Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza
1542

-ANEXOS-

Control de aforos mediante sistema multi-cámara

Francisco Morés Abad

Directora: Ana Cristina Murillo

Ingeniería Informática
Computación

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Fecha 21 de Junio de 2021

Índice general

Apéndice A

Anexo diagramas

Anexo donde se recogen los diagramas referentes al sistema. En la figura A.1 se puede ver un diagrama representando la jerarquía de módulos que existen dentro del sistema.

A.1. Diagrama de arquitectura

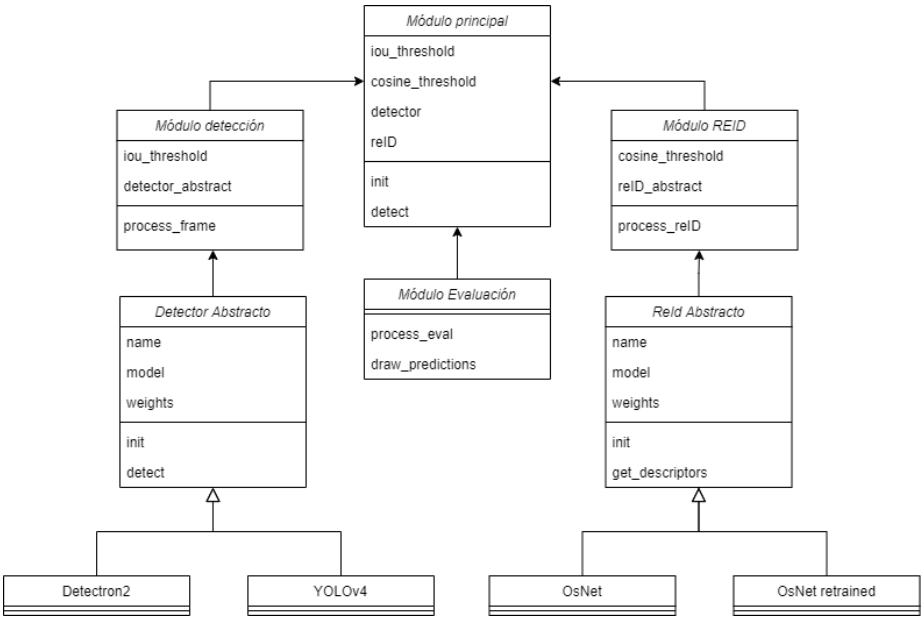


Figura A.1: Diagrama que muestra la organización, parámetros y métodos clave de los diferentes módulos: módulo de detección, módulo de re-identificación y módulo de evaluación y visualización.

A.2. Como utilizar el sistema

En esta sección se va a comentar el código y que parámetros se tienen para poder ajustar la ejecución del mismo con el fin de facilitar su uso en el futuro para cualquiera que lo deseé. En la tabla A.1 s encuentra un resumen de los diferentes parámetros del sistema y su utilidad.

Parámetro	***uso***
–path_videos	Directorio donde leer los diferentes vídeos. Dicho directorio debe contener un vídeo por cámara que se desee procesar.
-f, –frames	Directorio donde leer los diferentes frames. Dicho directorio debe contener una carpeta de imágenes por cámara que se desee procesar.
-a, –annotations	Directorio donde leer las anotaciones de los frames a procesar. Dicho directorio debe contener una carpeta de imágenes por cámara que se desee procesar.
-v, –verbose	Flag que permite que el sistema imprima un resumen de las operaciones que esta realizando. Solo útil cuando se poseen las anotaciones de los datos procesados.
-t, –show_time	Flag que permite mostrar por pantalla los tiempos requeridos para detección y reidentificación de cada frame procesado.
-npf, –num_processed_frames	Flag que indica el número de frames a procesar por el sistema.
-gt, –ground_truth	Opción para indicar al sistema que imprima solo las anotaciones.
-iou, –iou_threshold	Flag para modificar el umbral de IOU utilizado.
-cos, –cosine_threshold	Flag para modificar el umbral de similitud del coseno utilizado.
-p, –predictor	Opción para seleccionar el detector a utilizar. Nombres permitidos: YOLOv4 y detectron2.
-r, –reID	Opción para seleccionar el modelo de reidentificación a utilizar. Nombres permitidos: osnet.
-r_p, –reID_Path	Opción para indicar el directorio de donde cargar los pesos del modelo de reidentificación a utilizar.

Tabla A.1: Comandos del sistema y su utilidad.

Algunos ejemplos de invocación pueden ser:

- python v1_6.py –path_videos ../EINA_dataset/videos_sincronizados → Procesar los vídeos del dataset EINA.
- python v1_6.py -f ../EINA_dataset/frames_sincronizados -a no → Procesar

los frames del dataset EINA sin anotaciones.

- python v1_6.py -f ../Wildtrack -a si -v 1 -npf 40 -p detectron2 → Procesar los frames del dataset Wildtrack, con anotaciones, mostrando información adicional, mostrando solo 40 frames y utilizando el prodecitor detectron2 con máscaras.

Por ultimo comentar que todo el código desarrollado se puede encontrar en github [1] junto con algunos ficheros adicionales para poder instalar todos los paquetes necesarios para su uso.

[1] https://github.com/Fran-sw/Sistema-de-control-de-aforos

Apéndice B

Anexo experimentos

En este anexo se recogen resultados tanto de los experimentos realizados en *fine-tuning* como de extractos del dataset de la EINA. En la sección B.1 se puede ver una tabla con los diferentes entrenamientos realizados y en la sección B.2 diversos mosaicos con ejemplos del correcto o incorrecto funcionamiento del sistema.

B.1. Resultados *fine-tuning*

En esta sección se ha incluido la tabla con diversos entrenamientos del modelo OsNet probados. Es importante remarcar que están todos realizados con la misma configuración pensada para realizar *fine-tuning* (*learning rate* ligeramente más alto que en un entreno de cero y número de épocas más bajo debido al limite de tiempo). Como se ha comentado en la memoria principal se han utilizado las métricas de ***Rank-1*** y ***mPA***.

Los resultados empeoran generalmente por ciertos motivos, como la segmentación poco precisa en algunos casos y la falta de tiempo para realizar un experimento a mayor escala.

Un caso consigue superar los resultados de la versión de OsNet original, pero, ya que se realiza el entreno *fine-tune* con MSMT17 (dataset que el modelo original solo había visto en test), puede que esté generando sobre-ajuste.

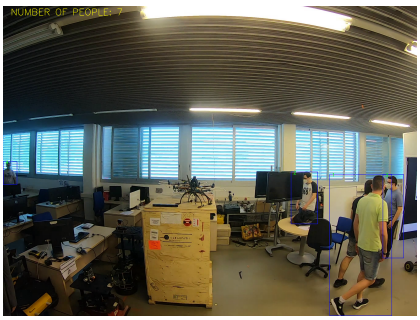
Original				Fine-tuning			
Train	Test	rank1(%)	mPA(%)	Train	Test	rank1(%)	mPA(%)
DukeMTMC, Market1501, CUHK03	MSMT17	40.2	16.2	Market1501	MSMT17	22.4	7.4
DukeMTMC, Market1501, CUHK03	MSMT17	40.2	16.2	MSMT17	MSMT17	55.0	22.7
DukeMTMC, Market1501, CUHK03	MSMT17	40.2	16.2	MSMT17	MSMT17	10.0	6.0
MSMT17	Market1501	70.1	43.3	MSMT17	MSMT17	42.6	15.5
MSMT17	Market1501	70.1	43.3	MSMT17	MSMT17	17.0	5.7
MSMT17	MSMT17	74.9	43.8	MSMT17	MSMT17	32.0	11.3

Tabla B.1: Resultados de las diferentes pruebas de *fine-tuning* realizadas. Todas tienen una configuración de 30 épocas, un *learning rate* de 0.0015 y *random sampling* de datos.

B.2. Ejemplos sobre el EINA dataset

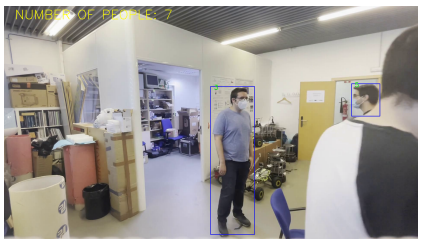
En esta sección se muestran ejemplos adicionales, instantes de tiempo de correcto e incorrecto funcionamiento, del dataset grabado en el laboratorio del edificio i3A en la EINA.

Figura B.1: Ejemplo del sistema completo funcionando en un instante de tiempo determinado en el EINA dataset. En este caso dentro del laboratorio solo hay una persona, pero ocurren dos problemas que provocan que el sistema indique 3 en aforo predicho. El primer problema se genera cuando el detector funciona demasiado bien y se detecta desde la cámara 2 a una persona justo en la entrada del laboratorio sin llegar a entrar. El segundo problema lo genera la cámara 5, donde se ve a la persona cortada y con un cambio de iluminación drástico que provoca que se re-identifique erróneamente.



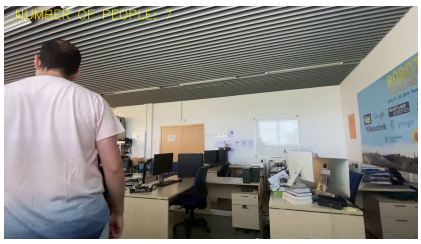
CONTEO REAL:7 - AUTOMÁTICO:7

cámara 1



cámara 2

cámara 3



cámara 4

cámara 5

Figura B.2: Ejemplo del sistema completo funcionando en un instante de tiempo determinado en el EINA dataset. En este caso, en el laboratorio se encuentran 7 personas y se estiman correctamente 7. No obstante, se pueden apreciar ciertos fallos en re-identificación: en la cámara 4 no se detecta correctamente a la persona de espaldas, en la cámara 2 se detecta una cabeza, pero no se re-identifica correctamente con una persona de la cámara 5 y en la cámara 2 no se detecta a la persona de espaldas. De forma general este instante representa de muy buena manera la gran cantidad de oclusiones que pueden suceder en el entorno monitorizado.

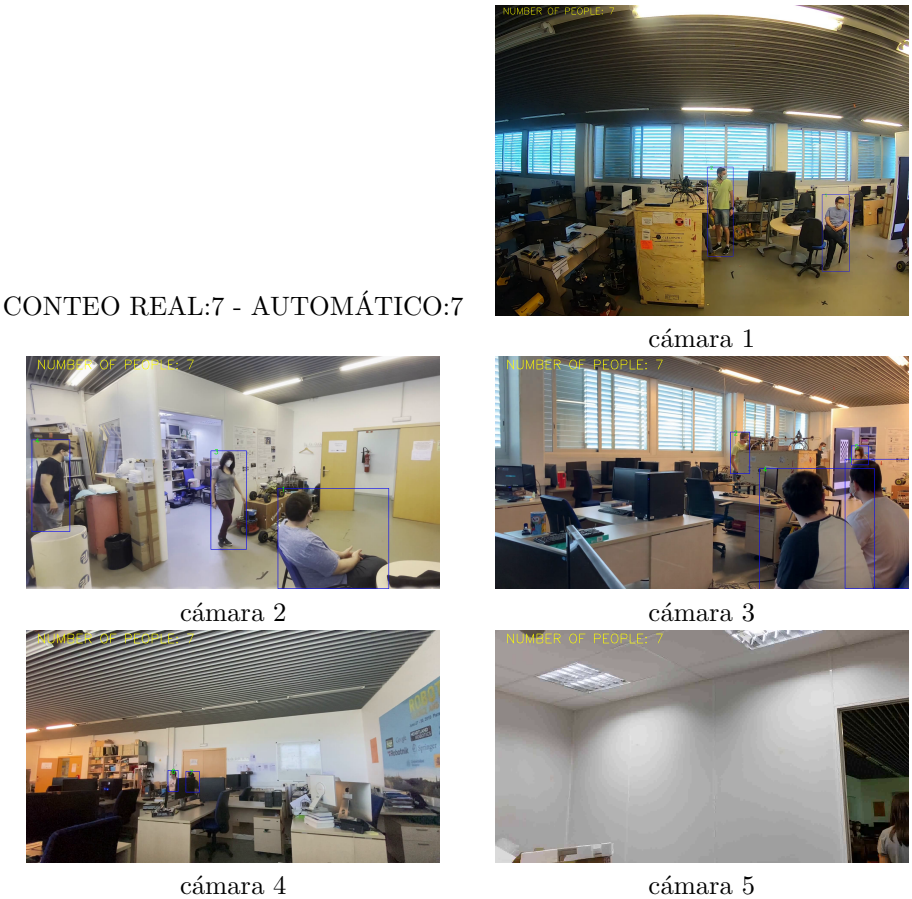


Figura B.3: Ejemplo del sistema completo funcionando en un instante de tiempo determinado en el EINA dataset. Este es otro instante de tiempo en el que se predice correctamente el aforo, pero se da un error de re-identificación. El error ocurre entre las dos personas sentadas, en la cámara 1, 2 y 3. La persona detectada en la tercera se confunde como la misma que en las dos primeras cámaras, ya que está de espaldas. No obstante, en la cámara 4 se detecta correctamente que es una persona distinta y no se le vuelve a dar el mismo identificador.

Bibliografía

[1] Le Cun Yan, B Yoshua, and H Geoffrey. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] Hong-Yuan Mark Liao Alexey Bochkovskiy, Chien-Yao Wang. Yolov4: Yolov4: Optimal speed and accuracy of object detection. *arXiv*, 2020.

[3] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[4] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5030–5039, 2018.

[5] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification. *TPAMI*, 2021.

[6] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3702–3712, 2019.

[7] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2109–2118, 2018.

[8] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2285–2294, 2018.

[9] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[10] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath

Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[13] Sumit Saha. A comprehensive guide to convolutional neural networks — the eli5 way.

[14] Prabhu. Understanding of convolutional neural network (cnn) — deep learning.

[15] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[16] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 79–88, 2018.

[19] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[20] Kaiyang Zhou and Tao Xiang. Torchreid: A library for deep learning person re-identification in pytorch. *arXiv preprint arXiv:1910.10093*, 2019.

[21] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.