

Capítulo 6

Conclusiones y trabajo futuro

6.1. Recapitulación

El resultado del proyecto es un prototipo de una aplicación de ayuda al reciclaje para dispositivos con sistema operativo Android. Muestra aquello a lo que se apunta con la cámara del dispositivo e identifica y ofrece información sobre el material que compone el objeto además de indicar la manera adecuada de desecharlo. 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 planteó 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 recorren 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. Debido a los inconvenientes surgidos durante la obtención de los modelos, finalmente se obtuvo un *dataset* compuesto por tres materiales diferentes (metal, vidrio y plástico), donde para metal se mezclan imágenes reales con sintéticas.

Una vez generado el *dataset* se dio 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 una librería de TensorFlow Lite que permite el entrenamiento por transferencia utilizando la red EfficientNet 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 exactitud 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 se utiliza en uno de los materiales sólo un 30 % de estas, 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 para el metal, que cuenta con un 93 % de confianza.

Finalmente, tiene lugar el desarrollo de la aplicación para dispositivos Android mediante la herramienta Andorid Studio y 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 disponibles ordenadas según el porcentaje de seguridad con el que considera que se trata de ese objeto, informando al usuario del material, el contenedor donde desecharlo y la confianza con la que considera que se trata de dicho material.

Para probar su funcionamiento se hicieron ensayos sobre objetos del mundo real. Para ello, se realizaron cambios en la aplicación dando lugar a otra de *test*. A través de ella se compara el funcionamiento de varios modelos simultáneamente. Con los datos obtenidos, se corroboró el correcto funcionamiento del modelo entrenado con el *dataset* seleccionado.

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. Una solución a este problema puede ser el uso de imágenes sintéticas para completarlos. Este trabajo ha demostrado que, al menos en el contexto planteado, esta opción tiene resultados altamente positivos, ya que poblando una parte significativa del *dataset* con imágenes sintéticas, la precisión apenas se ve afectada. Tras la realización de pruebas

en el entorno real, se corrobora el correcto funcionamiento de la aplicación obteniendo resultados acertados en la identificación y con confianza considerablemente positiva. Por este motivo, 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.

No obstante, para que esto se cumpla, conseguir imágenes sintéticas tiene que resultar más sencillo que conseguir fotografías, ya que sigue existiendo cierta dificultad para generar el *dataset*. Esto es provocado por la obtención de modelos tridimensionales, los cuales es importante que cuenten con un alto nivel de realismo en las texturas y materiales. Esto se debe a que 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, haciendo 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.

En vista de los resultados obtenidos durante el trabajo, y con el apoyo de otros proyectos donde se ha trabajado con *datasets* sintéticos, se puede afirmar que si se va a utilizar un *dataset* compuesto por imágenes sintéticas es necesario que estas se encuentren entremezcladas con reales para un correcto funcionamiento. Esto se debe a que las cámaras virtuales y las reales son sensores diferentes.

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

El proyecto cuenta con diversos puntos ampliables. El primero es la obtención de un *dataset* más amplio, añadiendo más variedad de materiales y objetos identificables. Esto puede hacerse a partir de imágenes reales únicamente, o bien mezclándolas con imágenes sintéticas. Para esta segunda

opción, 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 extensión 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, es necesario adquirir un *dataset* de imágenes reales de los materiales que se quieran agregar. La cantidad de imágenes de este estaría regido por si se va combinarse con imágenes sintéticas o no.

Una vez obtenido el *dataset*, simplemente es necesario entrenar y generar el modelo para, finalmente, importarlo en la aplicación mediante Android Studio. Todos los materiales añadidos además deben relacionarse con el contenedor o lugar de desecho apropiado, que es la información que busca el usuario.

Para mejorar la red y ampliar el *dataset*, podría desarrollarse la opción de que los usuarios puedan enviar sus propias fotografías etiquetadas con el material del que se trate, dando lugar a un *dataset* más completo que se encuentre en continuo crecimiento ofreciendo así un mejor servicio a los usuarios.

Otra mejora adicional interesante, es 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. Aunque la importación de modelos en tiempo de ejecución es algo poco extendido en el mundo de los videojuegos, debido a la cantidad y diversidad de usuarios con los que cuenta Unity, esta funcionalidad ha sido ampliamente discutida y explorada. Pueden incluso encontrarse paquetes para llevar esto a cabo, por ejemplo “TriLib 2 - Model Loading Package” de Ricardo Reis¹, “Runtime OBJ Importer” de Dummiesman² u “OBJReader” de Starscene Software³, así como numerosas discusiones en foros sobre el tema.

Respecto a la aplicación de Android también 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, permitiendo así su acceso a un mayor número de usuarios. Además, una ampliación necesaria de la aplicación, es la incorporación de

¹<https://assetstore.unity.com/packages/tools/modeling/trilib-2-model-loading-package-157548>

²<https://assetstore.unity.com/packages/tools/modeling/runtime-obj-importer-49547>

³<https://starscenesoftware.com/objreader.html#ObjReader>

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] R. Almond, G. M., and T. Petersen. Wwf (2020) living planet report 2020 - bending the curve of biodiversity loss. *World Wildlife Fund (WWF)*, 2020.
- [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. [Consultada 20-09-2012]*, 2008.
- [7] M. Caballero, S. Lozano, and B. Ortega. Efecto invernadero, calentamiento global y cambio climático: una perspectiva desde las ciencias de la tierra. *Revista digital universitaria*, 8, 2007.
- [8] 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.
- [9] 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.
 - [10] B. Cyganek. *Object Detection and Recognition in Digital Images: Theory and Practice*. Wiley, 2013.
 - [11] 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.
 - [12] R. Fonfría, R. Sans, and J. de Pablo Ribas. *Ingeniería ambiental: contaminación y tratamientos*. Colección productiva. Marcombo, 1989.
 - [13] S. Frintrop. *VOCUS: A visual attention system for object detection and goal-directed search*, volume 3899. Springer, 2006.
 - [14] L. García Rodríguez. *Algunas cuestiones notables sobre el modelo de Hopfield en optimización*. PhD thesis, Madrid, Noviembre 2018. Tesis de la Universidad Complutense de Madrid, Facultad de Ciencias Matemáticas, Departamento de Estadística e Investigación Operativa, leída el 15-12-2017.
 - [15] 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.
 - [16] G. Guridi Mateos et al. Modelos de redes neuronales recurrentes en clasificación de patentes. B.S. thesis, 2017.
 - [17] J. R. Hilera and V. J. Martínez Hernando. *Redes neuronales artificiales: fundamentos, modelos y aplicaciones*. 01 1995.
 - [18] S. Hinterstoisser, S. Benhimane, V. Lepetit, P. Fua, and N. Navab. Simultaneous recognition and homography extraction of local patches with a simple linear classifier. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2008. doi:10.5244/C.22.10.
 - [19] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

- [20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. We-
yand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional
neural networks for mobile vision applications, 2017.
- [21] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and
K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer param-
eters and <0.5mb model size, 2016.
- [22] A. Iguarán Guerra, S. Gómez Ruíz, et al. Análisis de las necesidades y
dificultades en la disposición de residuos sólidos en la fuente doméstica
para el desarrollo de un producto. B.S. thesis, Universidad EAFIT,
2010.
- [23] Y.-C. Jhang, A. Palmar, B. Li, S. Dhakad, S. K. Vishwakarma, J. Ho-
gins, 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.
- [24] R. Karim. *TensorFlow: Powerful Predictive Analytics with TensorFlow*.
Packt Publishing, Limited, 2018.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification
with deep convolutional neural networks. *Advances in neural informa-
tion processing systems*, 25, 2012.
- [26] P. Larranaga, I. Inza, and A. Moujahid. Tema 8. redes neuronales. *Redes
Neuronales, U. del P. Vasco*, 12, 1997.
- [27] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays,
P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco:
Common objects in context, 2015.
- [28] P. López and J. García-Consuegra Bleda. *Informática gráfica*, volu-
me 19. Univ de Castilla La Mancha, 1999.
- [29] M. A. López Pacheco. Identificación de sistemas no lineales con redes
neuronales convolucionales. *Cuidad de Mexico: Centro de investigación
y de estudios avanzados*, 2017.
- [30] D. J. Matich. Redes neuronales: Conceptos básicos y aplicaciones. *Uni-
versidad Tecnológica Nacional, México*, 41, 2001.
- [31] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent
in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 1943.
- [32] A. J. McMichael, D. Campbell-Lendrum, S. Kovats, S. Edwards, P. Wil-
kinson, T. Wilson, R. Nicholls, S. Hales, F. Tanser, D. L. Sueur,
M. Schlesinger, and N. Andronova. Chapter 20 global climate chan-
ge.

- [33] M. Minsky and S. A. Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.
- [34] S. Morant Gálvez. Desarrollo de un sistema de bajo coste para el análisis de tráfico mediante el uso de deep learning. 2021.
- [35] L. Moreno Díaz-Alejo. Análisis comparativo de arquitecturas de redes neuronales para la clasificación de imágenes. Master's thesis, 2020.
- [36] B. Müller, J. Reinhardt, and M. Strickland. *Neural Networks: An Introduction*. Physics of Neural Networks. Springer Berlin Heidelberg, 1995.
- [37] C. Parra Ramos and D. Regajo Rodríguez. Reconocimiento automático de matrículas. *Universidad Carlos III de Madrid*, 2006.
- [38] 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.
- [39] A. Polacco and K. Backes. The amazon go concept: Implications, applications, and sustainability. *Journal of Business and Management*, 24(1), 2018.
- [40] A. Polacco and K. Backes. The amazon go concept: Implications, applications, and sustainability. *Journal of Business and Management*, 24(1), 2018.
- [41] 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.
- [42] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [43] A. Rozantsev, V. Lepetit, and P. Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137, 11 2014.
- [44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088), 1986.
- [45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

- [46] R. Salas. Redes neuronales artificiales. *Universidad de Valparaíso. Departamento de Computación*, 1, 2004.
- [47] M. Sánchez and J. Castro. *Gestión y Minimización de Residuos*. Fundación Confemetal, 2007.
- [48] 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.
- [49] O. Simeone. A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*, 4(4), 2018.
- [50] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [51] 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*, June 2015.
- [52] A. Terceño Ortega. Análisis de un modelo predictivo basado en google cloud y tensorflow. Trabajo de Fin de Grado en Ingeniería Informática y Matemáticas (Universidad Complutense, Facultad de Informática, curso 2016/2017), 2017.
- [53] Unity Technologies. Unity Perception package, 2020.
- [54] S.-C. Wang. *Artificial Neural Network*. Springer US, Boston, MA, 2003.
- [55] P. Warden and D. Situnayake. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, 2019.
- [56] W. Yu and Y. Bai. Visualizing and comparing alexnet and vgg using deconvolutional layers. 2016.
- [57] J. Zamorano Ruiz et al. Comparación y análisis de métodos de clasificación con las bibliotecas scikit-learn y tensorflow en python. 2019.
- [58] J. Zurada. *Introduction to Artificial Neural Systems*. West, 1992.