En estos problemas se refiere comúnmente a los elementos de un conjunto como *agentes*, y a los elementos del otro como *tareas*. Existe un costo c_{ij} asociado a asignar una tarea j a un agente i, y el objetivo

es encontrar la forma de asignación que mantenga un costo mínimo, bajo la restricción de que ningún agente es asociado a más de una tarea y ninguna tarea es asociada a más de un agente.

Formalmente, si A es el conjunto de agentes, T el de tareas y x_{ij} representa la asignación de la tarea j al agente i, valiendo 1 si la asignación es realizada o 0 en caso contrario, el objetivo es minimizar

$$\sum_{i \in A} \sum_{j \in T} c_{ij} x_{ij} \tag{2.25}$$

sujeto a las restricciones

$$\sum_{i \in A} x_{ij} = 1, j \in T, \tag{2.26}$$

$$\sum_{j \in T} x_{ij} = 1, i \in A, \tag{2.27}$$

$$x_{ij} \geq 0, i \in A, j \in T \tag{2.28}$$

El algoritmo Húngaro [27], también conocido como algoritmo de Kuhn-Munkres, resuelve el problema de asignación en un tiempo polinomial con complejidad $\mathcal{O}(n^3)$, siendo n el número de agentes [12]. La entrada al algoritmo es una matriz de costos C, en la que cada entrada $C(i, j)$ es el costo c_{ij}. A través de cuatro pasos la matriz es manipulada para calcular la asignación óptima. Si el número de agentes y de tareas no coincide, se pueden agregar filas o columnas con valores altos a la matriz, para llevarla a una forma cuadrada y se pueda calcular la asignación de todas formas.

Para poder construir la matriz C de costos es necesario determinar una forma de definir el costo de asignar una detección a una predicción. Ya que la representación que se tiene de cada una es una caja delimitadora, el costo será el equivalente a una medida que se tenga de la semajanza de ambas cajas. Una forma de calcular esta semejanza es mediante la intersección sobre la unión, también conocida como el índice Jaccard

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2.29}$$

donde $|.|$ denota el área de la caja delimitadora. Esta forma de calcular la semejanza depende tanto de la distancia que tengan las cajas así como de la semejanza en sus tamaños y formas, y ha sido utilizada satisfactoriamente ya

sea como la única medida para determinar el costo [6], o junto con la semejanza en apariencia también [48].

Métricas de evaluación

Para definir la calidad de la detección de objetos se usa comúnmente la precisión y la precisión media promedio (mAP por las siglas de su nombre en inglés *mean average precision*), en un umbral de IoU dado. Es decir, se fija un valor de IoU según el cual se cuenta una detección como correcta si su IoU con la ubicación real del objeto es superior a este valor umbral. Por ejemplo, si se fija el umbral de IoU a 0.5, una detección se cuenta como correcta si su intersección sobre la unión con la ubicación real del objeto es mayor a 0.5. En el seguimiento por detección, la precisión del detector influye notablemente el desempeño general del seguimiento, por lo que no es significativo comparar dos seguidores que utilicen detectores diferentes. Incluso usando el mismo detector, dos algoritmos de seguimiento diferentes podrían obtener resultados totalmente diferentes, por lo que son necesarias métricas adicionales para medir la calidad en el problema del seguimiento.

En el 2008, Bernardin y Sttieflhagen [5] propusieron dos métricas estandarizadas para ser usadas en los problemas MOT, a las que llamaron las métricas *CLEAR MOT*. Son presentadas a continuación junto con las métricas adicionales usadas en el MOTChallenge.

FP Falsos positivos, número total de ocurrencias en las que un objeto es detectado pero en verdad no existe ninguno en esa posición.

FN Falsos negativos, número total de ocurrencias en las que un objeto existente no es correctamente detectado.

ID Sw Cambio de identidad, número de veces que se le asigna a un objeto que ya visto anteriormente un nuevo ID.

MOTA Exactitud del seguimiento de objetos múltiples (por sus siglas en inglés *Multiple Object Tracking Accuracy*), es la combinación de las tres métricas anteriores, definida como

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + id_s w_t)}{\sum_t g_t} \tag{2.30}$$

donde para el cuadro t, g_t es el número de objetos presentes en el cuadro, fn_t el número de falsos negativos, fp_t el número de falsos positivos y $id_s w_t$

el número de cambios de identidad. Así, un seguimiento perfecto alcanzaría 100 % de MOTA, ya que tendría $fp_t = 0$, $fn_t = 0$ y $id_s w_t = 0$.

MOTP Precisión del seguimiento de objetos múltiples (por sus siglas en inglés *Multiple Object Tracking Precisión*), mide el acoplamiento de las cajas delimitadoras predichas a las verdaderas. Pese a ser una métrica usada en MOT, está más relacionada a la calidad de la detección que a la de las trayectorias del seguimiento. Se define como

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{2.31}$$

donde d_t^i es la distancia entre la caja delimitadora verdadera y la predicción para el objeto i en el cuadro t, y c_t es la cantidad de detecciones encontradas.

FAF Falsas alarmas por cuadro, es el promedio de falsos positivos que se tuvo a lo largo de la secuencia de video.

MT Objetos mas seguidos (*Mostly Tracked*) es el número de objetos verdaderos a los que se les asigna la misma ID en más del 80 % del video.

ML Objetos mas perdidos (*Mostly Lost*) es el número de objetos verdaderos a los que se les asigna la misma ID en menos del 20 % del video.

2.3.3. Algoritmos de seguimiento

A continuación se presentan dos algoritmos de seguimiento por detección. El primero, SORT, fue propuesto por Bewley et al [6] en el 2016 como un algoritmo que manteniendo una gran simpleza lograba un buen rendimiento a una gran velocidad. En contraposición a los trabajos anteriores, que incorporaban al seguimiento numerosos componentes para tratar casos límites y corregir potenciales errores de detección, SORT deja la tarea de detección completamente a cargo del detector, sin intentar corregir errores desde las fases posteriores del seguimiento. Capitalizando los avances en detección visual de objetos logrados desde la introducción al área de las redes neuronales convolucionales, SORT logra un rendimiento similar al logrado por otros detectores del estado del arte más complicados, y a una velocidad de ejecución superior. El segundo, Deep SORT, fue propuesto por Wojke y Bewley un año después en el 2017 [48], es una modificación al SORT original que incorpora información de la apariencia al cálculo de semejanza entre detecciones encontradas y objetos seguidos.

2.3.4. SORT

SORT (por *Simple Online and Realtime Tracking*) mantiene un seguimiento del movimiento asociando cada objeto visto a un predictor implementado por un filtro de Kalman. En cada cuadro procesado, los objetos son primero detectados, luego las nuevas ubicaciones esperadas de los objetos ya seguidos son calculadas usando el predictor asociado a cada uno, para luego asociar las detecciones a alguno de estos seguimientos según su semejanza. Cada predictor es posteriormente actualizado con la ubicación de la detección que se le asoció, y la posición estimada a posteriori de cada uno es retornada como salida del seguimiento en ese cuadro. A los objetos que no superan el umbral de semejanza con ninguna detección se les asigna una nueva ID, y si un seguimiento permanece un tiempo sin volver a ser asociado a ninguna detección es eliminado.

La implementación original de SORT utiliza como detector una CNN llamada *Faster Region CNN* (FrRCNN) para la detección de peatones en la secuencia de video [37]. Cada objeto es representado por la ubicación de su caja delimitadora junto con su velocidad lineal con respecto al cuadro anterior. El estado interno de cada seguimiento es modelado como:

$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{2.32}$$

donde u y v representan la ubicación horizontal y vertical del píxel central del objeto y s y r la escala y la relación de aspecto del cuadrado que conforma su caja contenedora. Así, u, v y s son considerados dinámicos con una velocidad lineal $\dot{u}$, $\dot{v}$ y $\dot{s}$ respectivamente, mientras que la relación de aspecto r es constante. Estas velocidades son calculadas mediante un filtro de Kalman, actualizadas según la caja delimitadora observada cuando se asocia una detección al seguimiento. Si ninguna detección es asociada, entonces el estado interno es simplemente predicho sin corrección usando el modelo de velocidad lineal.

SORT utiliza como métrica de asociación para construir la matriz de costos la distancia de la intersección sobre la unión, y resuelve la asignación utilizando el algoritmo Húngaro. Un umbral de semejanza mínimo IOU_{min} es establecido como límite para rechazar asignaciones cuya semejanza no supera este valor. Una asignación rechazada significa que no se le asocia a la detección vista ningún objeto seguido anteriormente, sino que se crea para él un nuevo segui-

miento, asignándole un nuevo ID. Al crear un nuevo seguimiento a partir de una detección, el estado interno se inicializa con la geometría de su caja delimitadora y una velocidad nula. Si un seguimiento no es nuevamente detectado por T_{Lost} cuadros es eliminado, previniendo que el número de seguimientos en curso crezca continuamente conforme se analiza la secuencia de video, y limitando el error introducido por predicciones que han pasado largo tiempo sin ser corregidas por el detector. El pseudocódigo es expuesto en el Algoritmo 1.

```
foreach cuadro do
    Obtener detecciones del cuadro;
    Predecir nuevas ubicaciones de los seguimientos;
    Asociar detecciones a seguimientos;
    foreach seguimiento do
        if seguimiento tiene una deteccion asignada then
            Actualizar(seguimiento, deteccion)
        else
            Actualizar(seguimiento)
        end
    end
    foreach deteccion no asignada do
        Inicializar nuevo seguimiento con la deteccion
    end
    Eliminar seguimientos que no hayan tenido asignaciones en los
     ultimos T_Lost cuadros;
end
```

Algorithm 1: SORT Tracking

Evaluado en el MOTChallenge 2015, el rendimiento de SORT logra alcanzar el puntaje MOTA mas alto, y es incluso comparable con el de otros algoritmos del estado del arte contemporáneo que se salen de la categoría de seguimiento en línea y utilizan datos de cuadros futuros [6].

Aunque logra un buen rendimiento en términos de precisión y exactitud del seguimiento, SORT retorna un número relativamente alto de cambios de identidad [48]. La métrica de similitud que utiliza solo es correcta cuando la certeza en la estimación del estado es alta, lo que provoca una deficiencia en el seguimiento a través de oclusiones. Por ejemplo, si el objeto seguido pasa por detrás de un objeto que se interponga entre él y la visión de la cámara, estará por el tiempo que dure la oclusión fuera de detección y su predictor asociado no será actualizado con ninguna observación y su seguimiento, aunque óptimo según el filtro de Kalman, será muy incierto, dificultando su posterior

asociación con la detección correcta cuando vuelva nuevamente al campo de visión. Deep SORT intenta superar esta dificultad reemplazando la métrica de semejanza utilizada por otra que combina información del movimiento y de la apariencia del objeto seguido.

2.3.5. Deep SORT

Deep SORT conserva las mismas características fundamentales del algoritmo SORT original. El estado interno es igualmente representado por una tupla $[u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$ y se utiliza un filtro de Kalman que modela una velocidad constante, y las detecciones en curso son eliminadas si no son asociadas a ninguna detección por T_{Lost} cuadros. La métrica de asociación compara la información de ubicación de de dos detecciones mediante la distancia cuadrática de Mahalanobis:

$$d^{(1)}(i, j) = (\mathbf{d}_j - \mathbf{y}_i)^T \mathbf{S}_i^{-1} (\mathbf{d}_j - \mathbf{y}_i) \tag{2.33}$$

donde $(\mathbf{y}_i, \mathbf{S}_i)$ denota la proyección de la distribución del seguimiento i en el espacio de medidas y $\mathbf{d}_j$ la caja contenedora de la detección j. Esta distancia de Mahalanobis mide la cantidad de desviaciones estándar en que la detección se aleja de la ubicación media predicha por el filtro para el seguimiento. Se utiliza un umbral del 95 % del intervalo de confianza calculado por la distribución χ^2 inversa. Esta decisión se denota con un indicador

$$b_{i,j}^{(1)} = \mathbb{1}[d^{(1)}(i, j) \leq t^{(1)}] \tag{2.34}$$

que evalúa a 1 si la asociación entre el seguimiento i y la detección j es admisible según el umbral de confianza determinado $t^{(1)}$. Esta métrica es apoyada por una segunda métrica que mide la distancia entre las representaciones de la apariencia.

Para obtener información de la apariencia de cada seguimiento, Deep SORT utiliza una CNN entrenada para la re-identificación de personas en un conjunto de datos llamado MARS (*Motion Analysis and Re-identification Set*) que contiene más de 1.100.000 imágenes de 1.261 peatones diferentes [55]. Para su colección se ubicaron seis cámaras en las afueras del campus de la universidad de Tsinghua, dispuestas de forma de que en cada momento cada peatón visto es capturado por al menos dos cámaras desde ángulos diferentes. En este conjunto de datos, los autores de Deep SORT entrenaron una red neuro-

nal convolucional para clasificar correctamente a qué peatón corresponde cada imagen. Después del entrenamiento, la capa final de clasificación es removida, y use usa la salida de la red como un vector de características que describe al peatón. Así, al alimentar la red neuronal creada con dos imágenes diferentes que correspondan al mismo peatón aunque sea visto en diferentes momentos desde diferentes ángulos, la red devolverá dos vectores que serán similares entre sí pero no a otros vectores resultados de imágenes de peatones diferentes. Esta representación de características aprendida por la red es usada por DEEP Sort para obtener una métrica de la semejanza de la apariencia entre una nueva detección y un seguimiento visto anteriormente y determinar si corresponden efectivamente al mismo objeto.

La información de apariencia de una detección $\mathbf{d}_j$ es calculada alimentando a la CNN con la sección de imagen de la caja delimitadora, obteniendo un vector de descripción $\mathbf{r}_j$ con $\|\mathbf{r}_j\| = 1$. Se mantiene una galería $R_k = \mathbf{r}_k^{i\,L_k}{}_{k=1}$ de los últimos $L_k = 100$ vectores de apariencia de cada seguimiento k, y la métrica de apariencia mide la menor distancia de coseno entre esta galería para un track i y una detección j

$$d^{(2)}(i,j) = min1 - \mathbf{r}_j^T \mathbf{r}_j^{(i)} | \mathbf{r}_j^{(i)} \in R_i \tag{2.35}$$

De forma similar a la primera distancia, se introduce una variable binaria para indicar si la asociación según esta métrica es admisible según un umbral $t^{(2)}$

$$b_{i,j}^{(2)} = \mathbf{1}[d^2(i,j) \leq t^{(2)}] \tag{2.36}$$

Se tienen entonces dos métricas, una métrica $d^{(1)}$ que da información sobre la distancia Mahalanobis de la ubicación de los objetos, útil en predicciones de corto rango temporal, complementada por otra métrica $d^{(2)}$ que brinda la semejanza entre la apariencia de dos objetos, útil para recuperar identidades luego de oclusiones prolongadas. Ambas métricas se combinan en una suma ponderada

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{2.37}$$

siendo el costo $c_{i,j}$ de asociar un seguimiento i a una detección j, y aceptando una asociación solo si es aceptable según las variables binarias de ambas

Tabla 2.1: Resultados de algoritmos de seguimiento en el desafío MOT16. [48]

Algoritmo	Tipo	**MOTA**	**MOTP**	**MT**	**ML**	**ID**	**Velocidad**
LMP_p [22]	BATCH	71.0	80.2	46.9 %	21.9 %	434	0.5 Hz
POI [52]	ONLINE	66.1	79.5	34.0 %	20.8 %	805	10 Hz
SORT[6]	ONLINE	59.8	**79.6**	25.4 %	22.7 %	1423	**60 Hz**
Deep SORT [48]	ONLINE	**61.4**	79.1	**32.8 %**	**18.2 %**	**781**	40 Hz

métricas

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)} \tag{2.38}$$

La asignación es resuelta entonces por el algoritmo Húngaro, construyendo la matriz de costos C con los costos de asignación $c_{i,j}$, y solo considerando como válidas las asignaciones para las cuales $b_{i,j} = 1$.

2.3.6. Comparación de resultados

En la Tabla 2.1 se muestran los resultados obtenidos por diferentes algoritmos de seguimiento en el desafío MOTChallenge 2016, el año en que fue presentado Deep SORT. LMP_p [22] realiza un seguimiento del tipo *offline*, y POI [52] un seguimiento *online*.

Si bien LMP_p logra un mejor rendimiento en general, el no poder ser aplicado en tiempo real limita su utilidad. POI es del tipo *online* al igual que SORT, pero si bien las diferencias en su rendimiento no son sustanciales, el último es seis veces más rápido.

Deep SORT presenta resultados similares a SORT en MOTA, MOTP, MT y ML, pero logra reducir a casi la mitad el número de cambios de identidad. En contrapartida, el procesamiento necesario para calcular la descripción de las características de apariencia de los objetos penaliza la velocidad de ejecución, que es inferior a la registrada por SORT.

Capítulo 3

Solución propuesta

3.1. Introducción

En esta sección se describe la solución propuesta para una herramienta de análisis de comportamiento del público a través de videos.

El objetivo es desarrollar una herramienta que permita, a través de la recopilación de videos tomados por cámaras de seguridad, un análisis del comportamiento del público en el lugar.

El diseño se basa en dos pilares, por un lado garantizar una adopción del sistema fácil y de bajo costo de instalación, y por otro lado ofrecer funcionalidades que sean adaptables a diferentes escenarios de interés sin limitarse a un tipo en particular. Si bien un sistema especialmente diseñado para operar en un entorno bien definido y específico lograría un mejor rendimiento adaptándose a sus condiciones concretas, sería a costa de sacrificar su utilidad en otros escenarios y situaciones de uso. Por el contrario, la solución propuesta pretende permitir una aplicación universal que sea capaz de adaptarse a una multiplicidad de escenarios en los que es de interés realizar un análisis del comportamiento del público sin tener que limitarse al análisis de ningún tipo de conducta en particular. Esta generalidad, acompañada de un bajo costo de instalación, permite que el sistema pueda ser utilizado en diferentes escenarios como pequeñas y medianas tiendas, salas de espera de hospitales, restaurantes, oficinas de atención al público, etc.

El costo de instalación es la primer barrera al uso, ya que el usuario del sistema no verá ningún beneficio hasta que el valor aportado lo amortice. Como se mencionó anteriormente este costo se puede minimizar mediante la utiliza-

ción de videocámaras, debido a que su uso se encuentra bastante extendido y su precio de mercado es considerablemente inferior a otros sensores más sofisticados.

La mayoría de locales comerciales cuentan ya con un sistema de vigilancia consistente en una o varias cámaras de seguridad conectadas a un dispositivo central de grabación que se encarga de digitalizar la señal analógica de grabación y almacenar su contenido en un disco rígido. La solución propuesta utiliza las horas de grabación de estas cámaras de seguridad como único insumo para la detección y seguimiento de personas y el análisis de su comportamiento. Además del bajo costo de instalación ya mencionado, ofrece además mayor transparencia en su uso, ya que los visitantes del lugar no notarán ningún cambio durante su experiencia.

El segundo pilar de gran importancia para el diseño de la solución propuesta es la universalidad de sus funcionalidades. Se pretende un sistema que pueda ser aplicado para una variedad de fines en una variedad de escenarios, como puede ser determinar el tiempo de espera de pacientes en la sala de un hospital, de compradores en la fila de un supermercado, de personas que transitan por un pasillo, etc. Se necesita dar un conjunto base de funcionalidades que puedan resolver cada situación sin necesitar un desarrollo especial de capacidades a medida. Para cumplir con este requerimiento, se propone un sistema de consultas que permite la identificación de un público objetivo mediante filtros tales como áreas recorridas y tiempos de permanencia, para luego obtener de estos la métrica que se desee.

3.2. Arquitectura

En la Figura 3.1 se muestra un diagrama de la arquitectura de la solución propuesta. Los videos son inicialmente capturados a través de videocámaras ubicadas en el lugar de forma de cubrir con su campo de visión el área de interés. Los videos son guardados en formato digital por el sistema de grabación DVR (*Digital Video Recording*) que se esté utilizando en un almacenamiento al que luego accede el sistema para el procesamiento de su información, que acaba del otro lado con el usuario realizando consultas sobre métricas del uso de los espacios.

Los videos son alimentados al módulo de análisis de video, que realiza en ellos la detección y seguimiento de personas. El acceso a los videos puede ser

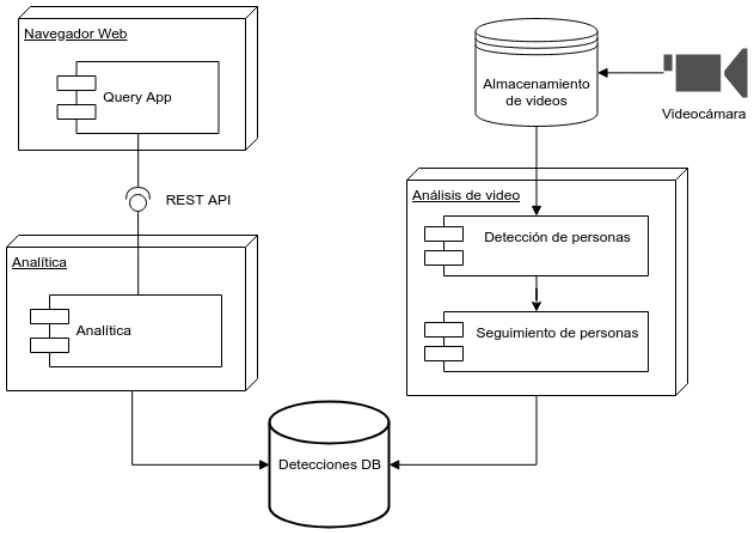

Figura 3.1: Arquitectura de la solución propuesta

realizado en vivo si el sistema DVR permite el acceso a las cámaras mediante un *streaming* o si se utiliza como almacenamiento un disco accesible por red, NAS (*Network Atacched Storage* o SAN (*Storage Area Network*). El análisis de la secuencia de video se realiza cuadro a cuadro, primero el detector devuelve la ubicación en la imágen de las personas encontradas y sus puntos corporales, y esas detecciones son la entrada al módulo de seguimiento, que las asociará con detecciones vistas anteriormente provenientes de las mismas personas. La detección se realiza local en cada cuadro, y el seguimiento realizado es del tipo online, solo tiene en cuenta información de cuadros pasados. Por lo tanto, en análisis puede operar en vivo dando como salida en cada cuadro el conjunto de personas vistas, sus puntos corporales y asignando a cada persona una id única.

Las detecciones resultado del módulo de análisis son almacenadas en una base de datos que guarda para cada persona vista la traza de su ubicación a lo largo de la secuencia de video. Además de estas detecciones, la base también almacena algunas métricas de costo computacional elevado, por ejemplo la información de la distribución de personas vistas en cada regíon de la pantalla que se utiliza para generar el mapa de calor. Esta métrica es consultada constantemente por lo que calcularla nuevamente cada vez sería muy ineficiente. En lugar de eso, se genera bajo demanda y su resultado es almacenado en la base de datos.

El usuario accede a la información del sistema a través de una interfaz

Web. Se le presenta en ella las cámaras instaladas, y al seleccionar una puede ver un resumen de la actividad registrada, en forma de un mapa de calor, y realizar búsquedas de personas bajo ciertos criterios de selección. Una vez seleccionado el conjunto de personas objetivo, puede consultar por las métricas de su comportamiento que le sean de interés.

Las consultas realizadas por el usuario son resueltas por un servidor Web que contiene el módulo de analítica. Este módulo se encarga de seleccionar las detecciones almacenadas en la base que cumplan con el criterio de selección y calcular sobre ellas las métricas que el usuario haya marcado.

3.3. Detalle de la arquitectura

Detección de personas

El módulo de detección de personas tiene como fin bien definido tomar como entrada cada una de las imágenes que conforman la secuencia de video recibida y segmentar en cada una las regiones que corresponden a personas.

Si bien cualquier detector de personas que localice puntos corporales podría utilizarse en este módulo, se utiliza OpenPose en la solución propuesta por su buena performance y su simplicidad. OpenPose ha logrado una gran adopción y ha generado una comunidad de colaboración y desarrollo activa y en crecimiento, lo que asegura que continuarán lanzando nuevas versiones y añadiendo funcionalidades. Desde su lanzamiento en Abril del 2017 se han liberado 10 nuevas versiones de OpenPose, y se han añadido características como nuevos modelos corporales, reconstrucción 3d y mejoras al rendimiento. Es necesario mencionar de todas formas que el uso comercial de OpenPose está sujeto a licencias, mientras que es gratuito para cualquier otro tipo de proyectos[1].

Las detecciones obtenidas por OpenPose son ruidosas y poco estables. Puede ocurrir que se obtengan falsas detecciones de partes corporales y hasta de personas enteras por varios cuadros seguidos, o que la asociación entre partes detectadas y personas se errónea por momentos. Estas imperfecciones en la detección pueden ser vistas como ruido en la detección del instrumento, que puede ser reducido al aplicar el filtrado de Kalman en el módulo de seguimiento posterior.

[1] Ver https://github.com/CMU-Perceptual-Computing-Lab/openpose#license

Seguimiento de personas

El módulo de seguimiento toma para cada cuadro las detecciones y asocia cada una a otras detecciones vistas en cuadros anteriores. Así, un conjunto de detecciones asociadas conforman la traza de una persona, la información de cómo se movió cada uno de sus puntos corporales por el campo de cobertura de la videocámara y que será almacenada en la base de datos para su posterior consulta.

Se utiliza para el seguimiento el algorito Deep SORT, que implementa un seguimiento por detección online, tomando como características la ubicación y velocidad de los objetos así como también información de su apariencia. El diseño de Deep SORT logra ser eficiente al mismo tiempo que mantiene una gran simplicidad, y la utilización de información de apariencia lo hace idóneo para el seguimiento de personas, en el que la capacidad de resolver correctamente oclusiones temporales y superposiciones entre personas es de vital importancia para una buena calidad del seguimiento realizado.

Deep SORT representa los objetos mediante su ubicación en la imágen, su ancho y su largo. Generalmente, estas dimensiones corresponde a la de la caja delimitadora del objeto detectado, es decir el mínimo cuadrado en pantalla que abarca la totalidad del objeto. Las personas detectadas por OpenPose son representadas como el conjunto de puntos correspondiente a cada una de sus partes corporales vistas, por lo que es necesario fijar un criterio para convertir este conjunto de puntos a la representación rectangular de Deep SORT.

La forma más natural de convertir el dominio de salida de OpenPose al de entrada de Deep SORT sería tomar un rectángulo cuyas coordenadas sean los puntos máximos y mínimos en cada eje del conjunto de puntos corporales detectadas. Este sería el cuadrado mínimo que incluiría a todos los puntos, similar al usado generalmente en problemas de segmentación. Sin embargo, las pruebas con este criterio de conversión demostraron que genera mucha inestabilidad en las dimensiones de las cajas delimitadoras, lo que termina empeorando la performance del seguimiento. Los puntos extremos en la pose corresponden la mayor parte del tiempo a manos y piernas, que a su vez son los que más movimiento suelen presentar. Así, si una persona sentada levanta su mano, el largo del rectángulo delimitador de sus detecciones será duplicado repentinamente, lo que generará una velocidad en el estado interno del filtro de Kalman. Estas deformaciones en las dimensiones agregan un ruido permanente

al filtrado que empeora su rendimiento, siendo mejor que las dimensiones de los objetos seguidos permanezcan lo mas estables posibles.

Para lograr dimensiones mas estables en los rectángulos seguidos se propone un criterio de construcción alternativo que en lugar de abarcar todos los puntos procura delimitar la mayor cantidad de puntos centrales, bajo el supuesto de que son esos los que mayor estabilidad en su ubicación tendrán.

El rectángulo delimitador de cada detección queda determinado por sus dimensiones $r = [u, v, w, l]$ donde u y v corresponden a las coordenadas de su centro en el eje x e y, y w y l a su ancho y largo. El centro u, v se calcula como el valor medio de los puntos en cada eje:

$$u = \frac{\sum_{p_i \in P_j} p_i^x}{\|P_j\|} \tag{3.1}$$

$$v = \frac{\sum_{p_i \in P_j} p_i^y}{\|P_j\|} \tag{3.2}$$

siendo $P_j = p_1 ... p_n$ el conjunto de puntos corporales detectados para la persona j, donde $||P|| = n$. Esto ubica el centro del rectángulo en el punto medio entre todas las detecciones. El rectángulo se termina de delimitar sumando de cada lado del centro la desviación estándar de los puntos en cada eje, por lo que el ancho w y el largo l se calculan:

$$w = 2\sqrt{\frac{\sum_{p_i \in P}(p_i^x - u)^2}{\|P\|}} \tag{3.3}$$

$$l = 2\sqrt{\frac{\sum_{p_i \in P}(p_i^y - v)^2}{\|P\|}} \tag{3.4}$$

Si bien podrían adoptarse otros criterios de construcción del rectángulo que usasen información cualitativa de los puntos, por ejemplo tomando solo los puntos de ciertas partes del cuerpo o ponderando según su tipo, se lograrían resultados semejantes al costo de una complejidad adicional.

En la Figura 3.2 se puede ver la diferencia entre aplicar ambos criterios de delimitación de cajas contenedoras. Los rectángulos rojos corresponden a la delimitación mediante puntos máximos, y los verdes al criterio propuesto. Como se observa, las dimensiones del rectángulo siempre son menores, y cubren mayoritariamente el centro del cuerpo de la persona mientras que dejan por fuera los puntos de las extremidades. En la tercera figura, se le asocia incorrec-

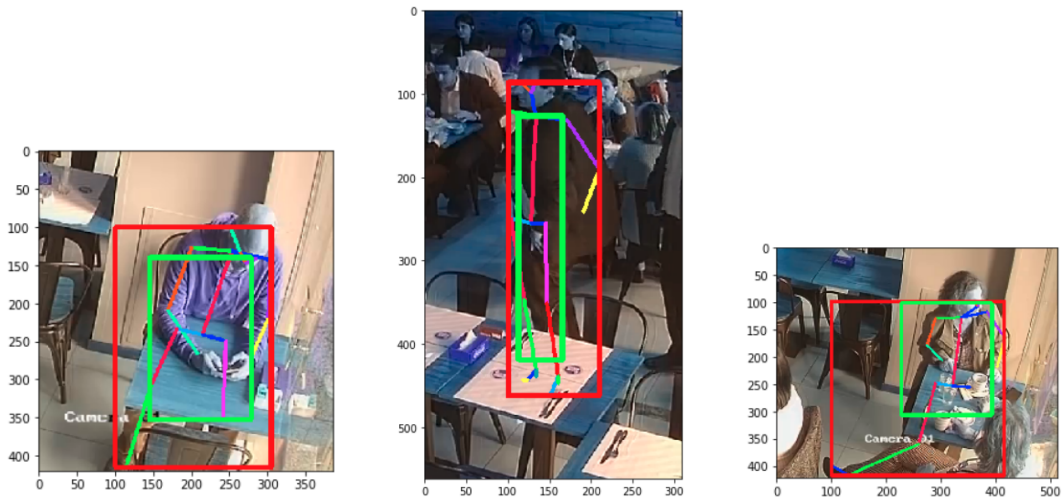

Figura 3.2: Comparación entre cajas delimitadoras.

tamente al cuerpo de la persona sentada una pierna de otra persona; la caja delimitadora por puntos máximos expande su tamaño para incluir este punto, mientras que la caja resultado del criterio propuesto simplemente lo deja por fuera. Si un cuadro siguiente el error en la detección es corregido, el rectángulo verde permanecerá estable, mientras que el rojo cambiará considerablemente su tamaño.

El filtro de Kalman utilizado por Deep SORT para el seguimiento de objetos debe ser parametrizado a las condiciones del problema en que se quiere aplicar. Las matrices de observación, transición y sus covarianzas deben de ser especificadas. Algunas de estas matrices son específicas al problema de seguimiento de cajas delimitadoras, como la matriz de transición de estados, que asocia cada dimensión a su correspondiente velocidad, la matriz de control B, que no es utilizada, y la matriz de observación H. Las matrices de covarianza por otro lado pueden ser personalizables a cada escenario si se tiene algún conocimiento a priori sobre el tipo demovimiento que realizarán los objetos a seguir.

Las matrices de covarianza en la transición y observación pueden utilizarse para regular la incidencia de las nuevas observaciones en el estado interno del objeto seguido, haciendolo más o menos suceptible a cambios. Según el escenario puede ser conveniente regular esta parametrización para reducir esta incidencia, si se sabe que los objetos seguidos tienden a mantener su posición o cambiarla levemente, o incrementarla si presentan gran dinamismo.

El módulo de seguimiento asocia entonces las detecciones en cada cuadro

con otras vistas anteriormente, asignando un mismo id único al seguimiento de cada persona vista. Estos seguimientos son almacenados en la base, donde se puede consultar para una persona cuál fue la traza de su movimiento, o para un área de la imágen cuáles fueron las personas que pasaron por ahí.

Análisis de las detecciones

El módulo de Análisis es el encargado de tomar las consultas realizadas por el usuarios y encontrar las detecciones de personas que cumplan con los criterios ingresados. Sobre este conjunto de personas encontradas, el usuario podrá realizar filtrados adicionales o consultar por métricas que le interese conocer.

Se necesita de un mecanismo de ingreso de las consultas que sea lo suficientemente expresivo de forma que el usuario pueda crear diferentes criterios de selección de su interés en distintos tipos de escenarios, y que sea también simple de usar y de entender su funcionamiento.

Se propone expresar las consultas en forma de una concatenación de filtros por zonas, dónde cada filtro corresponde a un área de la pantalla. Las personas seleccionadas por la consulta serán aquellas para las cuales existen detecciones observadas que estén dentro de alguna de las zonas delimitadas. Así, la forma de ingresar una consulta por el usuario es dibujando en el plano de imagen de la escena capturada por la videocámara un polígono dentro del cual deberán haber sido vistas en algún momento las personas buscadas.

El usuario debe poder además filtrar adicionalmente las personas imponiendo un tiempo de permanencia máximo o mínimo o un intervalo temporal de interés. Esto, además de dar más herramientas al usuario para realizar consultas más precisas, es un requerimiento necesario en caso de que la base de grabaciones sea muy extensa, para evitar consultas que requieran demasiado tiempo de procesamiento.

Mediante la concatenación de varios polígonos de búsqueda, el usuario puede formar consultas más complejas, como seleccionando las personas que hayan pasado por una zona y luego por otra. Los filtros también pueden ser acompañados de operadores lógicos, para poder expresar consultas como la búsqueda de personas que hayan pasado por una zona pero no por otra. Finalmente, el usuario puede especificar si el orden de los filtros debe ser respetado o si no es relevante, es decir si solo se quiere seleccionar a las personas que fueron vistas

pasando por una zona A y luego por una zona B, o si cualquier orden de visita es suficiente para su selección.

Una vez ingresado desde el sistema Web el criterio de búsqueda, la consulta es enviada al módulo de análisis por intermedio de una interfaz REST, el cual buscará en la base de datos las trazas de personas cuyas detecciones cumplan con las condiciones impuestas. Se trata de una búsqueda de gran costo computacional ya que su complejidad impide el uso de índices en campos de registros para incrementar la velocidad de selección. Todas las personas vistas son potencialmente seleccionables, es necesario iterar sobre cada una de ellas y aplicar los criterios de selección a sus detecciones para determinar si es seleccionable o no. El tiempo de ejecución de la consulta incrementa con la cantidad de horas de video analizadas y la cantidad de detecciones en ellas vistas.

Finalizada la búsqueda, el conjunto de personas seleccionadas es mostrado al usuario. Se muestra la cantidad de personas encontradas y el tiempo de permanencia de cada una de ellas.

En la sección 3.4 se presenta un prototipo de la solución propuesta, incluyendo capturas de pantalla que ilustran la interfaz Web para consultas.

3.3.1. Búsqueda de poses

La detección de los puntos corporales para las personas detectadas en cada uno de los cuadros de la secuencia de video realizada por OpenPose abre la posibilidad de buscar poses corporales que sean de interés para el usuario de la plataforma. De esta forma, el sistema podría ofrecer tambien la posibilidad de agregar a la consulta filtros que seleccionen personas cuya posición corporal sea semejante a una de interés. Por ejemplo, podría ser de interés detectar las personas que estuvieron sentadas en una zona, o las que extendieron sus manos hacia un objeto.

Tanto para el seguimiento de las personas (realizado a través de sus cajas contenedoras) como para los criterios de búsqueda por posición descritos en las secciones anteriores, los puntos corporales detectados eran utilizados solamente para determinar la posición de la persona en la imagen, solo se tomaba la información de su ubicación sin importar el punto antómico al que correspondían. La clasificación de puntos corporales es utilizada al realizar una búsqueda de pose corporal, justificando asi el uso de OpenPose sobre otro detector que solo

haga una detección de personas sin aportar información sobre su pose.

OpenPose representa la pose humana mediante 25 puntos, por lo que definir un criterio de búsqueda de una pose objetivo equivale a determinar cual es la posición relativa de cada uno de estos puntos. Para brindar al usuario una forma intuititva de realizar esto desde la interfaz gráfica, se consideraron dos posibles maneras. Una posibilidad es representar los 25 puntos en pantalla mediante un esqueleto con el cual el usuario pueda interactuar para ajustar la posición de cada punto corporal hasta lograr determinar la pose buscada. Si bien de esta forma es posible determinar cualquier posición, el uso puede resultar complicado a una persona que no esté familiarizada con el diseño 3D y las dificultades de mover la vista a través de los ejes, comprender la perspectiva de la cámara, etc.

Una segunda opción sería determinar la pose objetivo a través de una semejanza con otra pose vista ya en el video para alguna persona. Así, el usuario debería moverse por la secuencia grabada hasta encontrar una persona que haya realizado el comportamiento que es de su interés y crear un criterio de búsqueda en el que se filtrarían las personas a las que se haya detectado una pose similar en algún momento. Este método sería mucho más ágil que el primero, pero como contrapartida limitaría la expresividad de la consulta a poses que ya hayan sido vistas anteriormente. En todo caso, ambos métodos de búsqueda no son excluyentes y el sistema podría implementar ambos y permitir al usuario elegir el que prefiera utilizar.

3.4. Prototipo realizado

Con el fin de comprobar la viabilidad técnica del sistema propuesto, fue construido un prototipo con un conjunto mínimo de funcionalidades. El objetivo fue validar que la tecnología disponible hoy en día para la detección de personas y su seguimiento es suficiente como para construir la plataforma ideada, y evaluar la calidad de los resultados luego de su aplicación en un escenario en particular.

La viabilidad del proyecto depende fuertemente de la calidad lograda en la detección y el seguimiento. Un alto margen de error en cualquiera de las dos implicaría una distorsión en las métricas obtenidas que acabaría por quitarles cualquier utilidad. El prototipo construido tiene como principal objetivo relevar la calidad obtenida y el impacto que los errores de detección provocan

sobre las métricas de comportamiento de las personas.

Se utilizó para la detección la versión 1.3.0 de OpenPose con una resolución de 656x368. El modelo de cuerpo humano utilizado es BODY_25, sin incluir en la detección los puntos correspondientes a manos y caras, que introducen una degradación significativa en la performance. La detección es ejecutada sobre archivos de videos ya capturados. Se captura cada cuadro con OpenCV[1], y se le aplica OpenPose, que genera como salida, para cada cuadro, un archivo en formato JSON con las personas vistas y las coordenadas en pantalla para los puntos corporales detectados. Estos archivos de detecciones son alimentados al módulo de seguimiento, construido con la implementación de Deep SORT disponible en su repositorio [49]. Deep SORT fue modificado para utilizar en lugar de la implementación del filtro de Kalman con la que es distribuido, una diferente utilizando la librería PyKalman [36], que tiene un diseño más modular y permite una parametrización más fácil del filtrado.

Las detecciones son almacenadas en una base de datos no relacional MongoDB[2]. MongoDB almacena los datos en documentos de formato JSON[3], organizándolos en diferentes colecciones. Cada detección es almacenada en un documento diferente, registrando los puntos corporales observados y el ID de persona que se le asignó en el seguimiento. La utilización de una base no relacional facilita el prototipado al no ser necesaria una definición estática de la estructura de la base y de los tipos de datos almacenados.

El módulo de análisis expone en una interfaz REST[4] los servicios de búsqueda y consulta de métricas. Se utilizó el framework Flask[5] escrito en Python, que gracias a un diseño minimalista permite crear rápidamente aplicaciones Web. El usuario ingresa las consultas a través de una interfaz Web desarrollada en Angular 6[6].

Para mantener la simplicidad del prototipo y acotar los tiempos de desarrollo, la búsqueda fue limitada a un solo polígono. Si bien la potencialidad de búsqueda se ve limitada y se restringe al usuario a indicar una sola área de interés, es suficiente para evaluar el desempeño del sistema de búsqueda. En una búsqueda compuesta de una concatenación de filtros, será el primer

[1]Por más información sobre OpenCV, ver *https://opencv.org/*
[2]Por más información sobre MongoDB, ver *https://www.mongodb.com/*
[3]Por más información sobre el formato JSON, ver *https://www.json.org/*
[4]Por más información sobre APIs de tipo REST, ver *https://restfulapi.net/*
[5]Por más información sobre Flask, ver *http://flask.pocoo.org/*
[6]Por más información sobre Angular, ver https://angular.io/

filtro el que mayor costo computacional implique, pues todas las detecciones almacenadas en la base de datos son posibles candidatos. Los sucesivos filtros se aplican solo a los resultados de los filtros anteriores, por lo que el costo computacional será considerablemente menor.

La interfaz Web de búsqueda presenta al usuario un cuadro vacío (es decir, sin personas), tomado de los videos disponibles que conforman el dataset utilizado para la evaluación, como se describe en dicha sección. El usuario tiene la posibilidad en ese momento de dibujar, mediante un polígono, el área que le interesa consultar. El sistema buscará entonces todas las personas que fueron detectadas en esa zona, y retornará los siguientes resultados:

- un listado de personas detectadas en esa zona, identificados por su ID, e indicando la cantidad de minutos que permanecieron en la misma
- un histograma que resume la permanencia en minutos en función de la cantidad de personas detectadas
- filtros adicionales por tiempo, que permiten al usuario filtrar los resultados según un rango de fechas, o por permanencia mínima y/o máxima

La Figura 3.3 presenta una ilustración sobre la interfaz disponible al usuario, mientras que la Figura 3.4 ilustra la presentación de los resultados una vez efectuada la búsqueda.

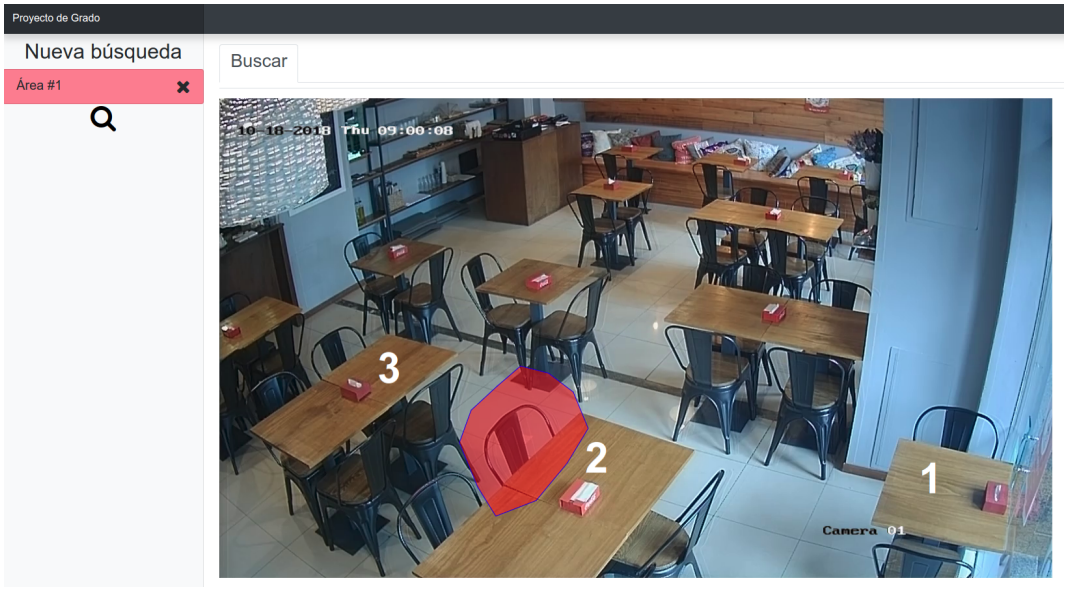

Figura 3.3: Ilustración de la interfaz Web de consulta disponible al usuario. El área de interés dibujada es la delimitada por el polígono rojo. La numeración de las mesas fue realizada a los efectos de la evaluación, como se describe más adelante en dicha sección.

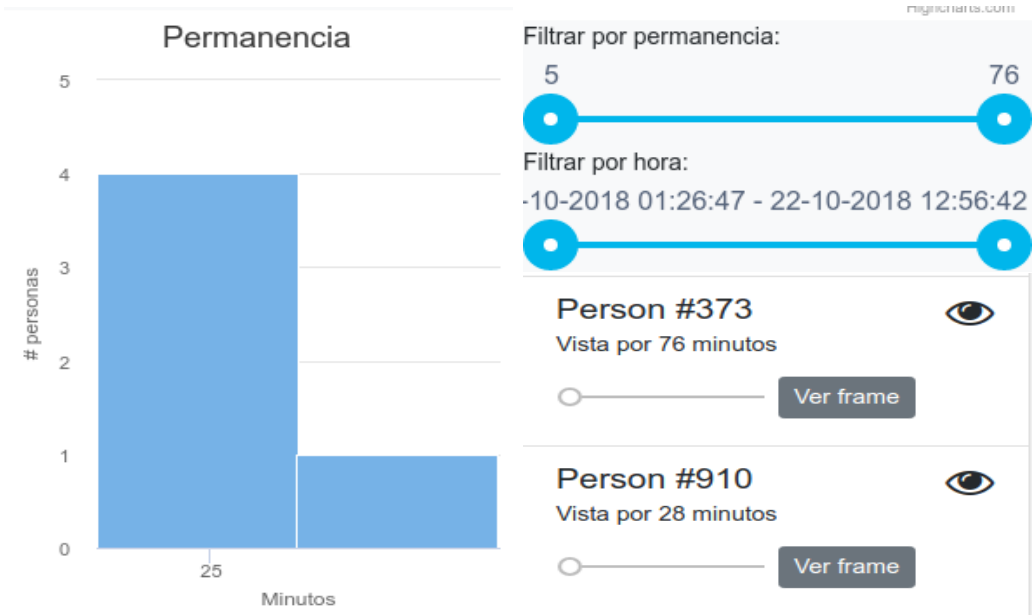

Figura 3.4: Ilustración de lo que retorna la interfaz Web al usuario luego de completada una búsqueda. A la izquierda se presenta un histograma que resume la permanencia de las personas. A la derecha, arriba, se ilustran los filtros por tiempo que el usuario puede utilizar para refinar los resultados, y a la derecha, abajo, se enumeran las personas que fueron detectadas en la zona consultada

La búsqueda de poses fue implementada directamente en un cuaderno Jupyter con Python, tomando las detecciones directamente de la base de datos. No fue implementado en el prototipo una interfaz gráfica de búsqueda de pose.

En la siguiente sección se presentan los resultados obtenidos en la utilización de este prototipo desarrollado para el análisis de un conjunto de videos de prueba.

Capítulo 4

Evaluación de la solución

El funcionamiento del prototipo implementado para la solución propuesta en el capítulo anterior es analizado en este capítulo. Se intenta validar la viabilidad tecnológica del sistema, que dependerá fundamentalmente del costo computacional de la detección y del análisis, y de la calidad por estos lograda.

Se evaluó el rendimiento de las dos grandes funcionalidades implementadas en el prototipo del sistema. Por un lado los criterios de selección de personas y la obtención de métricas, en particular del tiempo de permanencia en un área dado, y por otro lado la búsqueda de personas que hayan realizado una pose objetivo dada.

4.1. Conjunto de datos utilizados

Se utilizaron secuencias de video capturadas por una cámara de seguridad en una cafetería ubicada en la zona céntrica de la ciudad de Montevideo. El local permanece abierto de Lunes a Viernes de 09:00 a 19:00 horas. El mismo cuenta con un sistema de 8 cámaras de videovigilancia, que almacena las grabaciones en el disco duro de una unidad DVR central. Se seleccionó para el análisis 2 días de videos provenientes de una cámara que enfoca convenientemente las mesas del lugar, en el horario de 09:00 a 17:00. Los videos corresponden a los días Jueves 18 y Lunes 22 de Octubre del 2018.

Se le presentó al propietario del local el sistema en construcción, sus funcionalidades y posibilidades. Consultado sobre qué métricas de uso del lugar eran de su interés, destacó la importancia de conocer el tiempo de ocupación de las mesas, para poder analizar la utilización de la capacidad del local y los

Figura 4.1: Captura de video del Jueves 18 de Octubre a las 13:58

momentos del día más concurridos.

Bajo estos requerimientos, el sistema podría ser utilizado por el usuario para consultar el tiempo de permanencia de las personas en las mesas. La forma de obtener esta información utilizando el sistema de búsqueda y filtros propuesto, es realizar una consulta las mesas que resulten de interés. Es suficiente para cada búsqueda un solo filtro en el que se dibuje en el plano de imagen la zona en la que se colocan las sillas de la mesa. La consulta encontrará todas las personas que haya estado sentadas en esas sillas, y sus respectivos tiempos de permanencia en la mesa, como se ilustró en las Figuras 3.3 y 3.4.

La parametrización del filtro de Kalman fue realizada con el fin de mejorar la performance de la detección sobre personas quietas, que son las de mayor importancia para la búsqueda que será realizada. La matriz de covarianza en la transición utilizada fue de $10 * \mathbb{I}$ y la de covarianza en la observación de $500 * \mathbb{I}$, donde $\mathbb{I}$ es la matriz identidad de tamaños 8×8 y 4×4 respectivamente. Estos parámetros fueron determinados a partir de pruebas realizadas, buscando una configuración que maximice la calidad del seguimiento en las personas sentadas.

4.2. Análisis de permanencia de personas

El objetivo del análisis bajo estas condiciones es inferir, como métrica, la permanencia de las personas sentadas en las mesas del local. Idealmente se bus-

ca que las métricas inferidas por la solución sean lo más consistentes con las métricas reales (que podrían obtenerse por ejemplo, si alguien analiza el video manualmente, identifica a cada persona sentada y registra el tiempo que permanecieron sentadas y donde). En otras palabras, lo que se busca es identificar unívocamente a cada persona con un código, durante toda su permanencia en los videos. Sin embargo las métricas inferidas pueden verse severamente afectadas si se presentan errores en la detección y el seguimiento de las personas. La falla en la detección de una persona, sería interpretado por el módulo de análisis como personas distintas, que entran y salen constantemente del área filtrada, incrementando artificialmente la cifra de personas vistas y disminuyendo el tiempo de permanencia registrado. De manera similar, problemas en el seguimiento causarán que a una misma persona le sean asignadas múltiples identidades, o que se le asigne la misma a dos personas diferentes, en ambos casos deformando las métricas obtenidas durante el análisis.

Los errores en la detección, cometidos por OpenPose durante el procesamiento de los videos, pueden significar que algunos puntos corporales (o todos) de una persona no sean detectados, que sean asignados a partes del cuerpo incorrectas, o que se realice una falsa detección (por ejemplo, detectar una silla como si fuese una persona). Estos errores, de ocurrir, son locales a un cuadro específico de la secuencia de video, y tienden a ser corregidos si se tiene en cuenta la secuencia completa. Es decir, puede haber un error en la detección en un determinado cuadro, pero el movimiento de la persona y la variabilidad entre cuadros de la secuencia causarán que la probabilidad de que el error se mantenga decrezca conforme se tienen en cuenta más cuadros.

Con respecto al seguimiento, de los tipos de errores presentados en la Sección 2.3.1, los indicadores de FP y FN afectan a detecciones en particular, por lo cual al igual que los errores en la detección no tiene gran peso en las métricas. Tampoco resulta de importancia la precisión en las cajas contenedoras construidas a partir del seguimiento, ya que en el análisis no se trabajará con esas cajas sino con los puntos corporales de la detección. Sin embargo, la cantidad de cambios de identidad si resulta fundamental, porque este tipo de error incrementa el número de personas diferentes vistas, afectando considerablemente el tiempo de permanencia de cada una de ellas. Por esto entonces nos centraremos en analizar la cantidad de cambios de detección. Una baja cantidad de estos cambios nos permitiría lograr un análisis fiel a la realidad.

4.2.1. Procedimiento de evaluación

En primer lugar se procesaron los videos de videovigilancia, de acuerdo a lo descrito en la sección de Arquitectura. Como resultado se almacenaron, en la base de datos, las detecciones obtenidas para cada cuadro de cada video, registrando el código identificador de seguimiento de la persona a la cual pertenece cada detección. Utilizando Python y OpenCV, se generaron videos con las detecciones y códigos correspondientes a cada persona, a partir del procesamiento previo.

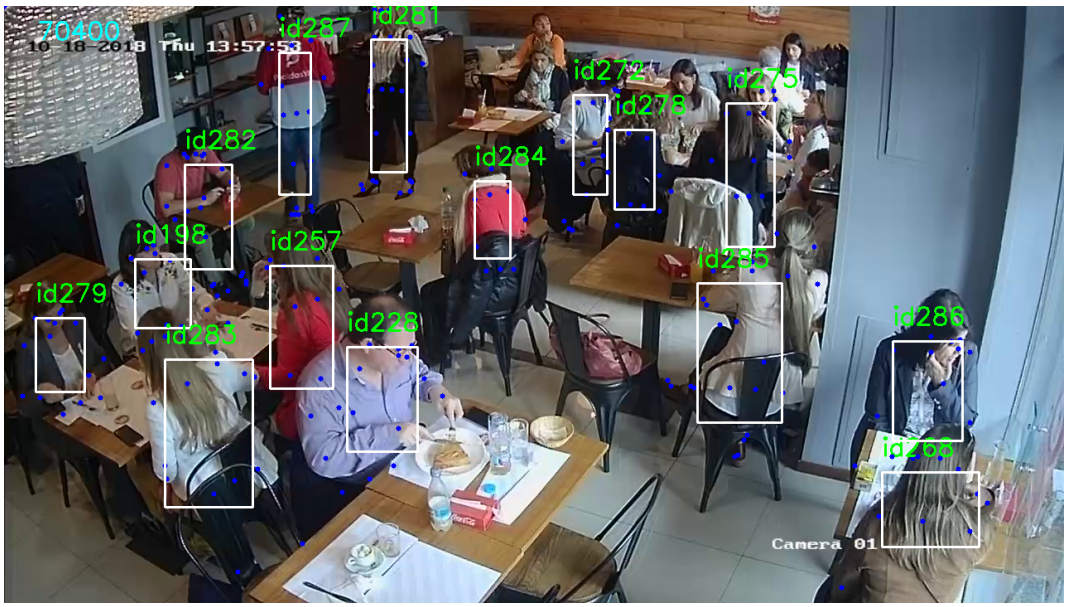

Figura 4.2: Captura de video conteniendo las detecciones y el código de seguimiento

El procedimiento de evaluación consistió en observar estos videos generados y contabilizar los cambios de identidad registrados para una misma persona. Es decir, contar la cantidad de veces que, a una misma persona, se le asignaron códigos de seguimiento distinto a lo largo de todos los videos.

Para contar con un poco más de granularidad sobre las razones de los cambios de identidad presentados, se categorizaron en:

Pérdida de detección. Cuando una persona que viene siendo identificada con un código ID_x deja de ser detectada por más de 20 cuadros (por una falla en OpenPose), su código de seguimiento se descarta. En la próxima detección de esa persona, se le asigna un nuevo código de seguimiento ID_y

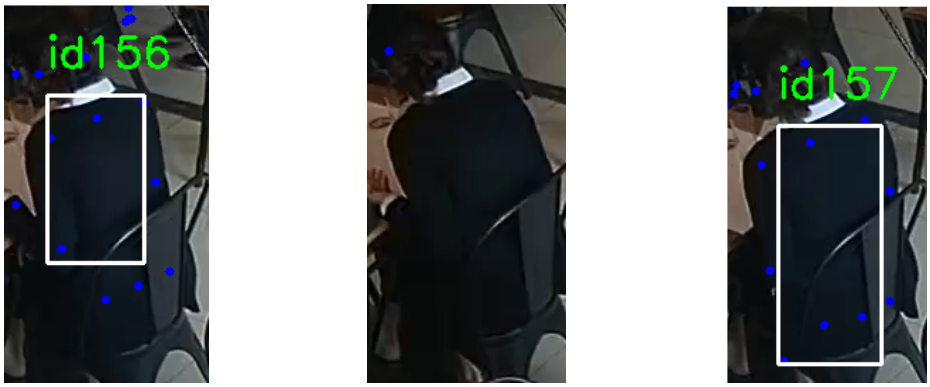

Figura 4.3: Ilustración de un cambio de identidad por pérdida de detección

Superposición. Al utilizar solamente una cámara de video, es posible que a una persona x que venía siendo identificada se le superponga total, o parcialmente, otra persona y que se posicione por delante (es decir entre la persona original y la cámara). Esto frecuentemente conlleva a que a y se le asigne el código de seguimiento de x, y a este último se le de uno nuevo.

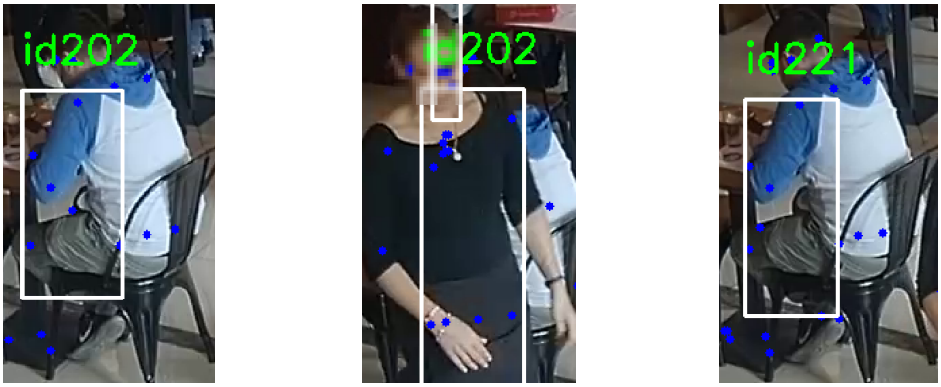

Figura 4.4: Ilustración de un cambio de identidad por pérdida de detección

Reingreso. Cuando una persona se sale del rango de filmación de la cámara, ya sea porque sale del local, o simplemente se desplaza hacia una zona que no es captada por la cámara, y vuelve, recibe un nuevo código de identificación. Esto no corresponde a un error, sin embargo implica de todas formas un cambio de identidad sobre una misma persona.

4.2.2. Resultados de la evaluación

Para evaluar la solución de acuerdo al procedimiento descrito en la sección anterior, se observaron 16 videos, uno por cada hora entre aproximadamente las 09:00 y las 17:00 de los días Jueves 18 y Lunes 22 de Octubre de 2018, y se anotaron manualmente los cambios de identidad clasificados conforme a las categorías ya mencionadas.

De acuerdo a nuestra observación, las mesas ubicadas en la parte superior de la imagen, y por tanto más alejadas de la cámara, presentaron un índice de error muy alto. Esto se debe principalmente a dos razones: poca precisión en OpenPose para detectar a las personas, por la distancia a la que están las mesas de la cámara, y la calidad de imagen de la misma; y la gran cantidad de superposiciones que se presentan debido a la cantidad de personas que transitan por el espacio medio de la sala. Por esa razón, se decidió enfocar el análisis en las 3 mesas más próximas a la cámara, numeradas como se ilustra a continuación:

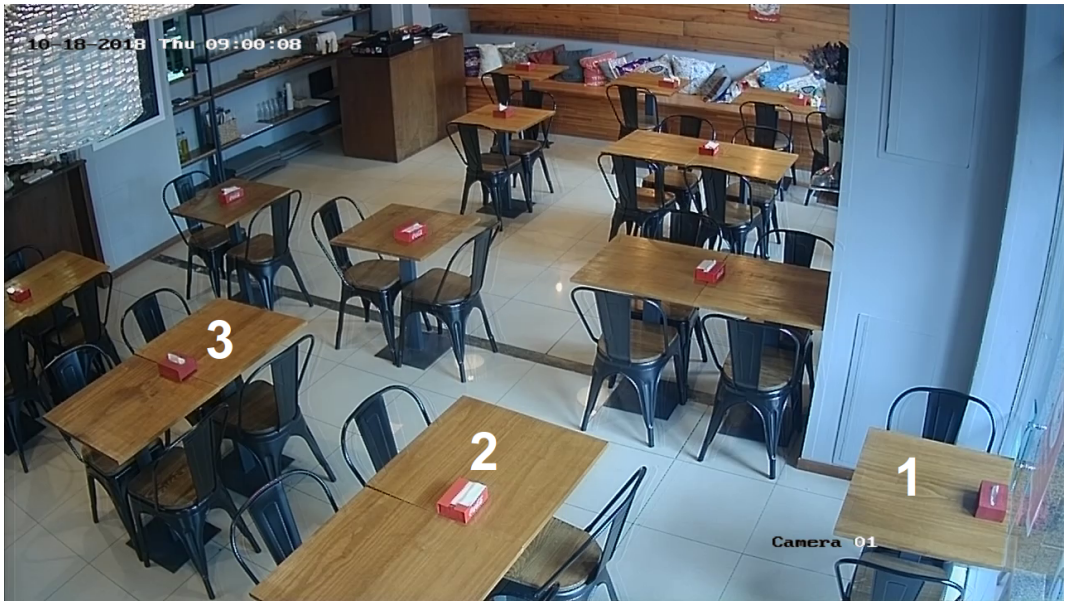

Figura 4.5: Mesas seleccionadas para la evaluación

Luego los resultados se registraron agrupados por franja horaria, como sigue:

Mañana. comprende los videos entre las 09:00 y las 12:00
Mediodía. comprende los videos entre las 12:00 y las 15:00
Tarde. comprende los videos entre las 15:00 y las 17:00

Los resultados obtenidos fueron los siguientes:

Mañana

Mesa	Pér. Det.	Super.	Reingreso	Total	Personas	P/persona
1	5	0	3	8	6	1,33
2	0	0	0	0	0	0,00
3	0	0	0	0	4	0,00
	5	0	3	**8**	**10**	**0,80**

Mediodía

Mesa	Pér. Det.	Super.	Reingreso	Total	Personas	P/persona
1	9	1	3	13	12	1,08
2	12	2	0	14	10	1,40
3	38	11	0	49	12	4,08
	59	14	3	**76**	**34**	**2,24**

Tarde

Mesa	Pér. Det.	Super.	Reingreso	Total	Personas	P/persona
1	20	2	2	24	6	4,00
2	6	0	1	7	5	1,40
3	1	0	0	1	2	0,50
	27	2	3	**32**	**13**	**2,46**

Figura 4.6: Resultados de la evaluación realizada sobre los videos del Jueves 18 de Octubre. En la tabla del mediodía se puede observar, por ejemplo, que 38 de los 49 cambios de identidad identificados en la mesa 3 se deben a que se perdió la detección de la persona que portaba esa identificación. Además, como se contabilizaron en total 12 personas sentadas en dicha mesa, y 49 cambios de identidad, se produjeron en promedio 4,08 cambios de identidad por persona (49/12)

Las tablas agrupan el resultado obtenido por mesa para cada franja horaria. Se contabilizó también la cantidad de personas sentadas en cada mesa (columna *Personas*), para luego obtener un promedio de la cantidad de cambios de identidad por persona (columna *P/persona*). Idealmente se desea que ese número sea lo más cercano a 0 posible.

Como se puede apreciar en los resultados de la evaluación, incluso enfocando el análisis en las mesas más próximas a la cámara, se presentan muchos errores que afectan directamente la consistencia de las métricas de permanencia obtenidas, tomando como referencia el escenario real. Para medir el impacto producidos por estos errores, corregimos manualmente los cambios de identidad detectados, para luego comparar las métricas obtenidas sobre una base de datos corregida con una sin corregir.

Mañana

Mesa	Pér. Det.	Super.	Reingreso	Total	Personas	P/persona
1	2	1	1	4	2	2,00
2	0	0	0	0	3	0,00
3	0	0	0	0	0	0,00
	2	1	1	**4**	**5**	**0,80**

Mediodía

Mesa	Pér. Det.	Super.	Reingreso	Total	Personas	P/persona
1	0	0	1	1	4	0,25
2	1	1	1	3	6	0,50
3	13	1	1	15	8	1,88
	14	2	3	**19**	**18**	**1,06**

Tarde

Mesa	Pér. Det.	Super.	Reingreso	Total	Personas	P/persona
1	0	0	0	0	3	0,00
2	0	0	0	0	0	0,00
3	0	0	3	3	5	0,60
	0	0	3	**3**	**8**	**0,38**

Figura 4.7: Resultados de la evaluación realizada sobre los videos del Lunes 22 de Octubre

Generación de base de datos sin errores

Como se describió en el capítulo anterior, nuestro prototipo almacena en una base de datos no relacional las detecciones generadas por OpenPose para cada persona y para cada cuadro de cada video, junto al código de seguimiento asignado a dicha persona. El procedimiento de corrección consistió entonces en modificar esos registros de la base de datos, como resultado de la observación manual de los videos y las detecciones generadas durante el procesamiento del módulo de análisis de nuestro prototipo.

Se tienen entonces dos bases de datos: una con las detecciones e identidades asignadas de forma automática durante la detección y el seguimiento, y otra corregida, resultado de aplicar las correcciones de errores en la primera. Esto permite comparar como varían las métricas obtenidas según se utilicen como insumo las detecciones automáticas o las corregidas, obteniendo una cuantificación de la distorsión que introducen en ellas los errores de cambio de identidad en el seguimiento.

Básicamente la idea para la corrección es la siguiente: supongamos que a una persona se le viene asignando un código de seguimiento ID_x, pero en el cuadro $frameInicial$ sufre un cambio de identidad (por alguna de las razones descritas en la sección anterior), y le es asignado un nuevo código ID_y que

se mantiene hasta el cuadro $frameFinal$. En este caso entonces es necesario contar con un procedimiento para modificar las detecciones en la base de datos y asignarle a las que tengan ID_y entre los cuadros $frameInicial$ y $frameFinal$ su verdadera ID_x.

Este mecanismo de cambio de identidades se implementó mediante un código externo que modifica directamente los registros en la base de datos. El procedimiento de corrección se limitó entonces a aplicar esta función para cada cambio de identidad detectado durante la observación de los videos.

Una vez corregidos los cambios de identidad, se tiene un conjunto de datos libre de este tipo de errores sobre el cual se puede trabajar y extraer métricas. Estas métricas serían los obtenidas por el sistema si el seguimiento tuviese una precisión tal que no se produjeran cambios de identidad. Es interesante de todas formas destacar que ni siquiera este conjunto de datos corregidos representa exactamente la realidad, porque para ello necesitaríamos contar con detecciones de las personas durante el 100 % de los cuadros (sin detección no es posible identificar, y por lo tanto hacer seguimiento, a una persona), lo cual no ocurre ya que OpenPose no está completamente exento de errores. Sin embargo sí representa una muy cercana aproximación.

Resultados obtenidos luego de la corrección

Comparando las diferencias en las métricas obtenidas entre el conjunto de datos obtenido por el sistema y el conjunto de datos corregido, se puede observar cuál es la diferencia que los errores de cambio de identidad introducen en el análisis realizado.

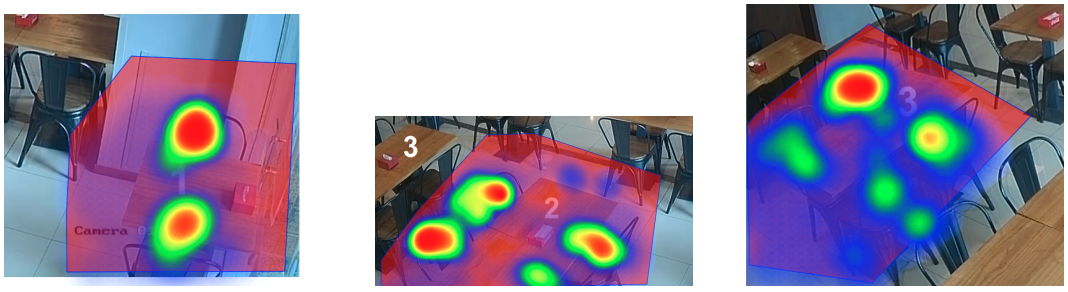

Figura 4.8: Búsquedas realizadas y los mapas de calor de las detecciones en cada una.

Se realizaron tres consultas, en cada una se delimitaba en el polígono de búsqueda una mesa diferente según la numeración indicada en la Figura 4.5.

Los mapas de calor obtenidos en cada mesa son mostrados en la Figura 4.8. Como es de esperar, muestran una mayor concentración de detecciones en los lugares que ocupan las personas cuando se sientan en cada una de las sillas. Además, se pude ver que mientras que en las mesas 1 y 2 el uso de cada silla es casi igual, no ocurre lo mismo para la mesa 3, en la que una silla es usada más frecuentemente que el resto.

Los resultados obtenidos para los tiempos de permanencia de personas en las areás de búsqueda para las distintas mesas son mostrados en forma de histogramas en las Figuras 4.9, 4.10 y 4.11. Nótese que estos histogramas no son exactamente los mismos que los mostrados en la interfaz de usuario, pero están construidos con los mismos datos y modificados para poder comparar en una misma gráfica los resultados de las detecciones automáticas y corregidas.

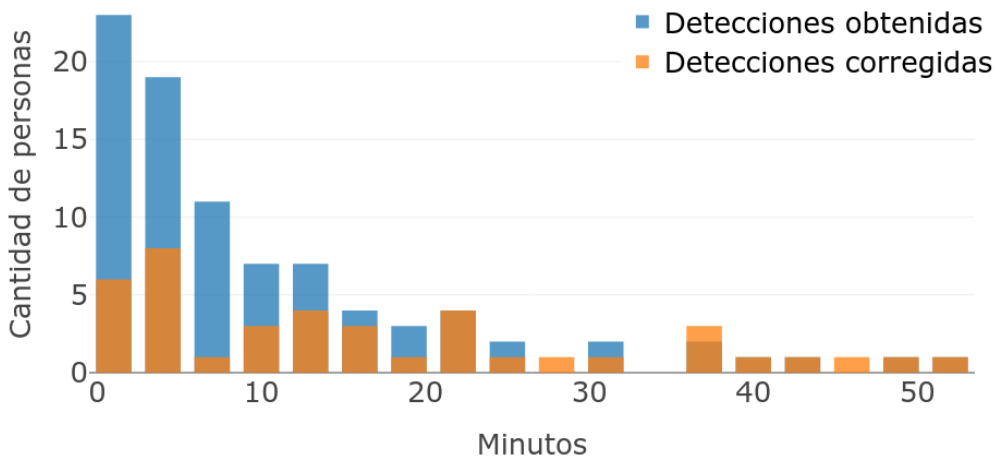

Figura 4.9: Histograma de tiempos de permanencia para la mesa 1

En los tres histogramas se puede ver que hay una gran diferencia entre los resultados obtenidos por las detecciones obtenidas y las corregidas. La corrección de datos introduce principalmente un fuerte decrecimiento en la cantidad de personas con tiempos de permanencia menores a 10 minutos, a la vez que introduce personas con tiempos de permanencia muy superiores a los registrados anteriormente.

Esta diferencia corresponde al error que los cambios de identidad en el seguimiento introducen en las métricas de permanencia. Estos errores de cambios de identidad hacen que se tengan varios seguimientos de corta duración que

	Detecciones automáticas	Detecciones corregidas
Mesa uno	11m	17m
Mesa dos	15m	29m
Mesa tres	12m	22m

Tabla 4.1: Minutos de permanencia para las personas vistas en las tres mesas para el conjunto de detecciones obtenidas de forma automática y el conjunto de datos en el que se corrigieron los erores de cambios de identidad.

en realidad corresponden a la misma persona. Al juntar estos segmentos en uno único, se tiene el tiempo de permanencia real de la persona, que será la suma todos ellos. Cuantos más cambios de identidad tenga el seguimiento, mayor será la diferencia que esta fragmentación inducirá en los tiempos de permanencia.

En la Tabla 4.1 se muestra como varía el tiempo de permanencia en las distintas mesas según se calcule la métrica con el conjunto de datos generado automáticamente o el corregido. Puede verse como la métrica varía menos en la mesa uno que en el resto de las mesas.

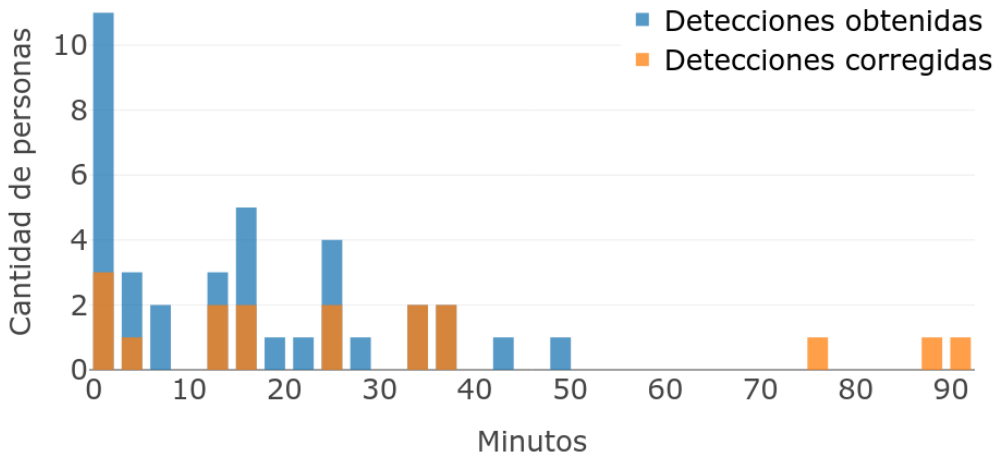

Figura 4.10: Histograma de tiempos de permanencia para la mesa 2

4.2.3. Conclusiones

Los resultados de la evaluación permiten, más allá de que se trata de un acotado a una cierta cantidad de mesas, extraer conclusiones sobre el compor-

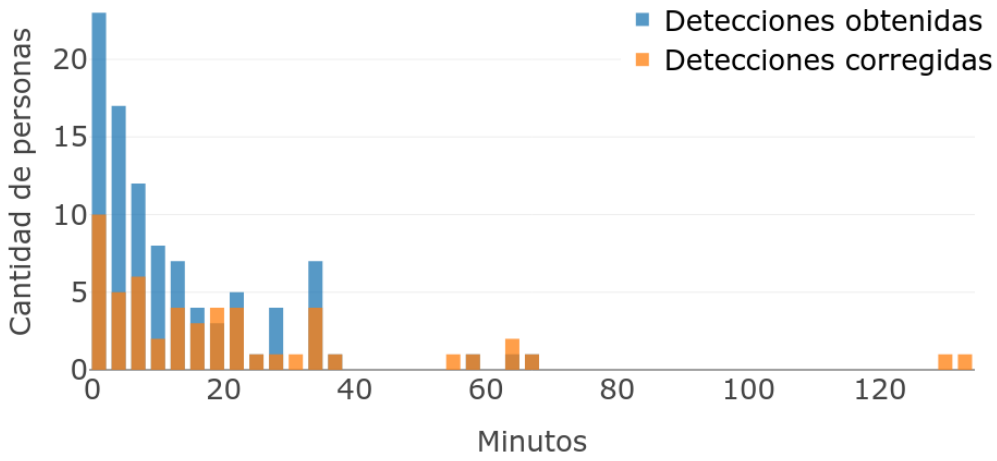

Figura 4.11: Histograma de tiempos de permanencia para la mesa 3

tamiento de los clientes en la cafetería, y la viabilidad de la aplicación a partir del prototipo.

En primer lugar resulta evidente a partir de las tablas de las figuras 4.6 y 4.7 que la gran mayoría de los cambios de identidad presentados se deben a pérdidas en la detección. Si bien es en este ítem en dónde OpenPose tiene más incidencia, es necesario aclarar que la gran mayoría de los casos se presentaron en personas sentadas en el límite del alcance de la cámara. Esto implica que ni siquiera el torso de la persona permanece visible completamente en los videos (en algunos casos solo la cabeza y el cuello), dificultando enormemente la tarea de OpenPose para detectar correctamente.

Quitando esos casos en los bordes de pantalla, los casos más frecuentes de pérdidas de detección en personas que si estaban contenidas completamente en la imagen se presentaron en personas vestidas con ropa de colores muy oscuros, cercanos al negro (como se ve en el ejemplo de la figura 4.3). Notamos una dificultad en OpenPose en estos casos.

Si comparamos mesa a mesa, vemos que en la mesa 2 se presentan menos cambios de identidad que en las 1 y 3, donde se presentan con mayor frecuencia los escenarios descritos en el párrafo anterior.

Si evaluamos con mayor granularidad dentro de cada mesa, y analizamos cada silla por separado, es posible afirmar que el prototipo sin corregir se acer-

ca considerablemente más al escenario real para aquellas sillas que permiten una detección completa frente a las que se ubican en el límite del alcance de la cámara.

Por otro lado, los cambios de identidad debido a superposiciones, si bien no se presentan en un número tan elevado como los debidos a pérdidas de detección, también tienen incidencia sobre los resultados. Esto se debe principalmente al ángulo de visión de la cámara desde la cual se filmaron los videos. Si se contara con cámaras, o se pudiese tener injerencia sobre el posicionamiento de las mismas, que enfocaran las mesas desde una posición más perpendicular, posiblemente se evitarían en gran medida estos errores.

Si nos centramos en la distribución de errores por franja horaria, resulta evidente que el momento del día más propenso a cambios de identidad es durante el mediodía. Durante la mañana y la tarde se presentan considerablemente menos errores. Esto se debe principalmente a la cantidad de personas presentes en el local en cada momento. No solo porque se cuenta con menos personas sentadas (es directo e intuitivo concluir que a menor cantidad de muestras, menores en cantidad serán los errores), sino porque durante las mañanas y tardes el tránsito de personas entre las mesas es considerablemente menor. Esto permite una detección más directa y con menor interferencia de las personas sentadas, que tiene gran incidencia en la calidad de los resultados obtenidos.

Estos cambios de identidad en el seguimiento tienen una gran incidencia en la métrica relevada en este caso, el tiempo de permanencia de las personas en el área. La asignación de múltiples identidades diferentes a una misma persona provoca que los tiempos relevados difieran bastante de los tiempos reales. Los histogramas muestran así una gran cantidad de permanencias menores a los 10 minutos, que en realidad son menos si se consideran los datos corregidos.

4.3. Búsqueda de pose

Para evaluar en los datos de prueba la potencialidad de la búsqueda de pose, se determinó primero una pose objetivo y luego se buscaron coincidencias con ella dentro del conjunto de todas las detecciones observadas en las horas de grabación disponibles. El comportamiento buscado fue la acción realizada

usualmente por los clientes de un restaurante para llamar la atención del mozo y pedirle que se dirija a la mesa, generalmente a los efectos de pedir y abonar la cuenta, ordenar, o simplemente para realizarle una consulta. Si bien existen numerosas formas de llamar la atención de un mozo, se acostumbra a levantar la mano hasta establecer un contacto visual con él.

Una vez definida la pose objetivo, el siguiente paso es determinar la cantidad de ocurrencias entre todas las detecciones disponibles. Para abordar este problema se plantearon dos enfoques posibles.

4.3.1. Búsqueda por reglas

El primero consiste en responder la pregunta ¿qué condiciones debe cumplir una pose para ser considerada *equivalente* a la pose objetivo? Es posible responder esa pregunta para la pose objetivo en base a las condiciones que deben cumplir las coordenadas del hombro, codo y muñeca de una persona (es necesario recordar que estos tres puntos son los únicos que detecta OpenPose con el modelo de cuerpo utilizado en el prototipo). Se puede establecer que una persona tiene la mano alzada (izquierda o derecha), si su codo está por encima (con respecto a la coordenada en el eje Y) de su hombro, y su muñeca está por encima de su codo.

Para detectar las poses en las que el brazo está totalmente extendido, se agregó la condición de que la coordenada correspondiente al codo esté a una altura superior a 1,5 veces la distancia entre el punto medio de la cabeza y el cuello. Esta heurística fue hallada de forma experimental, en pos de ajustar el método para que logre detectar correctamente la mayor cantidad de ocurrencias reales observadas.

4.3.2. Búsqueda por semejanza

Otro posible enfoque consiste en identificar, entre las detecciones, las poses *similares* a la objetivo. En este caso decimos que dos poses son similares si la distancia entre los pares de puntos de igual tipo para ambas poses está por debajo de un deteminado umbral.

Es fundamental entonces determinar que pares de puntos deben ser considerados para medir la similaridad de una pose dada. En algunos casos puede ser importante considerar las piernas de una persona, mientras que en otro la pose objetivo puede estar determinada en mayor medida por el torso y la

cabeza. Por esto, la selecciónń de pose objetivo es realizada partiendo de una pose observada, pero marcando en ella los puntos que son de interés para el criterio de semejanza.

Con estas consideraciones, se implementó un método para determinar la medida de semejanza que se basa en la suma de la distancia euclídea entre los pares de puntos corporales que están presentes tanto en la pose estudiada como en la pose objetivo (es decir, que fueron marcados en ella inicialmente como de interés). Ya que la similitud de pose es independiente de su posición en la imagen, las coordenadas de sus puntos son centradas en el centro de cada un antes de compararlas. Su pseudocódigo puede verse en el algoritmo 2.

Data: Pose estudiada P^x, Pose objetivo P^{obj}, Umbral k
Result: TRUE si P^x y P^{obj} son similares, FALSE si no
Distancia $d = 0$;
foreach *Punto corporal* i **do**
  if $\exists P_i^x, P_i^{obj}$ **then**
    $d = d + \left\| P_i^x - P_i^{obj} \right\|$
  end
end
if $d <= k$ **then**
  Return TRUE
else
  Return FALSE
end

Algorithm 2: Criterio de semejanza

A la hora de comparar ambos enfoques, podemos decir que si bien el método basado en reglas permite refinar el criterio de búsqueda con el grado de especificidad que se desee, requiere escribir una rutina específica para cada pose objetivo que se defina. Esto hace que su uso no resulte generalizable. El método basado en semejanza puede ser aplicado a cualquier pose objetivo sin tener que realizar ningún ajuste más que el de indicar una pose modelo. Con la potencialidad de poder responder a una consulta generada por una pose definida por el usuario a través de una plataforma como la diseñada para nuestro trabajo, resulta ademaś de especial interés.

4.3.3. Procedimiento de evaluación

Para estudiar el rendimiento de estos algoritmos de búsqueda de pose, primero se relevaron manualmente todos los casos de éxito a detectar. Para esto se revisaron las horas de video disponibles, a los efectos de detectar los momentos en que alguna persona fue vista levantando su mano para llamar la atención del mozo. Teniendo estos números, y suponiendo que hay evaluar de los métodos:

precisión, es decir, de las instancias detectadas, cuales efectivamente son correctas: $\frac{vp}{vp+fp}$

recall o exhaustividad, es decir, de las instancias correctas (identificadas manualmente), cuantas detectó: $\frac{vp}{vp+fn}$

siendo $tp = VerdaderoPositivo$, $fp = FalsoPositivo$, y $fn = FalsoNegativo$

Se estableció como criterio de aceptación que la persona levante cualquiera de sus brazos mientras dirige su mirada al mozo, y que además efectivamente el intento tenga éxito y el mozo se dirija hacia su mesa en la brevedad. Esto último no es detectable automáticamente por los algoritmos, pero representa sin embargo el caso de uso que se procura identificar.

En la figura 4.12 se pueden ver ejemplos de los casos identificados oportunamente.

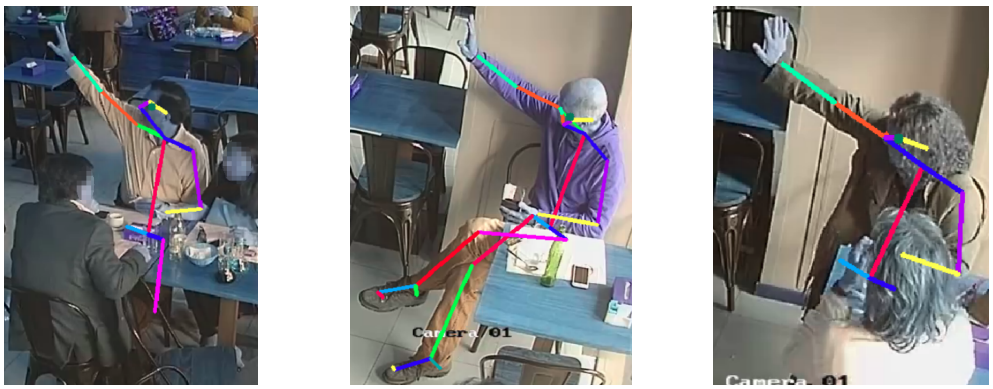

Figura 4.12: Ejemplos de la pose objetivo buscada

Con esto se logra obtener el conjunto de detecciones *correctas*, a los efectos de poder comparar el rendimiento de los dos algoritmos de búsqueda, que deberán maximizar la cantidad de poses detectadas correctamente y minimizar las falsas detecciones.

El resultado de la clasificación de la pose se evalúa por cada cuadro de la secuencia de video según sea correcta o no la detección de la pose para esa persona en ese cuadro específico. A cada detección de persona en cada cuadro se le asigna entonces una etiqueta que puede ser Verdadero Positivo (VP), Verdadero Negativo (VN), Falso Negativo (FN) o Falso Positivo (FP) según la clasificación de pose realizada y la clasificación real correcta obtenida manualmente.

Para tener más granularidad sobre las razones en las que el algoritmo detecta falsos positivos, los categorizamos en:

Error de detección. Para los casos en que OpenPose detectó erróneamente los puntos corporales, y los dispuso de forma que cumplan con los requisitos de aceptación de los algoritmos, cuando en realidad la persona estaba en una pose diferente.

Movimiento involuntario. Para los casos en que una persona efectivamente levantó uno (o ambos) de sus brazos pero no se trató de una llamada al mozo.

Perspectiva. Para los casos en que el ángulo de la cámara hace que la pose de la persona cumpla con los requisitos de aceptación, pero en la realidad no es así.

En la figura 4.13 se pueden ver ejemplos de los casos descritos.

4.3.4. Resultados obtenidos

El clasificador por regla y el clasificador por semejanza fueron aplicados al conjunto de datos y sus rendimientos obtenidos comparados para determinar cuál logra un mejor desempeño en el caso de la búsqueda de personas levantando la mano para llamar al mozo.

La clasificación realizada por semejanza depende del parámetro k que se fije para el umbral de tolerancia máximo que se permitirá para la distancia entre dos poses consideradas similares. Un menor valor de k restringirá el conjunto de poses similares a las que difieran muy poco de la pose fijada como modelo de referencia, por lo que se dejará por fuera detecciones en la acción realizada es la misma pero la configuración de la pose es distinta. Por otro lado, un mayor valor de k flexibilizará el criterio de semejanza y considerará mas poses como

Figura 4.13: Ejemplos de falsos positivos. A la izquierda se ve un error de detección, ya que OpenPose detecta incorrectamente que la persona tiene el brazo derecho levantado. La imagen del centro se corresponde con un movimiento involuntario, ya que la persona en realidad estaba desperezándose y no llamando al mozo. A la derecha se ve cómo, por la perspectiva de la cámara, la persona parece estar alzando su brazo cuando en realidad lo está extendiendo hacia adelante

semejantes a la modelo, y si bien detectará mas poses correctamente, también aumentará la cantidad de poses erróneamente clasificadas como positivas.

Para independizar la comparación de clasificadores de la elección del parámetro de umbral k, se tomaron los resultados obtenidos por hacer variar el valor en $k = \{50, 100, 250, 500, 750, 1000, 1500, 2000\}$. La sensibilidad del clasificador con respecto a esta variación en el umbral puede ser representada mediante una curva ROC (Característica Operativa Relativa), que mostrará como crece la relación entre el ratio de verdaderos positivos (VPR) frente al ratio de falsos positivos (FPR).

La curva ROC se presenta en la Figura 4.14. El clasificador por regla no es parametrizable por lo que determina un único TPR y FPR, que se visualiza en la curva como un solo punto. El rendimiento obtenido por la búsqueda por semejanza es inferior a de la búsqueda por regla, que obtiene una relación entre VPR y FPR mejor. Sobretodo, la principal debilidad de la búsqueda por semejanza es que al incrementar el umbral k, el FPR incrementa pero no así el VPR. Esto significa que el criterio de similitud es incapaz de detectar algunas instancias de la pose objetivo, debido a que no son comparables a la pose modelo por ser diferentes en cada una el tipo de articulaciones detectadas.

La clasificación por regla obtuvo una precisión de 7.6 % y una exhaustividad de 95 %. Esto significa que si bien la búsqueda fue efectiva encontrando

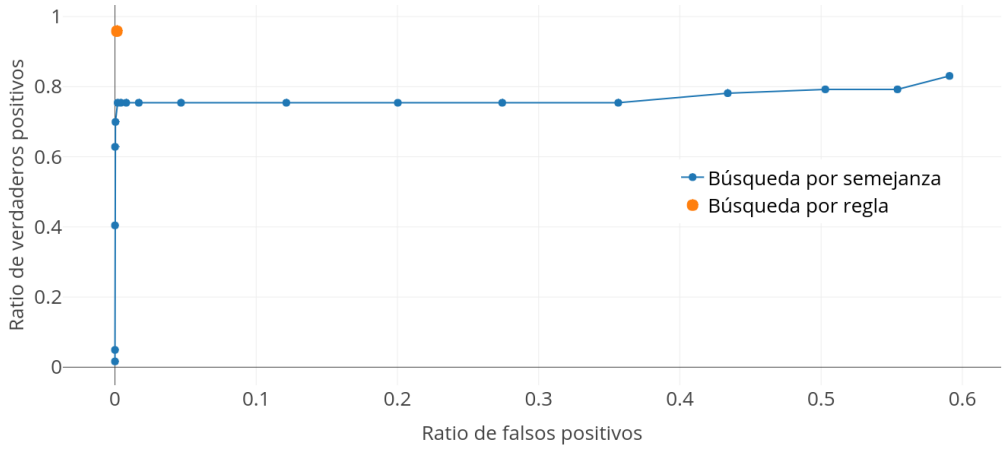

Figura 4.14: Curva ROC de los clasificadores en el conjunto de datos para la pose objetivo.

Tabla 4.2: Categorización de las clasificaciones por regla realizadas

Categoría	**Cantidad**	**% del total**
Error de detección	214	51 %
Mov. Involuntario	108	26 %
Perspectiva	77	18 %
Positivos verdaderos	18	4t %
TOTAL	417	

las detecciones correctas, también detectó incorrectamente muchas detecciones que en realidad no correspondían a la acción objetivo. Para la búsqueda por semejanza, la mayor precisión obtenida fue de 61 % para $k = 120$, pero con una exhaustividad tan solo 40 %.

Las clasificaciones realizadas por el clasificador por regla fueron relevadas manualmente y sus errores categorizados en los tipos mencionados en la sección anterior. La cantidad de errores de cada categoría puede verse en la Tabla 4.2. El conteo se realiza en base a las acciones detectadas, no a la cantidad de frames clasificados. Es decir, si un conjunto seguido de cuadros fueron clasificados como la acción objetivo pero correspondían a un movimiento involuntario, sumarán uno a la cantidad de ese error independientemente de la cantidad de cuadros involucrados.

Si bien solo el 4 % de detecciones de la acción objetivo es realizado correctamente, la clasificación de errores permite entender por qué este número es tan bajo. Cerca de la mitad de errores se deben a errores en la detección de la pose realizada por OpenPose, un error independiente del algoritmo de búsqueda de pose utilizado.

Los errores de perspectiva son del 18 % resultado de que la perspectiva con el plano de imagen varía mucho entre las distintas mesas observadas. El número podría ser reducido si la cámara se coloca en una posición en la que se tenga la misma perspectiva con todas las posibles ubicaciones de personas, y si se adecúa el algoritmo de búsqueda a esta perspectiva.

Finalmente los errores debidos a las ocasiones en que la persona levanta la mano por arriba de su cabeza pero no para llamar al mozo corresponden al 26 %. En estos casos el detector identificó correctamente la pose, pero no la semántica de su intención verdadera. Este error podría ser reducido si se le proporciona mayor información al detector, como por ejemplo el conjunto de poses observadas anteriormente en un intervalo dado (con lo que se podría buscar no solo la pose de brazo sino el movimiento entero), la dirección de su mirada, etc.

Si bien el rendimiento del clasificador por regla es aceptable y logra de hecho detectar las poses en las que la mano es levantada por encima de la cabeza (independientemente de la intención), la necesidad de programar un criterio de selección para cada pose objetivo es incompatible con el objetivo de generalidad perseguido por la plataforma propuesta. La búsqueda por semejanza implementada, si bien es compatible con este requerimiento, logra un

rendimiento inferior.

Capítulo 5

Conclusiones y trabajos a futuro

La propuesta de este trabajo es evaluar la viabilidad de una solución que logre analizar el comportamiento de las personas en un espacio físico a través de la detección y el seguimiento de las mismas por intermedio de cámaras de video. La premisa es apoyarse en métodos del estado del arte vigente, y utilizar solamente cámaras de videovigilancia, prescindiendo de hardware adicional. Esto simplifica en gran medida la implementación de la solución y contribuye a lograr un alto grado de adopción por parte de potenciales clientes interesados, pero también limita considerablemente las herramientas de trabajo para lograr un análisis preciso. En base a los resultados obtenidos y el estudio presentado en las secciones precedentes, en éste capítulo se habla sobre las conclusiones que se extraen de la aplicación de una solución con éstas características.

Para evaluar la viabilidad se planteó un prototipo basado en los dos pilares fundamentales de la solución: la detección de las personas, y el seguimiento de las mismas. Como conjunto de datos se utilizaron grabaciones provenientes de un sistema de videocámaras de vigilancia de una cafetería. No se tuvo injerencia sobre el posicionamiento de las cámaras ni del mobiliario, por lo que se trata de una situación que representa un ambiente no controlado, como se buscaba.

Los videos fueron procesados con el sistema, usando OpenPose para la detección de personas y DeepSort para el seguimiento. La calidad del seguimiento realizado fue relevada revisando los videos en busca de errores de cambio identidad, contabilizando su ocurrencia y corrigiéndolos manualmente, generando

así un conjunto de datos paralelo libre de este tipo de errores.

En total se contabilizaron 142 errores de cambio de identidad para un total de 88 personas vistas. Se detecta una variablidad entre las distintas mesas producto de la disposición de la cámara hacia ellas. En particular, la mesa 2, al ser la más cercana a la cámara y obtenerse una vista superior de ella, registra una cantidad de errores menor a los de las distintas mesas, que o bien no se encuentran totalmente incluidas en el plano de la imagen o bien la perspectiva que se tiene a ellas causa mayores errores de oclusión entre personas.

La comparación entre la métrica de tiempo de permanencia obtenida en el conjunto de datos obtenido y el corregido arrojó diferencias sustanciales, expuestas en los histogramas de las Figuras 4.9, 4.10 y 4.11, evidenciando la alta sensibilidad que presenta la métrica frente a los errores de cambio de identidad en las personas detectadas. Al no poder mantener con robustez la asignación de identidad a las personas a lo largo del video, no se puede determinar con precisión el tiempo de permanencia de ellas. Poder resolver con precisión el problema del seguimiento de personas es una actividad clave para la utilidad del sistema propuesto, ya que muchas de las métricas que puede ser de interés requieren conocer el comportamiento de una persona a lo largo de toda su estadía en el lugar.

El análisis de los errores de cambios de identidad reveló que en la mayor parte de los casos sus razones correspondían a la pérdida de la detección de la persona durante algunos cuadros del video, o a movimientos bruscos que causaran un gran cambio en su caja delimitadora. En estos casos, el filtro de Kalman utilizado da una predicción de la siguiente ubicación muy distinta a la real, lo que introduce errores en la asignación de nuevas detecciones a los seguimientos en curso.

Dado que Deep SORT recurre a la predicción de ubicación por el filtro de Kalman en las situaciones en las que no es posible asociar las nuevas detecciones a los seguimientos mediante la información de apariencia, una forma de lograr mejor performance en el seguimiento es mejorar la calidad de la descripción de apariencia de manera que permita una mayor cantidad de recuperación de identidades. El rendimiento de los algoritmos de seguimiento propuestos ha seguido en aumento desde la publicación de Deep SORT y los que mejores resultados obtienen hacen un uso intensivo de información de apariencia para re identificar personas vistas anteriormente en otros cuadros [38]. Utilizar en la plataforma un algoritmo de seguimiento más reciente que Deep SORT

podría incrementar considerablemente la performance y minimizar la cantidad de cambios de identidad experimentados.

Dejando de lado el seguimiento, se puede afirmar que el rendimiento de OpenPose en el módulo de detección de personas fue bueno. Más allá de algunos detalles identificados, como la dificultad que presentó para detectar correctamente a algunas personas vestidas con colores oscuros, si se tiene en cuenta que se vale de una sola cámara para generar las detecciones, obtuvo resultados aceptables en términos generales, y muy buenos para las personas más cercanas a la mesa. No demostró además ser sensible a la calidad de las filmaciones, ni a los cambios de iluminación presentados en la cafetería a lo largo del día.

Las categorización de puntos corporales de OpenPose fue utilizada para realizar una búsqueda en las horas de grabación de todas las ocurrencias de una acción objetivo, la de un cliente levantando la mano para llamar la atención del mozo. Se relevaron manualmente los videos disponibles y se etiquetaron los cuadros en los que alguna persona estaba realizando la acción. La acción fue buscada a través de la pose mediante dos algoritmos diferentes: una búsqueda por regla que detectaba una pose como positiva si el brazo estaba extendido sobre la cabeza de la persona, y otra búsqueda por semejanza que a partir de una pose modelo y un umbral, detectaba como positiva una pose dada si difería de la modelo en una distancia menor que el umbral.

Los algoritmos de búsqueda implementados lograron buenos resultados teniendo en cuenta su simplicidad. La búsqueda por regla logró obtener un alto valor de exhaustividad, 95 %, aunque con una precisión de tan solo 7.6 %. La búsqueda por semejanza obtuvo un desempeño algo peor (61 % de precisión y 40 % de exhaustividad para $k = 120$), y experimentó problemas en detectar algunas poses aún cuando su umbral era incrementado considerablemente.

Relevando manualmente la razón de los falsos positivos del detector por regla, se obtuvo que cerca de la mitad de las detecciones incorrectas son debidas a errores de OpenPose. Otro 18 % de errores es debido a la perspectiva de la cámara con la persona, que varía mucho a lo largo de la imagen. Estos errores podrían ser mitigados si en lugar de considerar la pose bidimensional en el plano de imagen, se trabajara con un modelo tridimensional unificando su representación independientemente de la perspectiva que se tenga con la cámara. El problema de estimar una representación 3D de la pose a partir de la ubicación 2D de articulaciones fue recientemente enfrentado con éxito, y podría ser incorporado a la plataforma para mejorar la performance [26].

Otra gran parte de los errores cometidos en la detección de pose, el 25 %, se atribuye a detecciones en las que la persona levantó de hecho el brazo por sobre la cabeza, pero no para llamar la atención del mozo sino que como parte de una gesticulación en una charla, un bostezo, etc. Si bien la pose es correctamente detectada, no es parte de la acción objetivo. Una posible forma de incorporar mayor información semántica a la búsqueda es describir las acciones como movimientos, como una secuencia de poses en lugar de como una sola. El reconocimiento de acciones en videos es un problema tratado con atención últimamente, para el que las soluciones del estado de arte utilizan redes neuronales convolucionales o recurrentes, tanto a nivel de imagen [11] [28] como de esqueletos corporales [51] [24]. Extender la búsqueda de poses a búsqueda de movimientos captaría mejor la semántica de la acción, y sobretodo daría al usuario más posibilidades de búsqueda y de aplicación.

Dadas las limitaciones actuales experimentadas en el prototipado del sistema, una forma de obtener un mejor funcionamiento sería relajar la restricción de su uso a ambientes no controlados. Tener control sobre el escenario, la cantidad de cámaras su ubicación y enfoque, podría mejorar considerablemente la calidad del seguimiento y de la búsqueda de poses. Es notorio en base a los resultados presentados en las figuras 4.6 y 4.7 que el rendimiento decrece en las horas altamente concurridas, básicamente porque el tránsito de las personas genera interferencia en las detecciones. Utilizar más cámaras y dedicar cada una de ellas al análisis de un espacio limitado (por ejemplo una sola mesa), permitiría ubicarlas de forma que el ángulo de visión minimice la superposición entre las personas en el plano de imagen.

Incrementar la cantidad de cámaras introduce la necesidad de soportar el manejo y tratamiento de múltiples cámaras en la plataforma. Así, el seguimiento de personas debería de ser capaz de identificar a una misma persona captada en diferentes momentos en diferentes escenas. Esto permitiría abarcar en el rango de visión de las cámaras la totalidad del espacio físico independientemente de su tamaño y arquitectura, pudiendo ser aplicable a una mayor cantidad de casos de uso. Soluciones propuestas al problema de seguimiento multicámara pueden encontrarse en [38] y [54].

Los problemas de visión artificial de cuya solución depende la plataforma propuesta reciben, como se ve en estas conclusiones, gran interés actualmente y su estado del arte está en continuo perfeccionamiento registrando cada año mejores resultados en los diferentes desafíos propuestos. Así, es de esperar que

en los próximos años estén disponibles nuevas técnicas y algoritmos similares a los utilizados en este prototipo pero de mejor rendimiento, que permitan una detección y seguimiento eficiente de personas en videos. Con ellos, una plataforma de características similares a la estudiada en este trabajo sería tecnológicamente posible, y sin duda sería de interés su uso en lugares que vean una propuesta de valor en conocer el comportamiento de su público visitante.

Referencias bibliográficas

[1] Amazon (2018). Amazon Go. http://www.amazongo.com/. Accedido: 07-11-2018.

[2] Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2d human pose estimation: new benchmark and state of the art analysis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[3] Angel List (2018). Angel list computer vision startups. https://angel.co/computer-vision. Accedido: 07-11-2018.

[4] Avidan, S. (2004). Support vector tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(8):1064–1072.

[5] Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: The clear mot metrics. *J. Image Video Process.*, 2008:1:1–1:10.

[6] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. *CoRR*, abs/1602.00763.

[7] Brickstream (2018). Brickstream. http://www.brickstream.com/. Accedido: 07-11-2018.

[8] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

[9] Cisco (2018). Cisco Meraki. https://meraki.cisco.com. Accedido: 07-11-2018.

[10] Cristopher Donnelly, Renato Scaff (2018). Who are the Millenial shoppers? And what do they really want? https://goo.gl/JXfBH4. Accedido: 07-11-2018.

[11] Ding, L. and Xu, C. (2017). Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation. *CoRR*, abs/1705.07818.

[12] Edmonds, J. and Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264.

[13] Fang, H., Xie, S., and Lu, C. (2016). RMPE: regional multi-person pose estimation. *CoRR*, abs/1612.00137.

[14] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *IJCV*.

[15] Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1):67–92.

[16] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852.

[17] Inmarket (2018). Inmarket. https://inmarket.com/. Accedido: 07-11-2018.

[18] Irisys (2018). Irisys. http://www.irisys.com/. Accedido: 07-11-2018.

[19] Kalman, R. E. and Others (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

[20] Karen Simonyan, A. Z. (2015). Very deep convolutional networks for large-scale image recognition. *ICLR*.

[21] Kasey Lobaugh, Bobby Stephens, Christina Bieniek, Preeti Pincha (2018). The great retail bifurcation. Why the retail apocalypse is really a renaissance. https://www2.deloitte.com/content/dam/insights/us/articles/4365_The-great-retail-bifurcation/DI_The-great-retail-bifurcation.pdf. Accedido: 07-11-2018.

[22] Keuper, M., Tang, S., Yu, Z., Andres, B., Brox, T., and Schiele, B. (2016). A multi-cut formulation for joint segmentation and tracking of multiple objects. In *arXiv:1607.06317*.

[23] Kim, C., Li, F., Ciptadi, A., and Rehg, J. M. (2015). Multiple hypothesis tracking revisited. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4696–4704, Washington, DC, USA. IEEE Computer Society.

[24] Kim, T. S. and Reiter, A. (2017). Interpretable 3d human action analysis with temporal convolutional networks. *CoRR*, abs/1704.04516.

[25] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

[26] Kudo, Y., Ogaki, K., Matsui, Y., and Odagiri, Y. (2018). Unsupervised adversarial learning of 3d human pose from 2d joint locations. *CoRR*, abs/1803.08244.

[27] Kuhn, H. W. and Yaw, B. (1955). The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97.

[28] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1012.

[29] M. Everingham, L. Van Gool, C. W. J. W. and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *In. J. Computer Vision*, 88:303–338.

[30] Marketing charts (2016). What's Mobile's Influence In-Store? https://www.marketingcharts.com/industries/retail-and-e-commerce-65972. Accedido: 07-11-2018.

[31] Mihai Fieraru, Anna Khoreva, L. P. B. S. (2018). Learning to refine human pose estimation. *CoRR*, abs/1804.07909.

[32] Milan, A., Rezatofighi, S. H., Dick, A. R., Schindler, K., and Reid, I. D. (2016). Online multi-target tracking using recurrent neural networks. *CoRR*, abs/1604.03635.

[33] Newell, A. and Deng, J. (2016). Associative embedding: End-to-end learning for joint detection and grouping. *CoRR*, abs/1611.05424.

[34] Ondruska, P. and Posner, I. (2016). Deep tracking: Seeing beyond seeing using recurrent neural networks. *CoRR*, abs/1602.00991.

[35] P. Felzenswalb, R. Girshick, D. M. and Ramanan, D. (2013). Visual object dectection with deformable part models. *Communications of the ACM*, 56(9):97–105.

[36] PyKalman (2018). Pykalman, the dead-simple kalman filter, kalman smoother, and em library for python. https://pykalman.github.io/. Accedido: 11-11-2018.

[37] Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497.

[38] Ristani, E. and Tomasi, C. (2018). Features for multi-target multi-camera tracking and re-identification. *CoRR*, abs/1803.10859.

[39] S.-E. Wei, V. Ramakrishna, T. K. and Sheikh, Y. (2016). Convolutional pose machines. *CVPR*.

[40] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117.

[41] Smith, K., Gatica-Perez, D., and Odobez, J.-M. (2005). Using particles to track varying numbers of interacting people. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:962–969 vol. 1.

[42] Standard Cognition Corporation (2018). Standard Cognition. https://standard.ai/. Accedido: 07-11-2018.

[43] Swirl (2018). Swirl. http://www.swirl.com/. Accedido: 07-11-2018.

[44] T. S. Huang (1996). Computer Vision: Evolution and Promise. https://cds.cern.ch/record/400313/files/p21.pdf/. Accedido: 07-11-2018.

[45] T.-Y. Lin, M. Maire, S. B. J. H. P. P. D. R. P. D. and Zitnick, C. L. (2014). Microsoft coco: common objects in context. *ECCV*.

[46] Welch, G. and Bishop, G. (1995). An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA.

[47] West, D. B. (2001). Introduction to graph theory. *Prentice hall Upper Saddle River*, 2.

[48] Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402.

[49] Wojke, N., Bewley, A., and Paulus, D. (2018). Simple online and realtime tracking with a deep association metric. https://github.com/nwojke/deep_sort.

[50] Xovis (2018). Xovis. http://www.xovis.com/. Accedido: 07-11-2018.

[51] Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *CoRR*, abs/1801.07455.

[52] Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., and Yan, J. (2016). POI: multiple object tracking with high performance detection and appearance feature. *CoRR*, abs/1610.06136.

[53] Yu Xiang, A. A. and Savarese, S. (2015). Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4705–4713.

[54] Zhang, Z., Wu, J., Zhang, X., and Zhang, C. (2017). Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *CoRR*, abs/1712.09531.

[55] Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., and Tian, Q. (2016). Mars: A video benchmark for large-scale person re-identification.