

## Capítulo 6

# Conclusiones y trabajo futuro

### 6.1. Recapitulación

El resultado del proyecto es un prototipo de una aplicación para dispositivos con sistema operativo Android de reciclaje. Esta muestra aquello a lo que se apunta con la cámara del dispositivo e identifica y ofrece información sobre el material que compone el objeto. Como se ha comentado a lo largo del documento, para llevarla a cabo es necesario utilizar técnicas de visión artificial, las que actualmente conllevan la generación de un modelo entrenado por una red neuronal. Esto supone la necesidad de establecer varios subobjetivos.

El primer subobjetivo que se plantea es la obtención de un dataset amplio, claro y variado con imágenes de todos los materiales que se vayan a tener disponibles. Buscando facilitar la obtención de datasets, se ha desarrollado una aplicación de generación de imágenes sintéticas a partir de modelos tridimensionales. Esta está generada en la plataforma Unity, motor de videojuegos multiplataforma creado por Unity Technologies. El funcionamiento de esta aplicación consiste en ir cargando los distintos modelos 3D disponibles en los recursos, separados por material, y realizar numerosas capturas a cada uno hasta que se recorran todos. En cada captura tomada tanto la posición y rotación del objeto, como el fondo de la imagen, se establecen de manera aleatoria, proceso necesario para contar con una alta diversidad en las imágenes.

Una vez generado el dataset se da paso al segundo propósito para realizar la aplicación. Este se divide en dos pasos diferentes. El primero, y primordial, es el entrenamiento de la red neuronal y la generación del modelo entrenado utilizando las imágenes generadas previamente. El resultado del entrenamiento es incorporado posteriormente en Android Studio para su uso en la aplicación final. Para este apartado se ha utilizado TensorFlow Lite, que utiliza la red AlexNet y proporciona herramientas para llevar a cabo el entrenamiento de manera fácil e intuitiva.

Como segundo paso, una vez finalizado el entrenamiento de la red neuronal, se realizan diversas pruebas y comparaciones con la finalidad de encontrar la proporción entre imágenes sintéticas y reales que ofrecen el mejor equilibrio entre facilidad de obtención del dataset y precisión del modelo entrenado. La conclusión de las pruebas fue que a partir del 30 % de imágenes reales la precisión sufre muy pocos cambios y se mantiene siempre por encima del 90 % de confianza. Es decir, la precisión del modelo cuando todas las imágenes son reales y cuando sólo lo son un 30 % es muy similar. De esta forma, se tomó la decisión de utilizar el modelo con el 70 % de imágenes generadas y el 30 % restante reales que cuenta con un 93 % de confianza.

Finalmente, tiene lugar el desarrollo de la aplicación para dispositivos Android mediante la herramienta Andorid Studio utilizando TensorFlow Lite. La aplicación, a través de la cámara, recibe imágenes de los frames capturados y consultando el modelo importado trata de identificar el objeto que aparece en la imagen. Devuelve como resultado las etiquetas ordenadas según el porcentaje de seguridad con el que considera que se trata de ese objeto.

## 6.2. Conclusiones

En el proceso se han ido observando diversas dificultades y conclusiones importantes que han afectado al desarrollo del trabajo y que se deben tener en cuenta en posibles ampliaciones futuras o el desarrollo de otros proyectos similares o relacionados.

Se ha observado la dificultad real de obtener *datasets* amplios sobre objetos concretos si no se cuenta con muchos recursos, como se ha experimentado en el desarrollo del proyecto, y lo que ha provocado limitarse a contar con solamente tres materiales. Debido a esto se considera que la aplicación de generación de imágenes es un aporte útil para muchos proyectos que se salen del estándar en el ámbito del reconocimiento de objetos e imágenes. A pesar de esto, sigue existiendo cierta dificultad para generar el *dataset*. Esto es provocado por la obtención de modelos tridimensionales, ya que es importante que estos cuenten con un alto nivel de realismo en las texturas y los materiales, ya que de otra forma la manera en que la luz incide y se refleja sobre ellos puede resultar poco realista y generar problemas en el entrenamiento de la red neuronal. Esto provoca que el problema de la generación de *datasets* no quede solventado en su totalidad.

La iluminación es otro punto decisivo, a pesar de las facilidades que ofrece Unity para crear distintos tipos de iluminación, si no se tiene suficientes conocimientos y experiencia, el resultado puede terminar siendo pobre y, de nuevo, poco realista. Esto es un problema ya que la aplicación va a utilizarse sobre objetos reales y si no hay un equilibrio entre lo que se utiliza para el entrenamiento y el uso final, la precisión de la aplicación queda notablemente disminuida. Esto pudo observarse en los entrenamientos en los que el *dataset*

contaba con más de un 80 % de imágenes sintéticas, al ocupar estas la mayor parte del entrenamiento y diferir cómo se ve en la realidad, los resultados fueron peores en comparación con el resto de proporciones.

También en relación al *dataset*, después de investigar e informarse sobre distintos *datasets* sintéticos, y con la experiencia obtenida durante el proyecto, se puede afirmar que para utilizar un *dataset* con imágenes generadas es necesario que estas estén entremezcladas con reales, debido a que las cámaras virtuales y las reales son sensores diferentes. Además, esa mezcla permite a la red obtener una mejor precisión en la identificación.

Por último, una última observación a tener en cuenta, es que al llevar a cabo el entrenamiento respecto a tres materiales solamente, y donde las imágenes de sólo uno de ellos se mezclan con sintéticas, se observa que esto afecta no solamente al material que tiene las imágenes generadas, sino que también influye a los demás, generando errores en la identificación de objetos compuestos de materiales sobre los que se ha entrenado por completo con imágenes reales.

### 6.3. Trabajo futuro

En orden para ampliar y desarrollar una aplicación más completa, es necesario llevar a cabo varias mejoras en varios de los apartados del proyecto. La primera sería la obtención de un *dataset* más amplio, añadiendo más variedad de materiales y objetos identificables. Para esto, es necesario conseguir o generar modelos tridimensionales de todos los materiales y objetos que se quieran incorporar, los cuales deben tener una calidad bastante elevada. Otro factor ampliable, para permitir mayor diversidad en las imágenes generadas, es la ampliación del número de imágenes disponibles para el fondo. Con el objetivo de generar imágenes más realistas, sería necesario revisar la iluminación presente en la escena en la que tiene lugar las capturas, acercándolo a cómo se ven los objetos posteriormente en un entorno real.

Asimismo, también es necesario adquirir un pequeño *dataset* de imágenes reales de los materiales que se quieran agregar, para mezclarlo con las imágenes generadas obtenidas y tener el *dataset* final para el entrenamiento de la red neuronal. Una vez obtenido el *dataset*, simplemente habría que entrenar y generar el modelo para, finalmente, importarlo en la aplicación mediante Android Studio. Todos los materiales añadidos además deben relacionarse con el cubo o lugar de desecho apropiado, que es la información que busca el usuario.

Otra mejora adicional que es interesante explorar, es el convertir la generación de imágenes en una aplicación ejecutable independiente de Unity, en la que los usuarios puedan generar *datasets* para sus proyectos a partir de modelos propios suyos que importen en esta, o aprovechar algunos de los que se ofrezcan por defecto, por ejemplo los utilizados en este trabajo.

Respecto a la aplicación de Android, existen varias posibilidades de trabajo futuro, una de ellas es llevar a cabo una interfaz más personalizada que pueda ofrecer más información sobre los distintos cubos disponibles o el proceso de reciclaje. Otra, es desarrollar la misma aplicación para dispositivos iOS.

Una ampliación necesaria de la aplicación, es la adición de accesibilidad con el objetivo de permitir su uso a todo tipo de público. Dichas mejoras corresponden, entre otras, a la introducción de una opción de voz, personalización del tamaño de la fuente o de los colores y contrastes.

# Bibliografía

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattemberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] C. Alfonso, R. Estévez Estévez, J. M. Lobo, B. Lozano Diéguez, F. Prieto, J. Santamarta, and A. Gaerter. Emergencia climática en España. Diciembre 2016.
- [3] O. Alsing. Mobile object detection using tensorflow lite and transfer learning. Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2018.
- [4] Y. Amit. *2D Object Detection and Recognition: Models, Algorithms, and Networks*. Mit Press. MIT Press, 2002.
- [5] Y. Amit, P. Felzenszwalb, and R. Girshick. *Object Detection*. Springer International Publishing, Cham, 2020.
- [6] X. Basogain Olabe. Redes neuronales artificiales y sus aplicaciones. *Dpto. Ingeniería de Sistemas y Automática, Escuela Superior de Ingeniería Bilbao. Open Course Ware.*[En línea] disponible en [http://ocw.ehu.es/enseñanzas-tecnicas/redes-neuronales-artificiales-y-sus-aplicaciones/Course\\_listing](http://ocw.ehu.es/enseñanzas-tecnicas/redes-neuronales-artificiales-y-sus-aplicaciones/Course_listing). [Consultada 20-09-2012], 2008.
- [7] J. Cohen, C. F. Crispim-Junior, C. Grange-Faivre, and L. Tougne. CAD-based Learning for Egocentric Object Detection in Industrial Context. In *15th International Conference on Computer Vision Theory and Applications*, volume 5, Valletta, Malta, Feb. 2020. SCITEPRESS - Science and Technology Publications.

- [8] G. Cortina Fernández. Técnicas inteligentes para su integración en un vehículo autónoma. Trabajo de Fin de Grado en Ingeniería del Software, Facultad de Informática UCM, Departamento de Ingeniería del Software e Inteligencia Artificial, Curso 2019/2020., 2020.
- [9] B. Cyganek. *Object Detection and Recognition in Digital Images: Theory and Practice*. Wiley, 2013.
- [10] R. Flórez López, J. M. Fernández, and J. M. Fernández Fernández. *Las Redes Neuronales Artificiales*. Metodología y Análisis de Datos en Ciencias Sociales. Netbiblo, 2008.
- [11] R. Fonfría, R. Sans, and J. de Pablo Ribas. *Ingeniería ambiental: contaminación y tratamientos*. Colección productiva. Marcombo, 1989.
- [12] S. Frintrop. *VOCUS: A visual attention system for object detection and goal-directed search*, volume 3899. Springer, 2006.
- [13] G. A. Gómez Rojas, J. C. Henao López, and H. Salazar Isaza. Entrenamiento de una red neuronal artificial usando el algoritmo simulated annealing. *Scientia Et Technica*, 2004.
- [14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.
- [15] G. Guridi Mateos et al. Modelos de redes neuronales recurrentes en clasificación de patentes. B.S. thesis, 2017.
- [16] J. R. Hilera and V. J. Martínez Hernando. *Redes neuronales artificiales: fundamentos, modelos y aplicaciones*. 01 1995.
- [17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.
- [18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size, 2016.
- [19] Y.-C. Jhang, A. Palmar, B. Li, S. Dhakad, S. K. Vishwakarma, J. Hoggins, A. Crespi, C. Kerr, S. Chockalingam, C. Romero, A. Thaman, and S. Ganguly. Training a performant object detection ML model on synthetic data using Unity Perception tools, Sep 2020.
- [20] R. Karim. *TensorFlow: Powerful Predictive Analytics with TensorFlow*. Packt Publishing, Limited, 2018.

- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [22] P. Larranaga, I. Inza, and A. Moujahid. Tema 8. redes neuronales. *Redes Neuronales, U. del P. Vasco*, 12, 1997.
- [23] M. A. López Pacheco. Identificación de sistemas no lineales con redes neuronales convolucionales. *Cuidad de Mexico: Centro de investigación y de estudios avanzados*, 2017.
- [24] D. J. Matich. Redes neuronales: Conceptos básicos y aplicaciones. *Universidad Tecnológica Nacional, México*, 41, 2001.
- [25] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 1943.
- [26] B. Müller, J. Reinhardt, and M. Strickland. *Neural Networks: An Introduction*. Physics of Neural Networks. Springer Berlin Heidelberg, 1995.
- [27] C. Parra Ramos and D. Regajo Rodríguez. Reconocimiento automático de matrículas. *Universidad Carlos III de Madrid*, 2006.
- [28] R. Pavón Benítez. Técnicas de deep learning para el reconocimiento de movimientos corporales. Trabajo de Fin de Grado en Ingeniería del Software, Facultad de Informática UCM, Departamento de Ingeniería de Software e Inteligencia Artificial, Curso 2019/2020, 2020.
- [29] A. Polacco and K. Backes. The amazon go concept: Implications, applications, and sustainability. *Journal of Business and Management*, 24(1), 2018.
- [30] A. Polacco and K. Backes. The amazon go concept: Implications, applications, and sustainability. *Journal of Business and Management*, 24(1), 2018.
- [31] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [32] A. Rozantsev, V. Lepetit, and P. Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137, 11 2014.
- [33] M. Sánchez and J. Castro. *Gestión y Minimización de Residuos*. Fundación Confemetal, 2007.

- [34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.
- [37] Unity Technologies. Unity Perception package, 2020.
- [38] S.-C. Wang. *Artificial Neural Network*. Springer US, Boston, MA, 2003.
- [39] P. Warden and D. Situnayake. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, 2019.
- [40] W. Yu and Y. Bai. Visualizing and comparing alexnet and vgg using deconvolutional layers. 2016.
- [41] J. Zurada. *Introduction to Artificial Neural Systems*. West, 1992.