Detección de personas en imágenes de profundidad mediante redes neuronales convolucionales

Roberto Martín López
Escuela Politécnica Superior
Universidad de Alcalá
Alcalá de Henares (Madrid)
Email: roberto.martin@edu.uah.es

David Fuentes Jiménez
Escuela Politécnica Superior
Universidad de Alcalá
Alcalá de Henares (Madrid)
Email: d.fuentes@edu.uah.es

Cristina Losada Gutiérrez
Escuela Politécnica Superior
Universidad de Alcalá
Alcalá de Henares (Madrid)
Email: cristina.losada@uah.es

Carlos Luna Vázquez
Escuela Politécnica Superior
Universidad de Alcalá
Alcalá de Henares (Madrid)
Email: carlos.luna@uah.es

***Abstract*—En este trabajo se presenta un sistema que realiza la detección de personas utilizando únicamente la información de profundidad proporcionada por una cámara RGB-D ubicada en posición frontal. La solución propuesta se basa en una red neuronal convolucional (CNN) con una arquitectura encoder-decoder, formada por bloques residuales de *ResNet*, que han sido ampliamente empleadas en detección y clasificación. El sistema parte de un mapa de profundidad adquirido por un sensor que puede estar basado en tiempo de vuelo (ToF) o en luz estructurada, y genera a su salida un mapa de confianza en el que cada persona detectada se representa mediante una función gaussiana, cuya media corresponde a la ubicación en coordenadas 2D de la cabeza de la persona en la imagen. Una vez generado este mapa, se aplican técnicas de refinado de hipótesis para mejorar la precisión de la detección. Para el entrenamiento del sistema se han utilizado únicamente imágenes de profundidad sintéticas, generadas mediante el programa de diseño *Blender*, evitando la necesidad de contar con grandes bases de datos reales. Para la evaluación se han utilizado datos reales adquiridos con una Kinect II. Además, se han comparado los resultados obtenidos con otros trabajos del estado del arte, comprobando que los resultados son similares a pesar de no haber incluido datos reales en el entrenamiento.**

I. Introducción

La detección de personas ha ido cobrando importancia a lo largo de los últimos años en el ámbito científico, debido a su aplicación en múltiples áreas como el control de aforos, la seguridad, vídeo-vigilancia, etc. En la mayoría de trabajos previos [1, 2] se realiza la detección a partir de imágenes de color. En [1], se propone un sistema que aprende modelos de apariencia de las personas, mientras que el trabajo [2] se basa en clasificación de puntos de interés. Los sistemas que utilizan información de color pueden presentar problemas relacionados con la privacidad, ya que la información disponible en una imagen permite reconocer la identidad de las personas que aparecen en ella. Como alternativa de cara a solucionar dichos inconvenientes han aparecido sistemas [3–6], como el utilizado en este trabajo, que emplean sensores de profundidad (2.5D) [7, 8], que proporcionan información de la distancia a cada punto de la escena, permitiendo detectar personas pero no identificarlas (reconocer su identidad).

En los últimos años, la utilización de sistemas basados en *Deep Learning* ha aumentado de forma muy notable tanto en sistemas basados en imágenes RGB o de color como en sistemas que combinen esa información con la de el mapa de profundidad RGB-D. Otros trabajos, como [9, 10] emplean únicamente la información de profundidad, igual que en el caso del sistema presentado.

Un aspecto importante a destacar es que, en la mayor parte de trabajos previos mencionados, que utilizan sensores de profundidad, ubican éstos en posición cenital, evitando el problema de las oclusiones, pero abarcando un área de estudio que en muchas aplicaciones puede resultar demasiado reducida. De cara a aumentar el área de estudio, en este trabajo se plantea una ubicación frontal elevada de la cámara. En la Figura 4 puede observarse la perspectiva de dicha ubicación frontal elevada. Uno de los principales problemas a solventar en esta posición son las oclusiones, las cuales deberá absorber el algoritmo para proveer de una detección robusta.

Este trabajo propone un sistema de detección que utiliza una red neuronal convolucional (CNN) para la detección robusta de múltiples personas en imágenes de profundidad con ubicación frontal elevada de la cámara. El sistema se ha entrenado de forma *end-to-end* utilizando imágenes sintéticas y las correspondientes salidas se han anotado de forma automática, con una gaussiana cuya media se sitúa sobre la posición de la cabeza de cada persona. La Figura 1 muestra un ejemplo de la imagen sintética de entrada, así como de la salida anotada. Para evaluar el funcionamiento del sistema entrenado se han utilizado en primer lugar, datos sintéticos, y posteriormente datos reales. Además, se han comparado los resultados con datos reales con los de otras propuestas del estado del arte.

Figura 1. Imagen de entrada e imagen de salida anotada

La estructura del artículo es la siguiente: la Sección I realiza una introducción y una revisión del estado del arte, la Sección II expone la arquitectura propuesta, la Sección III explica el entrenamiento, la Sección IV incluye los resultados y la Sección V contiene las conclusiones obtenidas del trabajo.

II. Arquitectura del Sistema

Como ya se ha comentado en la introducción, se propone un sistema basado en una CNN que procese las imágenes de profundidad de entrada y entregue a la salida un mapa de confianza que debe contener tantas detecciones como personas existan en la imagen. Dicho mapa de confianza tiene las mismas dimensiones que la imagen de entrada (240×320 píxeles) y su aspecto se muestra en la Figura 1, donde puede verse como cada detección se indica en el mapa de confianza con una función gaussiana alrededor de la posición 2D de la cabeza de la persona detectada. Una salida mediante un mapa de confianza del mismo tamaño que la imagen de entrada permite una mejor definición del problema y la inmunización temporal con respecto al número de personas detectadas en la imagen, es decir, la velocidad de ejecución del sistema no depende del número de personas que éste detecte.

Con un enfoque desde el exterior hacia el interior de la red, el primer paso es definir y explicar los dos bloques que forman el sistema, el *Bloque Principal* (BP) y el *Bloque de Refuerzo de Hipótesis* (BR), como se muestra en la Figura 2.

Ambos bloques se basan en una estructura *encoder-decoder* que posteriormente se describe más a fondo. La imagen de entrada es procesada por el BP, genera a su salida un primer mapa de confianza. Esta salida se concatena con la imagen de entrada, generando una matriz de $240 \times 320 \times 2$ que es procesada por el BR. El mapa de confianza que genera a su salida el BR es la salida definitiva del sistema. El BR mejora la detección que el BP realiza por sí solo, entregando un mapa de confianza con gaussianas mejor definidas y reduciendo la aparición de falsos positivos que puede generar el BP.

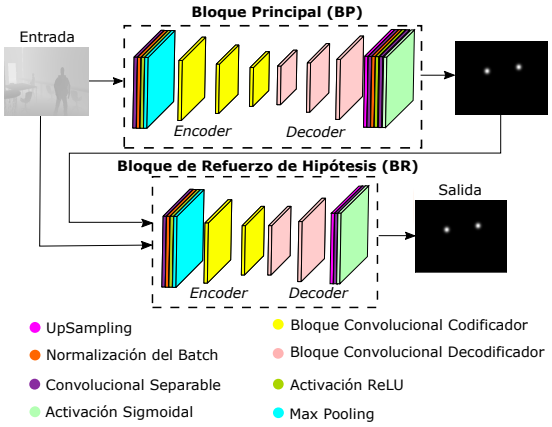

Figura 2. Arquitectura del Sistema Completo.

La estructura interna del BP es un *encoder-decoder* basado en el modelo *ResNet* [11] y que hace uso de capas convolucionales separables basadas en las utilizadas en [12]. Se han elegido este tipo de capas convolucionales debido a que son un tipo de convolución especial mucho más rápida que las capas convencionales pero que al mismo tiempo permite mantener una eficacia similar a la de éstas.

La Tabla I resume todas las capas del BP, indicando las dimensiones de la salida y los diferentes parámetros involucrados. Los parámetros *a*, *b* y *c* de los *Bloques Convolucionales Codificadores* (BCC) y *Bloques Convolucionales Decodificadores* (BCD), representan el número de filtros de cada capa convolucional interna.

En primer lugar, se emplea una capa convolucional separable compuesta por 64 kernels de 7×7 y con un *stride* de 2×2. Seguidamente se aplica una capa de Normalización del Batch (NB) [13], una activación ReLU y una capa de *Max Pooling* de 3×3. Posteriormente se aplican tres BCC y tres BCD, cuyas estructuras, basadas en el modelo *ResNet* [11], se detallarán más adelante. El BP, por último, contiene varias capas de *Cropping*, *ZeroPadding* y *UpSampling*, para ajustar el tamaño de salida, además de dos capas Convolucionales Separables, la primera seguida de Normalización del Batch y activación ReLU, y la última seguida de activación Sigmoidal.

Tabla I
Arquitectura detallada del Bloque Principal

Bloque Principal (BP)		
Capa	**Tamaño de salida**	**Parámetros**
Entrada	$240 \times 320 \times 1$	-
Convolución	$120 \times 160 \times 64$	kernel=(7, 7) / strides=(2, 2)
NB	-	
Activación	ReLU	
Max Pooling	$40 \times 53 \times 64$	size=(3, 3)
BCC	$40 \times 53 \times 256$	kernel=(3, 3) / strides=(1, 1) (a=64, b=64, c=256)
BCC	$20 \times 27 \times 512$	kernel=(3, 3) / strides=(2, 2) (a=128, b=128, c=512)
BCC	$10 \times 14 \times 1024$	kernel=(3, 3) / strides=(2, 2) (a=256, b=256, c=1024)
BCD	$10 \times 14 \times 256$	kernel=(3, 3) / strides=(1, 1) (a=1024, b=1024, c=256)
BCD	$20 \times 28 \times 128$	kernel=(3, 3) / strides=(2, 2) (a=512, b=512, c=128)
BCD	$40 \times 56 \times 64$	kernel=(3, 3) / strides=(2, 2) (a=256, b=256, c=64)
Cropping	$40 \times 54 \times 64$	cropping=[(0, 0) (1, 1)]
Up Sampling	$120 \times 162 \times 64$	size=(3, 3)
Convolución	$240 \times 324 \times 64$	kernel=(7, 7) / strides=(2, 2)
Cropping	$240 \times 320 \times 64$	cropping=[(0, 0) (2, 2)]
NB	-	
Activación	ReLU	
Convolución	$240 \times 320 \times 1$	kernel=(3, 3) / strides=(1, 1)
Activación	Sigmoidal	
Salida	$240 \times 320 \times 1$	-

La estructura del BR es similar a la del BP, con algunas modificaciones que reducen su tamaño. La Tabla II resume todas las capas del BP, indicando las dimensiones de la salida y los diferentes parámetros involucrados. Las capas iniciales son idénticas al BP: Convolucional Separable, Normalización del Batch, activación ReLU y *Max Pooling*. El número tanto de BCC como de BCD aplicados a continuación se reduce a dos de cada tipo. La etapa de salida es similar, dos Convolucionales, seguida la primera de Normalización del Batch y activación ReLU, y seguida la última de activación Sigmoidal. Varían las capas de *ZeroPadding*, *Cropping* y *UpSampling*, para ajustar el tamaño final de la salida.

Los BCC y los BCD tienen una estructura similar, formada por dos lazos desbalanceados, uno formado por tres capas

Tabla II
ARQUITECTURA DETALLADA DEL BLOQUE DE REFUERZO DE HIPÓTESIS

Bloque de Refuerzo de Hipótesis (BR)		
Capa	**Tamaño de salida**	**Parámetros**
Entrada	240 × 320 × 2	-
Convolución	120 × 160 × 64	kernel=(7, 7) / strides=(2, 2)
NB	-	
Activación	ReLU	
Max Pooling	40 × 53 × 64	size=(3, 3)
BCC	40 × 53 × 256	kernel=(3, 3) / strides=(1, 1) (a=64, b=64, c=256)
BCC	20 × 27 × 512	kernel=(3, 3) / strides=(2, 2) (a=128, b=128, c=512)
BCD	40 × 54 × 128	kernel=(3, 3) / strides=(2, 2) (a=512, b=512, c=128)
BCD	80 × 108 × 64	kernel=(3, 3) / strides=(2, 2) (a=256, b=256, c=64)
Up Sampling	240 × 324 × 64	size=(3, 3)
Cropping	240 × 320 × 64	cropping=[(0, 0) (2, 2)]
Convolución	240 × 320 × 64	kernel=(3, 3) / strides=(1, 1)
NB	-	
Activación	ReLU	
Convolución	240 × 320 × 1	kernel=(3, 3) / strides=(1, 1)
Activación	Sigmoidal	
Salida	240 × 320 × 1	-

convolucionales y otro por una única capa convolucional, que se suman para generar la salida del bloque. La diferencia entre los BCC (codificadores) y los BCD (decodificadores), es que las convoluciones son directas en los primeros y transpuestas en los segundos.

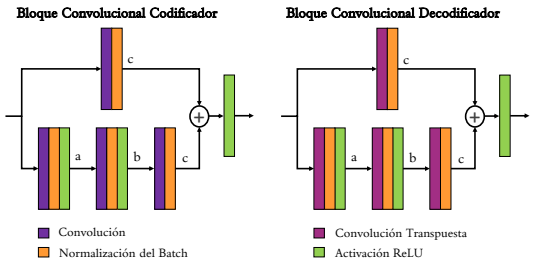

Figura 3. Arquitectura de los BCC y BCD

La Figura 3 muestra la estructura de un BCC y un BCD, donde los parámetros *a*, *b* y *c* corresponden a la tercera dimensión (profundidad) de la matriz en cada uno de los puntos, es decir al número de filtros de la capa anterior. Puede observarse como el número de filtros de la tercera convolución de la línea inferior debe coincidir con el número de filtros de la convolución de la línea superior (parámetro *c*). Los parámetro *a*, *b* y *c* de las Tablas I y II, y de la Figura 3 tienen el mismo significado.

III. ENTRENAMIENTO DEL SISTEMA

La base de datos utilizada para entrenar la CNN ha sido generada de forma sintética, lo cual ha permitido realizar el etiquetado de forma automática. Consta de 22000 imágenes de profundidad, que simulan haber sido tomadas con un sensor en posición frontal elevada, de las cuales 19800 se han utilizado para entrenar la CNN y las 2200 restantes para realizar la validación, que verifica el correcto progreso del entrenamiento, y el posterior test para obtener resultados. La escena simulada en las imágenes muestra una sala en la que caminan personas en diferentes direcciones. La perspectiva de la cámara no es fija, sino que realiza un giro de 360 grados a lo largo de toda la base de datos, lo cual permite evitar un fondo constante que sería aprendido por la CNN durante el entrenamiento. El hecho de emplear diferentes fondos alrededor de toda la sala sintética, permiten que la CNN asocie el fondo a ruido y solo se centre en las personas que aparezcan en la imagen, inmunizando la red y no atándola a unas condiciones de montaje y perspectiva exclusivas

La Figura 4 muestra diferentes perspectivas de la sala sintética en el entorno de simulación de Blender [14].

Las imágenes son de dimensiones 240×320 píxeles y codificadas en 16 bits sin signo. La Figura 5 muestra la apariencia de las imágenes de la base de datos en tres perspectivas diferentes.

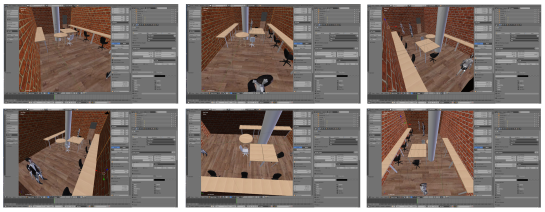

Figura 4. Diferentes perspectivas de la sala simulada en Blender

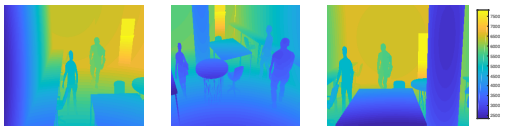

Figura 5. Imágenes de profundidad sintéticas para el entrenamiento

Respecto al etiquetado o anotado de las salidas, se han situado gaussianas sobre el punto medio de todas las cabezas, de forma que la media corresponde a la posición 2D del centro de la cabeza y tenga un valor normalizado de una unidad. La desviación típica es constante para todas las gaussianas, al margen del tamaño de cada cabeza y de la distancia de ésta a la cámara, y su valor se ha calculado a partir de una estimación del diámetro medio de las cabezas, que corresponde a 15 píxeles. La Ecuación 1 muestra la forma de calcular el valor de desviación típica utilizado.

$$\sigma = D/2{,}5 = 15/2{,}5 = 6 \tag{1}$$

Otro aspecto importante del etiquetado es el relativo a la superposición de gaussianas. Cuando dos cabezas se encuentran muy próximas o solapadas, las gaussianas anotadas sobre cada cabeza no se suman entre sí, sino que prevalece el valor mayor de ambas. Con ello se consigue que siempre exista separación aparente entre las gaussianas, de cara a que la CNN aprenda a generar de esta forma la salida, facilitando la posterior identificación de las personas detectadas.

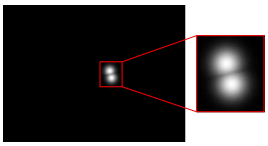

Figura 6. Detalle de Gaussianas etiquetadas

Respecto a los parámetros de la red relativos al proceso de entrenamiento, se ha utilizado como función de pérdidas el *Error Cuadrático Medio* entre el ground-truth y la salida de la red, tal y como indica la Ecuación 2, donde $\mathcal{L}$ representa la función de pérdidas, q_i la salida de la red y $\hat{q}_i$ el ground-truth.

$$\mathcal{L}(q_i, \hat{q}_i) = \frac{1}{N}\sum_{i=1}^{N}(\hat{q}_i - q_i)^2 \tag{2}$$

Se ha utilizado el optimizador *Adam* [15], con un *learning rate* inicial de 0.001, así como *early stopping* para evitar sobre-entrenamiento. Este mecanismo consiste en memorizar el valor de precisión obtenido al finalizar cada época. Si éste valor supera al valor memorizado en la época anterior significa que durante la última época el sistema ha mejorado y por tanto se guarda el nuevo valor de todos los pesos. Si por el contrario, el valor de precisión obtenido es inferior al último memorizado, significa que el sistema no ha mejorado durante la última época y los valores de los pesos no son actualizados.

IV. Resultados

Para la evaluación del sistema, en primer lugar se han obtenido resultados utilizando únicamente datos sintéticos, pertenecientes a la misma base de datos que las imágenes de entrenamiento, pero no utilizados durante el mismo. En concreto, se han empleado 2200 imágenes de 240×320 píxeles codificadas en 16 bits sin signo.

La segunda parte de la extracción de resultados se ha realizado con datos reales, pertenecientes a la base de datos utilizada en [16] y que puede encontrarse en [17]. De dicha base de datos, concretamente se han utilizado las imágenes cuyo escenario han llamado EPFL-LAB, que muestra una sala en la que las personas aparecen a una distancia y con una perspectiva similares a la base de datos de entrenamiento.

Respecto al algoritmo utilizado para extraer los resultados, ha sido el mismo tanto para el caso de datos sintéticos como para el caso de datos reales, únicamente variando el área de estudio dentro de la imagen, que debe ajustarse en función de las características de la base de datos. Para cada *frame*, se tiene un array de puntos indicados en el *ground-truth*, y otro array de puntos extraídos mediante una binarización (con un umbral de 0.6) del mapa de confianza generado a la salida de la red. Se realiza una asociación entre los puntos del ground-truth y los generados por la red, asociando entre sí aquellos que se encuentren a menor distancia, siempre que ésta sea menor a un límite impuesto en 37.5 píxeles, valor que corresponde a 2.5*D*, es decir, 2.5 veces el diámetro medio de cabeza estimado, utilizado en la Ecuación 1. Este límite no debe ser excesivamente restrictivo debido a que el ground-truth de las imágenes de la base de datos EPFL-LAB [16] se encuentra en formato *bounding box*, por lo que la estimación de la ubicación del centroide de cada cabeza no presenta gran exactitud. Aquellos puntos del *ground-truth* que queden sin asociar se consideran falsos negativos (FN), pues no se ha realizado la detección de esa persona. Por otro lado, aquellas detecciones de la red que queden sin asociar con puntos del *ground-truth* se consideran falsos positivos (FP), pues se ha realizado una detección incorrecta. Por último, se descartan aquellos fallos (FN o FP) correspondientes a puntos situados fuera del área de estudio, la cual se define en 3 dimensiones, siendo las dos primeras un rectángulo sobre el plano imagen, y la tercera un valor de distancia (profundidad) máxima. De ésta forma, para definir la ubicación del área de estudio se indicará una pareja de puntos (a y b), correspondientes a dos vértices opuestos del rectángulo sobre el plano imagen, y un valor de distancia máxima (d_{max}), de la siguiente forma: [(a_u, a_v), (b_u, b_v), d_{max}]. Los valores de FP y FN se entregan tanto en número (valor absoluto) como en porcentaje, respecto al total de puntos del *ground-truth* ubicados dentro del área de estudio.

A. Resultados con datos sintéticos

Para la extracción de resultados con datos sintéticos se ha utilizado como área de estudio la imagen completa, ya que al no existir ruido no es necesario descartar ni puntos del *ground-truth* ni detecciones del sistema que se encuentren en los márgenes de la imagen. Además, al tratarse de imágenes sintéticas no presentan el problema de distancia máxima válida y por tanto se han tenido en cuenta todos los puntos (tanto de *ground-truth* como detecciones), sea cual sea su distancia (valor de profundidad). La Tabla III, resume los resultados obtenidos.

Tabla III
Resultados con datos sintéticos

Área de Estudio	[(0, 0), (320, 240), inf]
Número de frames	2200
Puntos en el ground-truth	3176
Falsos Negativos	**212 (6.68 %)**
Falsos Positivos	**3 (0.09 %)**
Error (FN+FP)	**215 (6.77 %)**

Como se puede observar en la Tabla III, el sistema apenas genera detecciones incorrectas, sin embargo, hay un 6.68 % de personas en el *ground-truth* que no son detectadas. Analizando este hecho, se ha observado que esos fallos se producen en momentos en los que varias personas se encuentran muy próximas y aparecen oclusiones. Hay que tener en cuenta que la base de datos ha sido etiquetada automáticamente por el software de simulación, el cuál etiqueta todas las personas que aparecen en la escena, incluso cuando se producen oclusiones totales, caso en el que es imposible que la red sea capaz, por sí sola, de detectar a la persona.

B. Resultados con datos reales

Los base de datos reales utilizada, disponible en [17] y utilizada en otros trabajos como [16], está formada por 950 imágenes RGB-D (el sistema propuesto utiliza únicamente la información de profundidad) de 512×424 píxeles codificadas en 16 bits sin signo. Por tanto, ha sido necesario realizar su escalado a 320×240 para adaptarla a la capa de entrada de la red. Del mismo modo, aquellos puntos de las imágenes de valor nulo, que corresponden a puntos con una medición errónea de la distancia y que aparecen principalmente en el fondo de la escena, se han sustituido por el valor máximo de profundidad de la base de datos, pues la red no procesa píxeles de valor nulo como erróneos, sino como puntos a distancia nula. Las imágenes muestran una sala, similar a la simulada en los datos sintéticos, en la que aparecen hasta 4 personas. La ubicación de la cámara es frontal elevada, con una inclinación similar a la de los datos de entrenamiento. La Figura 7 muestra algunos ejemplos de imágenes pertenecientes a la base de datos EPFL-LAB.

Figura 7. Imágenes de la base de datos EPFL-LAB

El área de estudio utilizada se ha elegido descartando los bordes de la escena, en los que las personas no aparecen completas. La distancia máxima utilizada para la extracción de resultados se ha establecido en 3.5 metros debido a que a distancias mayores las medidas de distancia empeoran de forma significativa. La Tabla IV muestra los resultados obtenidos.

Tabla IV
RESULTADOS CON DATOS REALES

Área de Estudio	[(20, 55), (300, 205), 3500]
Número de frames	950
Puntos en el ground-truth	1959
Falsos Negativos	**474 (24.07 %)**
Falsos Positivos	**3 (0.15 %)**
Error (FN+FP)	**477 (24.35 %)**

Como se deduce de la Tabla IV, al igual que sucedía en el caso de datos sintéticos, el sistema no genera detecciones incorrectas, pues no hay un número significativo de Falsos Positivos. Sí son más frecuentes, por el contrario, los Falsos Negativos, que aparecen principalmente cuando existen oclusiones importantes.

En este trabajo se ha realizado también una comparativa con resultados obtenidos por otros sistemas del estado del arte que emplean la misma base de datos. Dichos sistemas de detección son DPOM, propuesto en [16] y que utiliza información de profundidad; ACF, detector de [18] y que utiliza información de color; PCL-MUNARO, propuesto en [19] y basado en información RGB-D; y por último KINECT2, con los resultados obtenidos por *Kinect for Windows SDK 2.0* [20] que utiliza información RGB-D. La Figura 8 muestra los resultados indicados en [16] para los algoritmos enumerados anteriormente. Se presentan también los resultados del sistema propuesto en este trabajo (identificado coo CNN) para dos valores de distancia máxima: 3.5 metros, valor con el que se han extraído los resultados presentados en la tabla anterior, y 4.5 metros, valor utilizado en [16]. Aquellos sistemas que expresan sus resultados mediante una curva (DPOM, ACF, PCL-MUNARO) disponen de un umbral en el algoritmo de detección que permite obtener diferentes valores para el punto *Precision-Recall*. Nuestro sistema, al igual que KINECT2, expresa sus resultados mediante un solo punto *Precision-Recall*, debido a que no se utiliza un umbral en el algoritmo de detección que pueda hacer variar el valor del punto *Precision-Recall*. Los valores numéricos obtenidos de los parámetros *Recall* y *Precision* se indican en la Tabla V.

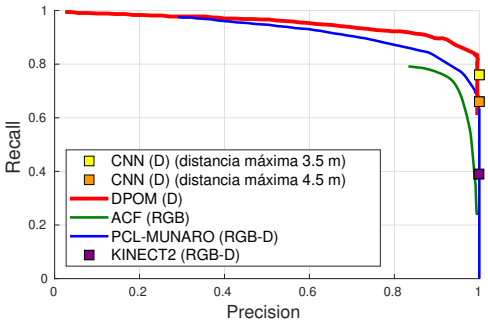

Figura 8. Comparativa entre diferentes sistemas

Tabla V
VALORES DE PRECISION Y RECALL DEL SISTEMA

Área de Estudio	[(20, 55), (300, 205), 3500]
Precision	**1.00**
Recall	**0.76**
Área de Estudio	[(20, 55), (300, 205), 4500]
Precision	**1.00**
Recall	**0.66**

Como puede observarse en la Figura 8 y en la Tabla V, el sistema propuesto basado en una CNN tiene una precisión del 100 %, pues no genera falsos positivos, siendo sus detecciones muy precisas. Los valores de *Recall* obtenidos de 0.66 y 0.76 indican que el sistema genera Falsos Negativos, y que su número aumenta si incrementamos la distancia máxima de evaluación. Es fácil deducir que estos Falsos Negativos se producen con personas parcialmente ocluidas, las cuales están situadas en posiciones más alejadas de la cámara. Sin embargo, el sistema funciona de forma similar o incluso mejor que algunos de los sistemas indicados en la Figura 8, especialmente si se tiene en cuenta que el entrenamiento se ha realizado únicamente con datos sintéticos.

V. Conclusiones

En este artículo se ha descrito un sistema para detección de personas a partir únicamente de información de profundidad, lo cual permite preservar la privacidad de las personas al no permitir reconocer su identidad. El sistema utiliza una Red Neuronal Convolucional (CNN) basada en los bloques residuales descritos en [11], entrenada únicamente con datos sintéticos, creados y etiquetados automáticamente por el simulador Blender [14], lo cuál permite entrenar con numerosos datos sin necesidad de adquirirlos ni etiquetarlos de forma manual. Para la evaluación del sistema se han utilizado tanto datos sintéticos como reales, [16], obteniendo en los segundos un 75.65 % de acierto, con una precisión del 100 %, es decir no genera falsas detecciones. Además, se han comparado los resultados con los de otras alternativas del estado del arte evaluadas sobre la misma base de datos, determinando que los resultados son similares, a pesar de que en el caso de este trabajo el entrenamiento se ha realizado utilizando únicamente imágenes de profundidad sintéticas. Con el fin de mejorar la robustez del sistema, la principal línea de trabajo futuro es el reentrenamiento con datos reales. Dicho entrenamiento debería realizarse con un número de datos reducido, pues se parte de una red pre-entrenada. De esta forma se consigue entrenar un sistema en dos etapas: en un primer lugar, con gran número de datos sintéticos que no necesitan etiquetado manual, y posteriormente con un número reducido de datos reales, evitando el costoso etiquetado manual de una base de datos de gran tamaño.

Agradecimientos

Este trabajo ha sido posible gracias a la financiación del Ministerio de Economía y Competitividad a través del proyecto HEIMDAL-UAH (TIN2016-75982-C2- 1-R), por la Universidad de Alcalá mediante los proyectos JANO (CCGP2017/EXP-025) y ACERCA (CCG2018/EXP-029) y las "Becas de Colaboración" del Ministerio de Educación.

Referencias

[1] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking People by Learning Their Appearance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 65–81, Nov. 2006.

[2] C. Y. Jeong, S. Choi, and S. W. Han, "A method for counting moving and stationary people by interest point classification," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Sept 2013, pp. 4545–4548.

[3] A. Bevilacqua, L. Di Stefano, and P. Azzari, "People tracking using a time-of-flight depth sensor," in *Video and Signal Based Surveillance, 2006. AVSS '06. IEEE International Conference on*, Nov 2006, pp. 89–89.

[4] X. Zhang, J. Yan, S. Feng, Z. Lei, D. Yi, and S. Li, "Water filling: Unsupervised people counting via vertical Kinect sensor," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, Sept 2012, pp. 215–220.

[5] C. Stahlschmidt, A. Gavriilidis, J. Velten, and A. Kummert, "People detection and tracking from a top-view position using a time-of-flight camera," in *Multimedia Communications, Services and Security*, ser. Communications in Computer and Information Science, A. Dziech and A. Czyazwski, Eds. Springer Berlin Heidelberg, 2013, vol. 368, pp. 213–223.

[6] C. A. Luna, C. Losada-Gutierrez, D. Fuentes-Jimenez, A. Fernandez-Rincon, M. Mazo, and J. Macias-Guarasa, "Robust people detection using depth information from an overhead time-of-flight camera," *Expert Systems with Applications*, vol. 71, pp. 240–256, 2017.

[7] R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *Quantum Electronics, IEEE Journal of*, vol. 37, no. 3, pp. 390–397, Mar 2001.

[8] J. Sell and P. O'Connor, "The Xbox one system on a chip and Kinect sensor," *Micro, IEEE*, vol. 34, no. 2, pp. 44–53, Mar 2014.

[9] C. Wang and Y. Zhao, "Multi-layer proposal network for people counting in crowded scene," in *Intelligent Computation Technology and Automation (ICICTA), 2017 10th International Conference on*. IEEE, 2017, pp. 148–151.

[10] J. Zhao, G. Zhang, L. Tian, and Y. Q. Chen, "Real-time human detection with depth camera via a physical radius-depth detector and a cnn descriptor," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 1536–1541.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, cite arxiv:1610.02357. [Online]. Available: http://arxiv.org/abs/1610.02357

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[14] Blender Online Community, "Blender a 3d modelling package," Blender Institute, Amsterdam. [Online]. Available: http://www.blender.org

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] T. Bagautdinov, F. Fleuret, and P. Fua, "Probability occupancy maps for occluded depth images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[17] T. Bagautdinov, "Rgb-d pedestrian dataset," https://cvlab.epfl.ch/data/data-rgbd-pedestrian/.

[18] P. Dollár, "Piotr's Computer Vision Matlab Toolbox (PMT)," https://github.com/pdollar/toolbox.

[19] M. Munaro and E. Menegatti, "Fast rgb-d people tracking for service robots," *Auton. Robots*, vol. 37, no. 3, pp. 227–242, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1007/s10514-014-9385-0

[20] Microsoft, "Kinect for windows sdk 2.0," Microsoft, 2014. [Online]. Available: http://www.microsoft.com/