I Workshop en Deep Learning (DeepL)

SESIÓN 1

Optimización de las técnicas de Transfer Learning para la clasificación de la calidad estética en fotografía

Fernando Rubio
SIMD Lab, I3A
UCLM
Albacete, España
Email: fernando.rubio@uclm.es

M. Julia Flores
Departamento de Sistemas Informáticos
UCLM
España
Email: julia.flores@uclm.es

Jose M. Puerta
Departamento de Sistemas Informáticos
UCLM
España
Email: jose.puerta@uclm.es

***Abstract*—La evaluación automática de la calidad estética es un problema de visión por ordenador que consiste en cuantificar el atractivo de una fotografía. Esto es especialmente útil en las redes sociales, donde la cantidad de imágenes que se generan cada día requieren de la automatización para su procesamiento.**

Aunque ha habido progresos notables en la investigación de este campo, aún es difícil encontrar soluciones aplicables. Con este trabajo buscamos la optimización de las soluciones más prometedoras basadas en Transfer Learning. Para ello, hemos reducido la complejidad de las redes neuronales propuestas manteniendo los resultados obtenidos, mediante técnicas de "finetuning" sobre redes pre-entrenadas.

***Index Terms*—Deep Learning, Transfer Learning, Finetuning, Classification, Computer Vision**

I. Introducción

El campo de visión por ordenador es uno de los más activos en la comunidad científica debido a la gran cantidad de aplicaciones que tiene como la robótica y la seguridad. En los últimos años, las redes neuronales profundas han permitido resolver problemas, que hasta hace poco parecían inabordables. Esto ocurre también con el problema de la calidad estética.

El concepto de calidad estética en la fotografía hace referencia a las propiedades de las imágenes que las hacen atractivas o "bonitas" para la mayoría de la gente, como pueden ser los filtros aplicados, armonía de los colores, etc. No hay que confundir con la calidad de una imagen en términos de resolución. Se trata de uno de los problemas más complejos dentro del campo de visión debido a la subjetividad de la tarea, ya que la opinión de diferentes personas sobre la calidad de una única imagen puede ser muy distinta. Incluso entre expertos de fotografía puede haber opiniones diferentes.

A pesar de su dificultad, se trata de un problema que ha visto incrementado su interés enormemente debido a la gran cantidad de imágenes que se generan continuamente con las redes sociales. La automatización de esta tarea tiene aplicaciones muy interesantes como la ordenación de álbumes de imágenes en base a su calidad, especialmente útil para sitios como Flickr o Instagram. Pero también se puede utilizar para recomendaciones de filtros o incluso para evaluaciones online en una cámara, para mejorar la calidad de las fotografías que tomamos. Otro de los campos de aplicación es la publicidad, con la creación de imágenes más atractivas.

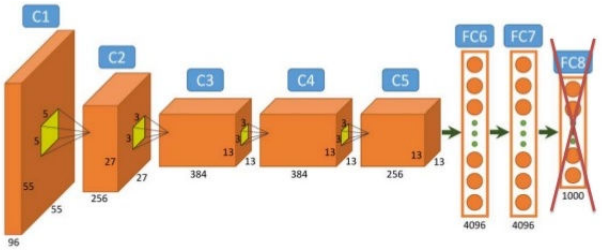


Fig. 1. Extracción de características de una red neuronal.

Tradicionalmente, la automatización de la evaluación de la calidad estética se centra en resolver un problema de clasificación binaria, donde las imágenes son clasificadas como "snapshots" (mala calidad) o "professional shots" (buena calidad). El uso del Deep Learning mediante las redes convolucionales ha mejorado los resultados en los últimos años e incluso ha permitido utilizar las probabilidades generadas en la última capa como indicadores más precisos de la calidad de la imagen.

En este artículo nos centramos en una de las estrategias más utilizadas recientemente en Deep Learning, Transfer Learning. Este concepto se basa en utilizar redes neuronales pre-entrenadas con otros datasets y aplicar dicho conocimiento a nuestro problema.

Actualmente, dos son las técnicas principales de Transfer Learning. La primera de ellas, puede verse en la Fig. 1 y que consiste en la extracción de características de una red neuronal, lo que se conoce como **ConvNet features** o **DeCaf** [1]. En este caso, obviamos la salida de la red y obtenemos las activaciones que se producen en las capas anteriores, para utilizarlas como *inputs* en otros modelos.

La segunda es el concepto de **finetuning**, que puede verse en la Fig. 2. Esta técnica de Transfer Learning consiste en la modificación de la (o las) última(s) capa(s) para ajustar

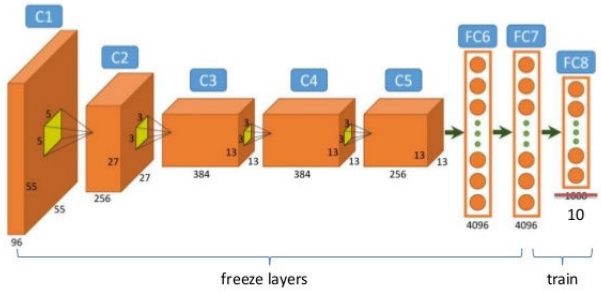


Fig. 2. Proceso de *finetuning*, donde se ha modificado la última capa de una red pre-entrenada, cuyos pesos van a ser aprendidos. El resto de capas no sufre modificaciones.

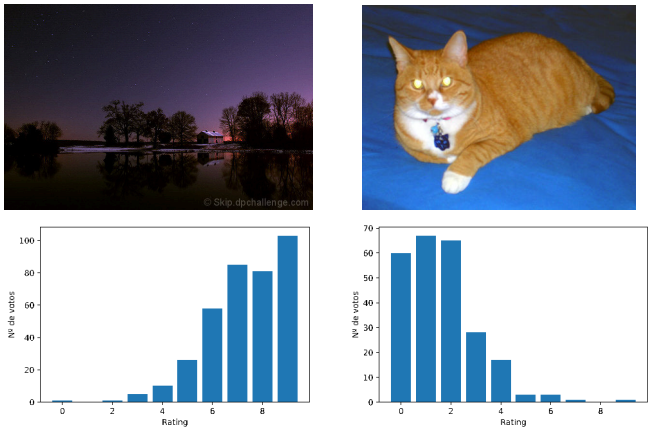


Fig. 3. Ejemplo de una imagen profesional (arriba, izquierda) y de una de mala calidad (arriba, derecha). En la parte inferior se muestra la distribución de los votos para cada imagen.

la salida de una red a nuestro problema. El siguiente paso consiste en el entrenamiento de los pesos sólo de las capas que han sido modificadas. Esto reduce el tiempo y la cantidad de datos necesarios para entrenar la red.

Cuando hablamos de Transfer Learning en imágenes, existen diferentes modelos pre-entrenados que podemos utilizar como son AlexNet [2], VGG [3], Inception [4], ResNet [5] o MobileNet [6]. Todos estos comparten una serie de propiedades y la base de datos con la que fueron entrenadas, ImageNet [7]. Este dataset consiste en un conjunto de más de 2 millones de imágenes de objetos (valla, barco, avión), animales (perro, gato, tortuga) o conceptos (atardecer, paisaje) que han sido etiquetadas con 1 de 1000 posibles "tags". Por lo tanto, ImageNet es un problema de clasificación donde la clase puede tomar 1000 posibles valores. El objetivo de utilizar Transfer Learning sobre las redes aprendidas para este problema, es aprovechar todo el conocimiento generado por una red capaz de identificar 1000 conceptos diferentes en imágenes.

Los mejores resultados obtenidos para la clasificación de la calidad estética vienen de propuestas de Transfer Learning sobre dichos modelos, especialmente con *finetuning* [8]. Sin embargo, en el estudio de estas soluciones, por lo general, sólo la última capa es modificada para el reentrenamiento. En este trabajo proponemos realizar un proceso de *finetuning* de más capas, con el objetivo de reducir el tamaño de la red, pero sin afectar a los resultados obtenidos. Esto permitirá que las redes neuronales puedan utilizarse en dispositivos más limitados en recursos computacionales como son los dispositivos móviles o las cámaras fotográficas, permitiendo la creación de aplicaciones reales.

II. Estado del arte

Las primeras aproximaciones para la evaluación de la calidad estética consisten en una clasificación binaria de las imágenes en "snapshots" o "professional shots". Sin embargo, al tratarse de un problema con tanta incertidumbre y donde la clase no está bien definida, las principales bases de datos están compuestas por imágenes donde un grupo de individuos han asignado unos ratings o votos [9]. En la Fig. 3, podemos observar dos imágenes con su distribución de votos.

Generalmente, a partir de los ratings asignados a una imagen, obtenemos estadísticos que nos permiten convertir el problema en uno de clasificación binaria. Por ejemplo, podemos utilizar métricas como la media o la mediana para separar las imágenes en "snapshots" o "professional shots". Normalmente, se fija un umbral en el punto medio del rango de los ratings, es decir, si los votos van en una escala del 1 al 10, el corte se establece en 5. Finalmente, se obtiene la media para cada imagen μ_i a partir de sus votos y se comparan con el umbral. En la Fig 3, en el caso de la izquierda, tenemos una $\mu = 8.31$, y en la derecha $\mu = 2.62$. Al tener un rango de 10 posibles valores, el umbral se situa en el 5, por lo que la imagen de la izquierda sería clasificada como "professional shot" y la de la derecha como "snapshot".

[10] y [11] proponen resolver el problema con características hechas a mano de bajo nivel para tratar de identificar propiedades más complejas de las fotografías y que sean capaces de separar ambas categorías. Sin embargo, estas propuestas pronto fueron superadas por técnicas generales de extracción de características, como GIST o SIFT [12].

Con la aparición del dataset AVA [13], el problema de la calidad estética ya contaba con una considerable base de datos de imágenes que permitía el aprendizaje de redes de Deep Learning como en [14]. Sin embargo, los resultados de entrenar una red neuronal desde cero, para este problema concreto, no han sido tan relevantes como en otros campos de visión.

[8] demuestra que realizando el proceso de *finetuning* en la última capa de los modelos AlexNet y VGG es posible obtener unos resultados más fiables que los presentados hasta ese momento. [15] también hace uso de esta técnica de Transfer Learning para predecir directamente la distribución de los votos, donde la capa de *output* tiene un tamaño de 10, correspondiente al rango de ratings de AVA, en vez de 2 de la clasificación binaria. En este último caso, en vez de *softmax* como función de activación de la última capa, utilizan Earth Mover's Distance (EMD), que obtiene la distribución de probabilidad acumulada para la imagen.

Hay que destacar que la mayoría de los trabajos sufrían de un problema con la evaluación, como se indica en [16], ya que la única métrica utilizada en muchas propuestas para validar los modelos era la tasa de aciertos o *accuracy*. En AVA, esta métrica es poco informativa, ya que si binarizamos la clase a partir de los votos, cogiendo como umbral el 5 de media (ya que el rango va de 1 a 10), observamos un desbalanceo de la clase, donde el 70% de los casos son "professional shots" y el 30% "snapshots". En este caso, reportar una tasa de aciertos del 70% no tiene valor, ya que se pueden estar clasificando todas las imágenes como buenas.

Para resolver esta situación, [16], [8] y [15] hacen uso de diferentes métricas como son el *balanced accuracy* que tiene en cuenta la tasa de aciertos por cada una de las clases o el valor AUC (Area Under the Curve), que relaciona la tasa de Verdaderos Positivos con la tasa de Falsos Positivos. En este trabajo utilizaremos esas métricas cuando trabajemos con problemas desbalanceados.

Es frecuente encontrar en la literatura propuestas que tienden a reducir las bases de datos originales en subconjuntos. Principalmente se eliminan imágenes cuya media de votos se encuentra cerca del umbral de corte. [14] y [12] utilizan una δ, de forma que si consideramos 5 el punto de corte, las imágenes $< 5 - \delta$ son consideradas "snapshots" y las $> 5 + \delta$ son etiquetadas como "professional shots", descartando el resto. Otros, como [17] o [18] seleccionan el 10% mejor y peor valoradas y el resto no se tienen en cuenta.

Es comprensible tratar de dar más peso a aquellas imágenes más informativas y tratar de reducir el ruido que pueden generar los casos cercanos a la frontera de decisión. Sin embargo, en algunas de estas propuestas, los resultados presentados han eliminado del conjunto de evaluación las imágenes, con el fin de simplificar el problema. Consideramos que estos resultados no son válidos para un escenario real, donde la mayoría de las imágenes se encuentran cerca del umbral de corte. La eliminación de imágenes sólo debe realizarse en los conjuntos de entrenamiento.

Por último, en las Fig.4 y 5 se muestran resultados preliminares de la evaluación de la calidad estética en AVA utilizando características generales de la imagen. En ambas se refleja la evolución de los resultados en base al valor δ donde sólo se descartan las imágenes del conjunto de entrenamiento. Se observa que eliminar las imágenes cercanas al punto de corte no afecta de forma significativa a los resultados, he incluso en el caso del AUC vemos un peor comportamiento. Esto corrobora lo resultados de [14], donde la reducción de las imágenes también perjudica a los modelos de Deep Learning. Por estos motivos, en este trabajo utilizaremos las bases de datos completas.

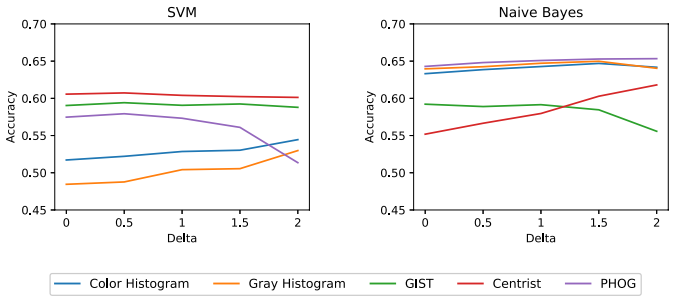


Fig. 4. Evolución del *accuracy* (tasa de aciertos) en base a la δ (Delta) para la reducción de imágenes de la base de datos AVA.

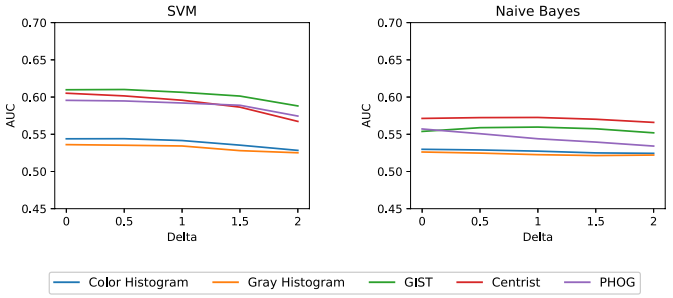


Fig. 5. Evolución del AUC en base a la δ (Delta) para la reducción de imágenes de la base de datos AVA.

III. Transfer Learning

A. Dataset

Actualmente, la base de datos referente para el tratamiento de la calidad estética es AVA. Este dataset esta compuesto por cerca de 250.000 imágenes pertenecientes a una página de retos fotográficos, DPChallenge. Cada foto ha sido valorada del 1 al 10 por diferentes usuarios, siendo el 1 que la foto es muy mala y 10 que es muy buena. Además, cada imagen viene etiquetada con el tipo de reto en la que se subió, el estilo fotográfico (si tiene alguno) y ciertas etiquetas sobre objetos que aparecen en la imagen. En los trabajos previos, esta base de datos es particionada en train y test, siendo este último de unas 20k imágenes y las 230k restantes para entrenamiento. En este trabajo utilizaremos la misma partición.

B. Extracción de ConvNet features

Durante la evaluación de una imagen en una red neuronal (proceso *forward*), no sólo se obtienen los valores de salida, en cada capa de la red se generan una serie de activaciones que pueden ser extraídas. Aunque es difícil interpretar esta información, puede utilizarse como vectores de características de la imagen, ya que tienen una gran capacidad descriptiva. Esto es lo que se conoce como *ConvNet features* y se utilizan generalmente en problemas donde no tenemos suficientes datos como para entrenar una red neuronal. Estas características son extraídas de redes pre-entrenadas y se utilizan para aprender otros modelos de clasificación.

En [16] se realiza un estudio del rendimiento de las *ConvNet features* extraídas de dos redes neuronales (AlexNet y ResNet) en la evaluación de la calidad estética. En las Fig. 6 y 7 se observan los resultados, donde los modelos entrenados con las *ConvNet features* superaban a los clasificadores que utilizaban descriptores generales de la imagen. Sin embargo, debido a la gran cantidad de características que se obtienen de las redes neuronales, no todos los modelos se muestran en este estudio.

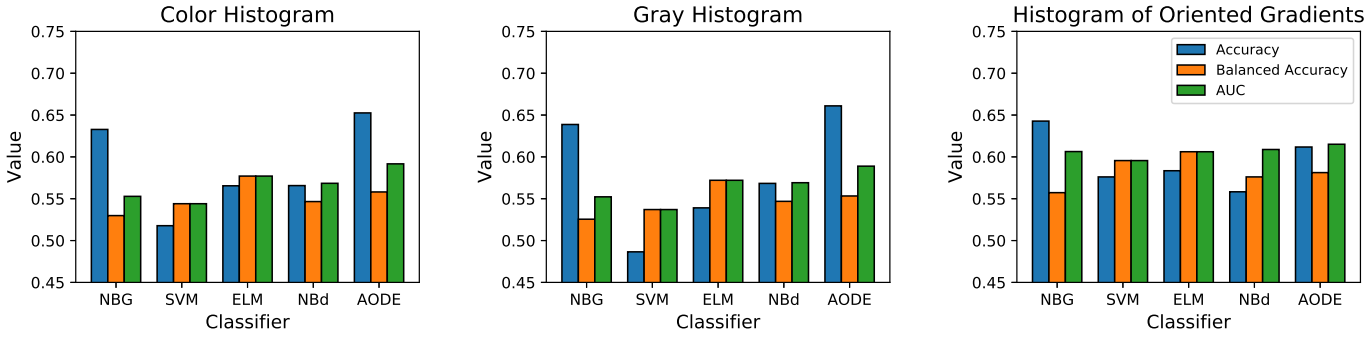


Fig. 6. Resultados de 5 clasificadores entrenados con descriptores generales de la imagen en términos de *accuracy*, *balanced accuracy* y AUC.

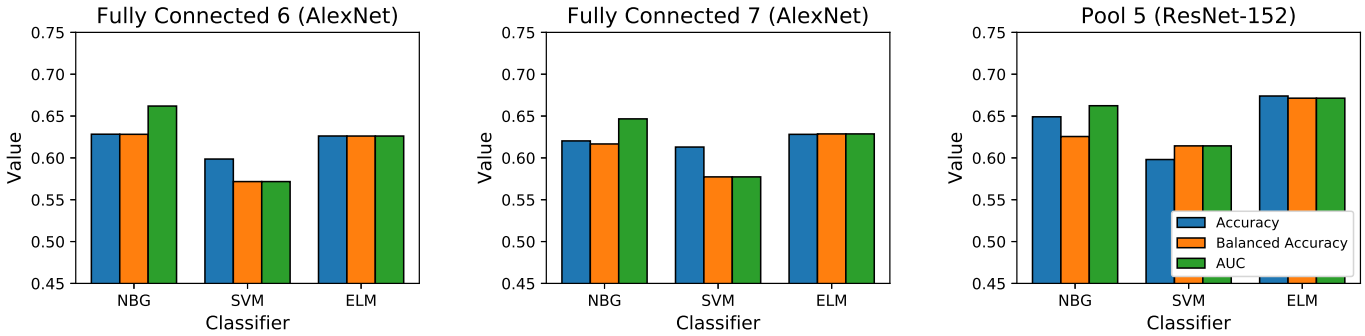


Fig. 7. Resultados de 3 clasificadores entrenados con *ConvNet features* de AlexNet y ResNet en términos de *accuracy*, *balanced accuracy* y AUC.

La reducción que vamos a ver a continuación, también tiene el objetivo de disminuir el tamaño de los vectores de características que se extraen de las redes, con el fin de poder utilizar modelos más complejos con esta técnica de Transfer Learning.

C. Reducción de las capas densas mediante finetuning

Al igual que en la extración de *ConvNet features*, para el proceso de *finetuning* es necesario disponer de modelos pre-entrenados en problemas similares para utilizar la información en nuestro beneficio.

AlexNet y VGG16 son redes convolucionales que siguen estructuras muy parecidas y que cuyas tres capas finales son del tipo *fully connected* o densas, y la última corresponde con la salida de las 1000 etiquetas del problema de ImageNet. Podemos ver la estructura de ambas redes en las Fig. 8 y 9, respectivamente. Las dos capas densas previas a la salida de la red, son llamadas "fc6" y "fc7", y tienen dimensiones de 4096 nodos en cada una.

Como se ha comentado antes, la técnica de *finetuning* consiste en realizar modificaciones a una red pre-entrenada, para adaptar la salida del modelo a nuestro problema (Fig. 2). En el caso actual, donde la evaluación de la calidad estética se realiza mediante una clasificación binaria y las redes pre-entrenadas son del problema de ImageNet, sustituimos la capa densa de salida con 1000 nodos por una de 2 nodos.

En los trabajos propuestos que hacen uso de *finetuning*, todas las capas del modelo, exceptuando la de *output*, permanecen inalterables. Sin embargo, consideramos que esta

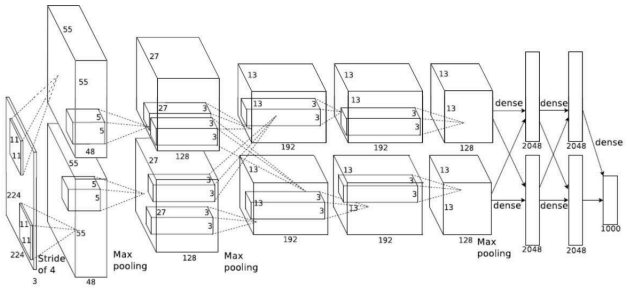


Fig. 8. Estructura original de AlexNet en dos columnas.

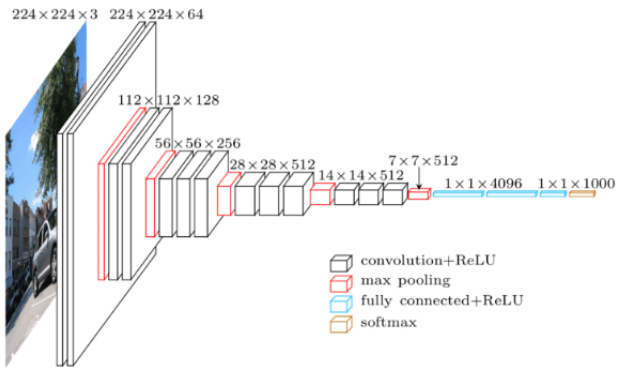


Fig. 9. Estructura de la red VGG16.

Modelo	Nodos en las capas densas	Nº de params de la red	Tamaño del modelo
AlexNet	4096	60M	223MB
	1000	14M	54MB
	500	8,6M	33MB
	250	6,1M	24MB
VGG16	4096	134M	513MB
	1000	41M	156MB
	500	28M	106MB
	250	21M	81M

TABLA I
NÚMERO DE PARÁMETROS (EN MILLONES) DE DIFERENTES REDES Y DEL ESPACIO NECESARIO EN MEMORIA (MEGABYTES) PARA ALMACENARLA, EN FUNCIÓN DEL TAMAÑO DE LAS CAPAS DENSAS.

Modelo	Nodos en las capas densas	Accuracy	Balanced Accuracy	AUC
AlexNet	4096	0.70	0.69	0.76
	1000	0.70	0.67	0.74
	500	0.69	0.67	0.73
	250	0.70	0.66	0.73
VGG16	4096	0.70	0.71	0.79
	1000	0.72	0.71	0.79
	500	0.74	0.71	0.79
	250	0.72	0.71	0.79

TABLA II
RESULTADOS EN TÉRMINOS DE ACCURACY, BALANCED ACCURACY Y AUC DE LAS REDES EN BASE AL TAMAÑO DE LAS CAPAS DENSAS.

técnica de Transfer Learning puede utilizarse de forma más eficaz aplicando modificaciones a las capas previas "fc6" y "fc7", tanto en AlexNet como en VGG16.

En este trabajo, se propone reducir el número de nodos de las tres últimas capas de los modelos AlexNet y VGG16 para resolver la clasificación binaria de la calidad estética. Las capas densas son las que concentran el mayor número de parámetros de una red, por lo que mediante este proceso, disminuiremos considerablemente el tamaño de la red.

D. Implementación

Para este trabajo se ha utilizado la librería TensorFlow [19] en Python para realizar el proceso de *finetuning* a las capas "fc6", "fc7" y el *output* de las redes neuronales AlexNet y VGG16. En ambos casos se ha utilizado el optimizador SGD (Stochastic Gradient Descent) con un learning rate de 0.001 con un tamaño de batch de 128 y se han realizado 10 epochs. Para las capas de Dropout se ha utilizado un factor de 0,5. Todo esto se ha llevado a cabo sobre una GPU Tesla K40c, donde los tiempos de entrenamiento son de 6-7 horas para AlexNet y de 23 horas para VGG16.

IV. RESULTADOS

Se han realizado experimentos reduciendo los nodos de las capas densas "fc6" y "fc7" a 1000, 500 y 250. En la tabla I se muestra la diferencia de tamaño de las redes y la memoria necesaria. Como se puede observar, la mayoría de los parámetros de nuestra red se encuentran en estas capas densas, por lo que al reducir su tamaño afecta significativamente al modelo.

Una vez reentrenadas las redes, vamos a comparar sus resultados utilizando tres métricas de evaluación sobre el conjunto de test. Estas son el *accuracy* o tasa de acierto, el *balanced accuracy* y el valor AUC (Area Under the Curve).

En la tabla II se puede observar que la modificación de las capas densas en AlexNet afecta al *balanced accuracy* y al AUC. Existe un empeoramiento de un 3% del modelo con las capas densas de tamaño 250, frente a las de 4096, pero la red reducida ocupa un 10% de memoria con respecto a la original. Con VGG16 los resultados son casi idénticos en todas las redes, independientemente del tamaño de las capas densas. Cabe destacar que sólo el *accuracy* se ha visto afectado y que la red con las capas densas de tamaño 500 funcionan mejor que las originales de 4096.

V. CONCLUSIÓN

En este trabajo se ha presentado una estrategia basada en *finetuning* capaz de reducir el tamaño de redes pre-entrenadas, en este caso AlexNet y VGG16, sin perder eficacia en la resolución del problema de la calidad estética.

Hemos visto como podemos obtener los mismos resultados con redes del 10% del tamaño de las propuestas hechas hasta ahora. Esto permitirá que los requisitos para la evaluación de nuevos casos sea mucho menor, permitiendo utilizar dichos modelos en dispositivos como móviles o, por ejemplo, una Raspberry. Estamos seguros de que es un gran paso para el desarrollo de aplicaciones reales que se beneficien del Deep Learning para la evaluación de la calidad estética.

Como trabajo futuro, planeamos seguir optimizando las redes pre-entrenadas en el problema de la calidad estética, especialmente los diseños presentados en [15], donde la salida es la distribución de los votos, en vez de la clasificación binaria en "snapshots" y "professional shots".

La reducción del número de nodos de las últimas capas de las redes neuronales, además del ahorro de memoria que supone, permite extraer *ConvNet features* con una menor dimensionalidad, pero con la misma capacidad descriptiva. Esto es de especial utilidad para aprender modelos más complejos que antes no podíamos, como por ejemplo, algunos clasificadores probabilísticos, ya que estos modelos manejan de forma natural la incertidumbre y son especialmente útiles en problemas donde la clase no está definida.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado con los fondos FEDER y el Gobierno Español (MICINN) a través del proyecto TIN2016-77902-C3-1-P.

REFERENCES

[1] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich *et al.*, "Going deeper with convolutions." Cvpr, 2015.

[5] 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.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[8] Y. Deng, C. C. Loy, and X. Tang, "Image aesthetic assessment: An experimental survey," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 80–106, 2017.

[9] R. Datta, J. Li, and J. Z. Wang, "Algorithmic inferencing of aesthetics and emotion in natural images: An exposition," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 105–108.

[10] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Studying aesthetics in photographic images using a computational approach," in *European Conference on Computer Vision*. Springer, 2006, pp. 288–301.

[11] Y. Ke, X. Tang, and F. Jing, "The design of high-level features for photo quality assessment," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 419–426.

[12] L. Marchesotti, F. Perronnin, D. Larlus, and G. Csurka, "Assessing the aesthetic quality of photographs using generic image descriptors," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1784–1791.

[13] N. Murray, L. Marchesotti, and F. Perronnin, "Ava: A large-scale database for aesthetic visual analysis," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2408–2415.

[14] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, "Rapid: Rating pictorial aesthetics using deep learning," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 457–466.

[15] H. Talebi and P. Milanfar, "Nima: Neural image assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[16] F. Rubio, M. J. Flores, and J. M. Puerta, "Drawing a baseline in aesthetic quality assessment," in *Tenth International Conference on Machine Vision (ICMV 2017)*, vol. 10696. International Society for Optics and Photonics, 2018, p. 106961M.

[17] X. Tian, Z. Dong, K. Yang, and T. Mei, "Query-dependent aesthetic model with deep learning for photo quality assessment," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2035–2048, 2015.

[18] Z. Dong, X. Shen, H. Li, and X. Tian, "Photo quality assessment with dcnn that understands image well," in *International Conference on Multimedia Modeling*. Springer, 2015, pp. 524–535.

[19] 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. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

Redes Neuronales Convolucionales para Una Clasificación Precisa de Imágenes de Corales

Anabel Gómez-Ríos, Siham Tabik, Julián Luengo and Francisco Herrera
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada
Granada, España
{anabelgrios, julianlm, herrera}@decsai.ugr.es, siham@ugr.es

ASM Shihavuddin
Dpto. Matemática Aplicada y Ciencias de la Computación
Universidad Técnica de Dinamarca
Kgs. Lyngby, Dinamarca

Bartosz Krawczyk
Dpto. Ciencias de la Computación
Universidad Virginia Commonwealth
Virginia, EEUU

***Resumen*—El reconocimiento de especies de corales basado en imágenes submarinas de texturas plantea una gran dificultad para los algoritmos de aprendizaje automático, debido a la naturaleza de los datos y las características que tienen. Entre otras, encontramos que los conjuntos de datos no incluyen información sobre la estructura global del coral, que muchas especies de coral tienen características muy similares y que definir los límites espaciales entre clases es difícil ya que muchos corales tienden a vivir juntos. Por ello, la clasificación de especies de corales siempre ha requerido de la ayuda de un experto. El objetivo de este trabajo es desarrollar un modelo de clasificación precisa para imágenes de textura de corales. Nos hemos centrado en dos conjuntos de datos de imágenes de texturas y hemos analizado 1) varias arquitecturas de redes neuronales convolucionales, 2) *transfer learning* y 3) técnicas de *data augmentation*. Hemos alcanzado los porcentajes de clasificación más altos usando diferentes variaciones de ResNet en ambos conjuntos de datos.**

***Index Terms*—Redes Neuronales Convolucionales, Clasificación, Imágenes de Corales, Inception, ResNet, DenseNet**

I. Introducción

Los arrecifes de coral son ecosistemas marinos típicos de los mares cálidos y poco profundos de los trópicos. Se crean por la acumulación de esqueletos de carbonato de calcio que las especies de coral duro dejan cuando mueren. Otros corales viven después en ellos y expanden el arrecife. Son uno de los ecosistemas más valiosos del mundo al ser extremadamente biodiversos. Soportan hasta dos millones de especies y una cuarta parte de toda la vida marina en el mundo [1]. Son también muy importantes desde el punto de vista humano [2]: ayudan a limpiar el agua eliminando el nitrógeno y el carbono, son una fuente de investigación médica y económica a partir de la pesca y el turismo, una barrera natural contra huracanes y tormentas y su estudio ayuda a comprender fenómenos climáticos del pasado debido a la cantidad de años que tienen.

Este trabajo ha sido realizado con el apoyo del Ministerio de Ciencia y Tecnología de España bajo el proyecto TIN2017-89517-P y de la Junta de Andalucía bajo el proyecto P11-TIC-7765. Siham Tabik cuenta con una beca del programa Ramón y Cajal (RYC-2015-18136) y Anabel Gómez-Ríos con una beca FPU 998758-2016.

El estudio de la distribución de los arrecifes de coral a lo largo del tiempo provee información sobre el impacto del calentamiento global y los niveles de contaminación del agua. Según [1], ya se ha perdido el 19 % del área de arrecifes de coral desde la década de 1950 y, según la Unión Internacional para la Conservación de la Naturaleza (IUCN por sus siglas en inglés) y su Lista Roja de Especies Amenazadas [3], en 2017 había 237 especies amenazadas sólo en el evaluado 40 % del total estimado de especies. Esto se debe en gran medida a problemas causados por humanos: la contaminación del agua, el cambio climático, ya que los corales no toleran los cambios de temperatura, y el dióxido de carbono emitido a la atmósfera, donde un cuarto es absorbido por el océano.

Con los avances en la adquisición de imágenes y el creciente interés de la comunidad científica, se están recolectando una gran cantidad de datos sobre corales. Sin embargo, es complicado llevar un registro de todas las especies porque hay miles de ellas y la taxonomía es mutable debido a nuevos descubrimientos o por cambios en las especies conocidas a medida que se adquiere más conocimiento sobre ellas. Además, algunas especies de coral pueden parecer idénticas entre ellas para un observador humano. Por esto, se ha necesitado siempre de un experto biólogo para obtener una buena clasificación. Si conseguimos automatizar esta clasificación usando la cantidad de imágenes de corales que se están recogiendo, podemos ayudar a los científicos a estudiar más de cerca esa cantidad de datos. De hecho, la automatización de la clasificación de imágenes de corales se ha abordado en algunos trabajos. La mayoría de ellos [4], [5], [6], [7] usan modelos de aprendizaje automático combinados con técnicas de mejora de imagen y varios extractores de características. Entre estos trabajos, sólo [6] utiliza varios conjuntos de datos.

En los últimos años, las redes neuronales convolucionales (CNNs por sus siglas en inglés) han mostrado una precisión excepcional para la clasificación de imágenes [8], [9], especialmente en el campo de la visión por computador. Actualmente, sus aplicaciones se extienden a muchos campos

en los que se requiere un análisis de imágenes. El uso de CNNs en clasificación de corales supone un reto por varias causas: las diferencias entre imágenes de la misma clase, las variaciones de luz debidas al agua o el hecho de que algunas especies de coral tienden a aparecer juntas. Por otra parte, las CNNs necesitan grandes conjuntos de datos para lograr un buen rendimiento. En la práctica, se usan dos técnicas para sobrevenir esta limitación: *transfer learning*, que consiste en usar el conocimiento obtenido de otro problema, usualmente más grande, y *data augmentation*, que consiste en aumentar artificialmente el conjunto de datos de entrenamiento. Hay algunos trabajos que utilizan CNNs para clasificación de corales [10], [11], [12], pero evalúan CNNs más simples, como VGGnet o LeNet y sólo usan un conjunto de datos.

En este trabajo se propone utilizar CNNs más nuevas y con mayor capacidad para superar las limitaciones de las CNNs que se han usado anteriormente. Queremos desarrollar un sistema más preciso que se enfrente a los problemas específicos de la clasificación de imágenes de corales usando varios conjuntos de datos. En particular, hemos considerado tres de las las CNN más prometedoras, Inception v3 [13], ResNet [14] y DenseNet [15]. Inception es una nueva versión de GoogleNet [16], que ganó el premio *ImageNet Large Scale Visual Recognition Competition* (ISLVRC) [9] en 2014. ResNet ganó la misma competición en 2015 y DenseNet mejoró los resultados de ResNet en 2016. Para la clasificación, hemos considerado dos conjuntos de imágenes submarinas de corales, RSMAS y EILAT [17], que tienen la particularidad de que muestran texturas de corales, no los corales completos. Además, son conjuntos de datos pequeños que contienen muchas clases. Por otra parte, hemos comparado nuestros resultados con el modelo actual más preciso para estos dos conjuntos de datos, el propuesto por Shihavuddin y otros en [6], el cual está compuesto por distintos algoritmos clásicos de aprendizaje automático y requiere una gran supervisión.

Las contribuciones de este trabajo son las siguientes:

- Analizar las tres CNNs que hemos evaluado en este trabajo: Inception v3, ResNet y DenseNet.
- Evaluar estas tres CNNs y analizar su rendimiento usando *transfer learning* desde ImageNet.
- Analizar distintas técnicas de *data augmentation* en este tipo concreto de imágenes.
- Comparar nuestros resultados con el actual estado del arte en EILAT y RSMAS.

El resto del trabajo está organizado de la siguiente forma: en la Sección II estudiamos y analizamos Inception, ResNet y DenseNet. En la Sección III se comentan los avances que se han hecho en la clasificación de corales basada en imágenes. En la Sección IV se exponen las características de los conjuntos de datos que se han usado en este trabajo. En la Sección V se exponen y analizan los resultados que se han obtenido con los distintos modelos de CNN y tipos de *data augmentation*. Finalmente, en la Sección VI se muestran las conclusiones obtenidas y los trabajos futuros.

II. Convolutional Neural Networks

Las CNNs han logrado precisiones excepcionales en varias aplicaciones actuales [18]. De hecho, desde 2012 la prestigiosa competición ILSVRC [9] sólo la han ganado CNNs. Las capas de una CNN capturan características cada vez más complejas a medida que aumenta la profundidad de la red. En los últimos años, estas arquitecturas han ido evolucionado, primero incrementando la profundidad de las redes, después el ancho y finalmente usando características obtenidas en las primeras capas inferiores en capas más profundas. Esta sección proporciona una visión general de las CNNs utilizadas en este trabajo. Hemos considerado tres CNNs influyentes, Inception v3 (Subsección II-A), ResNet (Subsección II-B) y DenseNet (Subsección II-B). Estas tres CNNs se basan en la repetición de un módulo o bloque base compuesto por una serie de capas convolucionales, de *poolings* o de capas de normalización. La diferencia entre ellas es la composición de los módulos y cómo están distribuidos dentro de la red. Para finalizar, describimos las técnicas de optimización que hemos usado para paliar el problema de los conjuntos de datos pequeños, *transfer learning* y *data augmentation* (Subsección II-D).

II-A. Inception v3

El módulo base de Inception [13] consta de cuatro ramas en paralelo: una convolución de kernel 1×1 seguida de dos convoluciones 3×3, una convolución 1×1 seguida de una convolución 3×3, un *pooling* seguido de una convolución 1×1 y por último una convolución 1×1. La salida del módulo es la concatenación de las cuatro ramas. Inception consta en total de 10 módulos, aunque dichos módulos se van modificando ligeramente según la red se va haciendo más profunda. En concreto, se cambian cinco de los módulos con el fin de reducir el coste computacional sustituyendo las convoluciones $n \times n$ por dos convoluciones, una 1×7 seguida de una 7×1. Los dos últimos módulos sustituyen las dos últimas convoluciones 3×3 de la primera rama por dos convoluciones cada una, una 1×3 seguida de otra 3×1, esta vez en paralelo. En total, Inception v3 tiene 42 capas con parámetros.

II-B. ResNet

ResNet [14] no tiene una profundidad fija y depende del número de módulos consecutivos que se usen. Sin embargo, aumentar la profundidad de la red para obtener una mayor precisión hace que la red sea más difícil de optimizar ya que es más fácil que se produzca el problema de la desaparición de gradientes. ResNet aborda este problema ajustando una aplicación residual en lugar de la original, y añadiendo varias conexiones entre capas. Estas nuevas conexiones saltan varias capas y realizan una identidad o una convolución 1×1. El bloque base de esta red se llama *residual block* y está compuesto, cuando la red tiene 50 o más capas, por tres convoluciones secuenciales, una 1×1, una 3×3 y una 1×1, y una conexión que une la entrada de la primera convolución a la salida de la tercera convolución. En nuestro caso hemos usado dos modelos con esta arquitectura, ResNet-50, compuesta por 50 capas y ResNet-152, compuesta por 152 capas.

II-C. DenseNet

DenseNet [15] tampoco tiene una profundidad fija y depende del número de módulos que se usen. En este caso el módulo o bloque se llama *dense block* y como el de ResNet, introduce conexiones entre capas no consecutivas. En este caso, se conecta la salida de todas las capas dentro del *dense block* a la entrada de todas las capas siguientes dentro del bloque. Las conexiones entre bloques, llamadas capas de transición, funcionan como un factor de compresión en el sentido de que generan menos mapas de características de los que reciben. El bloque está compuesto de una repetición de las siguientes operaciones: *Batch Normalization* (BN), ReLU, convolución 1×1, BN, ReLU y convolución 3×3. Las capas de transición son una convolución 1×1 seguida de un *average pooling* con kernel 2×2. En este trabajo, hemos analizado DenseNet-121 y DenseNet-161, que incluyen 121 y 161 capas.

II-D. Técnicas de Optimización de CNNs

Todas las redes anteriores son demasiado profundas como para entrenarlas desde cero con nuestros conjuntos de datos, ambos muy pequeños. Por ello, hemos usado las técnicas de *transfer learning* y *data augmentation*.

La técnica de *transfer learning* consiste en usar el conocimiento aprendido de otro problema, generalmente más grande, comenzando el entrenamiento de los pesos, en lugar de con valores aleatorios, con los pesos entrenados con otro problema. En concreto, nosotros hemos usado las redes preentrenadas con ImageNet [19], y como los conjuntos de datos siguen siendo demasiado pequeños, en lugar de ajustar todos los pesos de las redes hemos añadido dos capas completamente conectadas al final de las redes anteriores. La primera de estas dos capas tiene 512 neuronas y una activación ReLU y la última tiene tantas neuronas como clases tiene el conjunto de datos que queremos clasificar y una activación softmax. De esta forma sólo entrenamos estas nuevas dos capas.

La técnica de *data augmentation* [20] consiste en aumentar artificialmente el conjunto de entrenamiento aplicando varias transformaciones a las imágenes originales, como cambiar el brillo, escalarlas, acercarlas, girarlas, reflejarlas verticalmente, etc. Las nuevas imágenes deben ser lo suficientemente reales como para que un observador externo no sepa distinguir una imagen generada por *data augmentation* de una original. En nuestro caso hemos aplicado las siguientes transformaciones:

- Desplazamiento: consiste en desplazar las imágenes horizontal y verticalmente un número determinado de píxeles calculado como una fracción del ancho o alto de la imagen. Dicha fracción se elige aleatoriamente en cada caso en el intervalo $[0, x]$ para un x dado.
- Zoom: consiste en acercar o alejar las imágenes. Dado un valor x, cada imagen será redimensionada a un valor aleatorio en el intervalo $[1 - x, 1 + x]$.
- Rotación: consiste en rotar aleatoriamente las imágenes un ángulo determinado. Dado un valor x, cada imagen se rotará un ángulo aleatorio en el intervalo $[0, x]$.
- Reflejar: consiste en reflejar las imágenes horizontalmente de forma aleatoria.

III. AVANCES EN LA CLASIFICACIÓN DE IMÁGENES DE CORALES

En esta sección comentamos los avances previos que ha habido en la clasificación automática de imágenes de corales, mostrando los retos que conlleva dicha clasificación (Subsección III-A) y los trabajos que la han analizado previamente (Subsección III-B).

III-A. Retos en la Clasificación de Imágenes de Corales

La clasificación de especies de corales basada en imágenes de texturas de corales es compleja por las siguientes razones:

- Oclusión parcial de objetos y animales.
- Variaciones en la luz debido al movimiento de mareas y a las propiedades ópticas del agua. Es común que la única fuente de luz sea la de la cámara, lo que implica iluminación no uniforme en las imágenes obtenidas.
- Anotación subjetiva de los conjuntos de datos por diferentes expertos.
- Variación en puntos de vista, distancias y calidad de imagen.
- Variabilidad en la morfología de corales que pertenecen a la misma especie.
- Separación entre clases compleja, ya que distintas especies de corales tienden a aparecer juntas.

III-B. Trabajos Relacionados

Los trabajos previos en la clasificación de imágenes de corales se pueden dividir en dos grupos: métodos que combinan algoritmos clásicos del aprendizaje automático con algoritmos de extracción de características y métodos que usan CNNs.

La mayoría de los enfoques existentes para clasificar las imágenes de corales combinan un extractor de características con un clasificador y muestran su rendimiento utilizando un único conjunto de datos [4], [5], [7]. En el primer trabajo sobre este tema [5], los autores analizaron más hábitat marino además de corales. Utilizaron un descriptor SIFT y un modelo de bolsa de características, consistente en elegir del conjunto de entrenamiento las imágenes que son más similares a cada imagen de test. En [4] se introduce el conjunto de datos *Moorea Labeled Corals* (MLC), que contiene imágenes de gran tamaño de diferentes especies de coral, y se usa SVM junto con filtros y un descriptor de textura para clasificarlo. Obtuvieron una precisión del 83,1 % sobre este conjunto de datos utilizando las imágenes de 2008 y 2009 para entrenamiento y las imágenes de 2010 para test. En [7] se utilizó un espacio de color normalizado y la transformación discreta del coseno como descriptor de textura. Sólo utilizaron un conjunto de datos, proporcionado por el *National Oceanic and Atmospheric Administration* (NOAA) del Departamento de Comercio de EE.UU.

Estos métodos no se han probado aún con nuevos conjuntos de datos. Sólo Shihavuddin y otros [6] desarrollaron un algoritmo de clasificación unificado para varios conjuntos de datos de características diferentes, entre los que podemos encontrar a RSMAS y EILAT. Los autores combinaron múltiples técnicas

de mejora de imágenes, extractores de características y clasificadores en un modelo compuesto de seis pasos. Cada paso se compone de varios algoritmos que pueden ser obligatorios u opcionales. La clasificación se realiza usando SVM, KNN, una red neuronal o la distancia media ponderada por la densidad de probabilidad. Configurando los hiperparámetros de los algoritmos y las diferentes combinaciones entre ellos, el modelo puede adaptarse a los diferentes conjuntos de datos. Este método es el estado del arte para RSMAS y EILAT. Sin embargo, implica mucha supervisión, ya que es necesario evaluar todas las combinaciones posibles de algoritmos y los hiperparámetros de cada uno de ellos.

Por otro lado, hay varios trabajos que usan CNNs. El primero de ellos fue [10]. El autor mejoró primero las imágenes de entrada mediante corrección del color y un filtrado de suavidad de la imagen para después entrenar un modelo basado en LeNet-5 en el que la capa de entrada consistía en tres canales para una imagen en color y canales adicionales para descriptores de textura y forma.

El siguiente trabajo que usó CNNs fue [11], donde los autores utilizaron VGGnet ya entrenada en ImageNet y el conjunto de datos BENTHOZ-2015 [21] para seguir entrenando la red. BENTHOZ-2015 contiene más de 400,000 imágenes y datos de sensores asociados recogidos por un vehículo autónomo sobre Australia. Los autores extrajeron varios trozos de cada imagen centrados en diferentes píxeles usando diferentes escalas. Propusieron un mecanismo para etiquetar automáticamente las imágenes de corales de forma que se obtuviera la cobertura de los corales en la región donde se recolectaron las imágenes (clasificando las nuevas imágenes como coral o no).

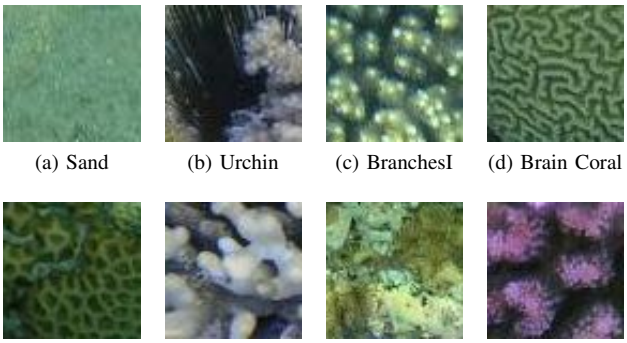
Por último, en [12] los autores utilizaron el conjunto de datos MLC y usaron CNNs junto con características extraídas de las imágenes, introduciendo un nuevo mecanismo para extraerlas. Se basaron en que las CNNs no pueden ser entrenadas desde cero usando los conjuntos de datos de corales disponibles debido a su pequeño tamaño. La extracción de características con CNNs se realizó con la red VGGnet preentrenada en ImageNet. Para clasificar ambos tipos de características, utilizaron un Perceptrón de dos capas. En sus experimentos, obtuvieron mejores precisiones con esta técnica que sólo con VGGnet, aunque la diferencia era pequeña.

Estos trabajos usan CNNs clásicas, como VGGnet o LeNet, y no usan EILAT ni RSMAS. Además, los resultados obtenidos son bajos [10], la clasificación es demasiado simple [11] o combinan las redes con otras características [12].

IV. Conjuntos de Datos

En este trabajo, hemos utilizado los conjuntos de datos RGB más recientes y pequeños que contienen el mayor número de especies de coral, RSMAS y EILAT [17]. Estos dos conjuntos de datos se componen de imágenes que muestran trozos de corales. Estos trozos capturan la textura de diferentes partes del coral y no incluyen información sobre la estructura global del mismo. Sus principales características son las siguientes:

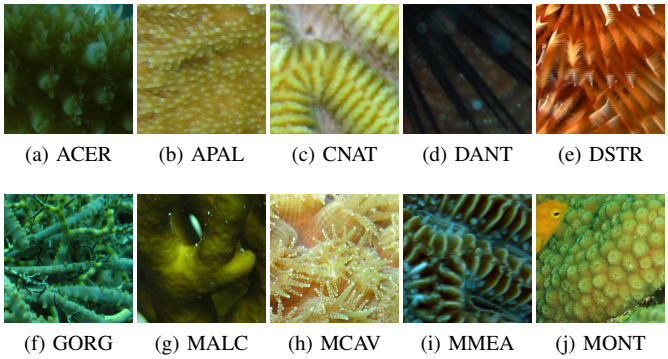
- EILAT contiene 1123 imágenes de tamaño 64×64, tomadas de arrecifes de coral cerca de Eilat en el Mar Rojo.



(a) Sand (b) Urchin (c) BranchesI (d) Brain Coral

(e) Favid Coral (f) BranchesII (g) Dead Coral (h) BranchesIII

Figura 1. Un ejemplo de cada clase de EILAT.



(a) ACER (b) APAL (c) CNAT (d) DANT (e) DSTR

(f) GORG (g) MALC (h) MCAV (i) MMEA (j) MONT

(k) PALY (l) SPO (m) SSID (n) TUNI

Figura 2. Un ejemplo de cada clase de RSMAS.

  Estas imágenes son trozos de imágenes más grandes, que fueron tomadas en iguales condiciones y con la misma cámara fotográfica. Las imágenes han sido clasificadas en ocho clases, pero las etiquetas utilizadas no corresponden a los nombres de las especies de coral. En la Fig. 1 podemos ver ejemplos de este conjunto de datos.
- RSMAS contiene 766 imágenes de tamaño 256×256. Las imágenes fueron recogidas por buzos de la Escuela Rosenstiel de Ciencias Marinas y Atmosféricas de la Universidad de Miami. Fueron tomadas bajo diferentes condiciones, con diferentes cámaras y en lugares diferentes. Las imágenes se han clasificado en 14 clases, cuyas etiquetas corresponden a los nombres en latín de la especie de coral que se muestra en cada clase. En la Fig. 2 podemos ver ejemplos de este conjunto de datos.

V. Clasificación de Imágenes de Corales usando CNNs

Esta sección está organizada en tres partes. Primero, mostramos el marco experimental que hemos usado en todos los ex-

perimentos llevados a cabo en este trabajo (Subsección V-A). Después, mostramos y analizamos los resultados de la clasificación de EILAT y RSMAS usando sólo *transfer learning* (Subsección V-B). Por último, mostramos y analizamos los resultados de las distintas técnicas de *data augmentation* en la clasificación usando el mejor modelo en cada caso hallado en la Subsección V-B (Subsección V-C).

V-A. Marco Experimental

Para evaluar todas las CNNs, hemos usado Keras [22] y Tensorflow [23] como *back-end*. Para Inception hemos usado la implementación disponible en la versión 2.0.4 de Keras y para ResNet y DenseNet, hemos adaptado el código disponible en GitHub por [24]. Todas han sido evaluadas en una gráfica NVIDIA Titan Xp.

Para la evaluación, hemos usado la medida *accuracy*, es decir, el porcentaje de acierto en la clasificación, y un esquema de validación cruzada con cinco particiones, por lo que todos los resultados que se dan en esta sección son la media de los porcentajes obtenidos sobre las cinco particiones.

Además, hemos evaluado el impacto de distintos hiperparámetros en el rendimiento de las redes, como el número de capas, el número de épocas que las entrenamos y el tamaño del *batch* que usamos para entrenarlas. En concreto, hemos usado 50 y 152 capas con ResNet, y 121 y 161 con DenseNet, con lo que tenemos cinco modelos de CNNs. Para cada uno de estos modelos, hemos probado todas las combinaciones posibles de los valores para los hiperparámetros número de épocas = $\{100, 300, 500, 700, 1000, 1300\}$ y *batch* = $\{32, 64, 128\}$.

V-B. Clasificación Usando Transfer Learning

En esta subsección analizamos el comportamiento de las redes con *transfer learning* desde ImageNet. Hemos evaluado todas las combinaciones de los hiperparámetros comentados en la subsección anterior y en la Tabla I se muestran aquellos con los que se han obtenido los mejores resultados para cada modelo junto con el *accuracy* correspondiente para dicha combinación. Se muestra además el resultado para el método de Shihavuddin, el estado del arte para estos dos conjuntos de datos. Como vemos, ResNet-152 supera al método de Shihavuddin y al resto de las CNN. Inception proporciona un mejor *accuracy* que el método de Shihavuddin en RSMAS, pero es peor en EILAT. DenseNet muestra los peores resultados en ambos conjuntos de datos. En general, estos resultados muestran que las CNNs son capaces de convertirse en el estado del arte en tareas de clasificación de corales. En RSMAS, el mejor modelo es ResNet-152, con una mejora de más del 5 % con respecto al método de Shihavuddin. En EILAT, ResNet-50 y ResNet-152 alcanzan exactamente el mismo *accuracy* y superan al método de Shihavuddin en más de un 2 %.

Los resultados obtenidos permiten concluir que sólo entrenando las últimas capas de una CNN que ya está preentrenada en ImageNet podemos superar a un método que requiere mucho tiempo y una alta supervisión humana como es el método de Shihavuddin, que se compone de seis pasos y cada paso se compone de varios algoritmos, de forma que para obtener el mejor rendimiento es necesario evaluar todas las combinaciones de algoritmos posibles a través de todos los pasos y optimizar los hiperparámetros de cada algoritmo.

V-C. Clasificación Usando Data Augmentation

En esta subsección analizamos el impacto de las técnicas de *data augmentation* enumeradas en la Subsección II-D. Tanto para EILAT como para RSMAS hemos evaluado dichas técnicas sobre los modelos que obtenían mejor rendimiento en cada caso. Para EILAT hemos usado ResNet-50 por ser más simple que ResNet-152 y obtener ambos el mismo *accuracy*. Para RSMAS hemos usado ResNet-152.

La notación `rot. = 2` significa que estamos aplicando una rotación a cada imagen de un ángulo elegido aleatoriamente para cada una en el intervalo $[0, 2]$. Esta notación es equivalente para el resto de técnicas.

La Tabla II muestra los resultados de usar *data augmentation* con ResNet-50 en EILAT y la Tabla III muestra los resultados de ResNet-152 en RSMAS. Hemos evaluado diferentes parámetros para cada técnica y varias combinaciones entre ellas, aunque en las tablas sólo se muestran las que obtuvieron un mejor resultado en cada caso. En ambos casos, la diferencia en *accuracy* entre no usar *data augmentation* y la mejor de las técnicas es muy pequeña, menos del 1 %.

La poca mejora obtenida puede explicarse por la naturaleza de las imágenes utilizadas. Dado que las imágenes originales son pequeñas y están tomadas muy de cerca, las modificaciones que aplicamos tienen que ser muy pequeñas y por tanto no tienen mucho efecto en el aprendizaje de los modelos: el desplazamiento implica perder parte de unas imágenes que ya son pequeñas y el acercamiento o zoom implica perder calidad y partes de las imágenes. Además, las imágenes están tomadas tan de cerca que la rotación y el giro no introducen variaciones significativas. Por otra parte, el rendimiento de los modelos sin usar *data augmentation* es bastante bueno, por lo que es más difícil mejorarlo.

VI. Conclusiones

La clasificación de imágenes submarinas de corales es difícil debido a la gran cantidad de especies de coral que existen, las grandes diferencias entre imágenes de una misma especie, las variaciones de luz debido al agua, o el solapamiento existente entre diferentes clases. Pocos trabajos han abordado este problema, y el único que clasifica EILAT y RSMAS es un método complejo que utiliza varios algoritmos y necesita de mucho tiempo e intervención.

Nosotros hemos abordado estos problemas utilizando algunas de las CNNs más potentes, Inception v3, ResNet y DenseNet. Hemos realizado un estudio de los fundamentos de estas CNNs, sus hiperparámetros y la posibilidad de utilizar *transfer learning* y *data augmentation*. De esta forma, hemos sido capaces de mejorar el estado del arte en EILAT y RSMAS, demostrando que las CNNs son una excelente técnica para la clasificación automática de imágenes submarinas de corales. En particular, ResNet ha sido la mejor CNN tanto en EILAT como en RSMAS.

Tabla I
Mejor accuracy, y el conjunto de hiperparámetros que lo provee en cada caso, obtenido por Inception v3, ResNet-50, ResNet-152, DenseNet-121, DenseNet-161 y el modelo de Shihavuddin. El mejor resultado está resaltado en negrita.

		Método de Shihavuddin	Inception v3	ResNet-50	ResNet-152	DenseNet-121	DenseNet-161
EILAT	*Accuracy*	95.79	96.23	**97.85**	**97.85**	91.03	93.81
	Batch	—	32	**64**	**64**	32	32
	Épocas	—	700	**500**	**300**	300	700
RSMAS	*Accuracy*	92.74	96.71	97.67	**97.95**	89.73	91.10
	Batch	—	32	64	**32**	32	64
	Épocas	—	1300	1300	**300**	700	1000

Tabla II
Mejores accuracies obtenidas por ResNet-50 en EILAT con sus mejores parámetros usando distintas técnicas de data augmentation. El mejor resultado está resaltado en negrita.

	despl. = 0.2	acer. = 0.2	rot. = 2	refl.	despl. = 0.2, acer. = 0.2
Acc.	**98.03**	97.85	97.40	97.53	97.85

Tabla III
Mejores accuracies obtenidas por ResNet-152 en RSMAS con sus mejores parámetros usando distintas técnicas de data augmentation. El mejor resultado está resaltado en negrita.

	despl. = 0.2	acer. = 0.4	rot. = 2	refl.	despl. = 0.2, acer. = 0.4
Acc.	98.36	**98.63**	97.40	97.578	98.08

Mientras que el uso de *transfer learning* sí ha dado resultados muy buenos, el uso de técnicas de *data augmentation* en este tipo de imágenes no introduce una mejora significativa en los modelos.

Este trabajo abre nuevos retos como la clasificación de especies de corales basándonos no sólo en imágenes de texturas, sino también en imágenes que contengan la estructura completa de los corales.

Referencias

[1] ESI, "Endangered species international." http://www.endangeredspeciesinternational.org/, 2017. Accessed on 13-02-2018.

[2] F. Ferrario, M. W. Beck, C. D. Storlazzi, F. Micheli, C. C. Shepard, and L. Airoldi, "The effectiveness of coral reefs for coastal hazard risk reduction and adaptation," *Nature communications*, vol. 5, p. 3794, 2014.

[3] IUCN, "Iucn red list table of number of threatened species by major groups of organisms." http://cmsdocs.s3.amazonaws.com/summarystats/2017-3_Summary_Stats_Page_Documents/2017_3_RL_Stats_Table_1.pdf, 2017. Accessed on 13-02-2018.

[4] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman, "Automated annotation of coral reef survey images," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1170–1177, IEEE, 2012.

[5] O. Pizarro, P. Rigby, M. Johnson-Roberson, S. B. Williams, and J. Colquhoun, "Towards image-based marine habitat classification," in *OCEANS 2008*, pp. 1–7, IEEE, 2008.

[6] A. Shihavuddin, N. Gracias, R. Garcia, A. C. Gleason, and B. Gintert, "Image-based coral reef classification and thematic mapping," *Remote Sensing*, vol. 5, no. 4, pp. 1809–1841, 2013.

[7] M. D. Stokes and G. B. Deane, "Automated processing of coral reef benthic images," *Limnol. Oceanogr.: Methods*, vol. 7, no. 157, pp. 157–168, 2009.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[9] 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 (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[10] M. Elawady, "Sparse coral classification using deep convolutional neural networks," *arXiv preprint arXiv:1511.09067*, 2015.

[11] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, G. Kendrick, and R. Fisher, "Automatic annotation of coral reefs using deep learning," in *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–5, IEEE, 2016.

[12] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, G. Kendrick, and R. Fisher, "Coral classification with hybrid feature representations," in *Image Processing (ICIP), 2016 IEEE International Conference on*, pp. 519–523, IEEE, 2016.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[14] 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*, pp. 770–778, 2016.

[15] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1, p. 3, 2017.

[16] 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*, pp. 1–9, 2015.

[17] A. Shihavuddin, "Coral reef dataset, v2.." Mendeley data https://data.mendeley.com/datasets/86y667257h/2, 2017. Accessed on 12-02-2018.

[18] M. D. Ferreira, D. C. Corrêa, L. G. Nonato, and R. F. de Mello, "Designing architectures of convolutional neural networks to solve practical problems," *Expert Systems with Applications*, vol. 94, pp. 205 – 217, 2018.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition (CVPR), 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[20] S. Tabik, D. Peralta, A. Herrera-Poyatos, and F. Herrera, "A snapshot of image pre-processing for convolutional neural networks: case study of mnist," *Int J Comput Intell Syst*, vol. 10, pp. 555–568, 2017.

[21] M. Bewley, A. Friedman, R. Ferrari, N. Hill, R. Hovey, N. Barrett, E. M. Marzinelli, O. Pizarro, W. Figueira, L. Meyer, *et al.*, "Australian sea-floor survey data, with images and expert annotations," *Scientific data*, vol. 2, p. 150057, 2015.

[22] F. Chollet *et al.*, "Keras." https://github.com/keras-team/keras, 2015.

[23] 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. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[24] F. Yu, "Resnet and densenet cnns in keras." https://github.com/flyyufelix/cnn_finetune, 2017.