

Fine-Grained Geometric Shapes: A Deep Classification Task

Jorge Diaz-Ramirez  Fabrizio Alvarez-Alvarez y Ximena Badilla-Torrico 

Abstract—Although the importance of Deep Learning has been well established in recent years, its role in classifying objects in images is far from being understood in fine categories and this open problem remains to be solved in geometric shapes. Here we compare deep learning models using convolutional neural networks, in order to classify fine categories in geometrical figure type images. Through the proposed method we found that there are several configurations of base models that obtain accuracies close to 80%. The proposed method also allowed us to identify that using Transfer Learning increases the accuracy by about 7% compared to the base models. Overall, these data show that the number of examples plays an important role in obtaining good classification results, as well as their quality, since noisy data in a dataset can severely reduce the generalization performance of the model in question.

Index Terms—Convolutional Neural Networks, Fine-Grained, Image Classification, Deep Learning, Transfer Learning

I. INTRODUCCIÓN

En los últimos años, la implementación de la Inteligencia Artificial (IA) en diferentes áreas se ha convertido en un factor clave y diferenciador con respecto a sus resultados de exactitud en comparación con el ser humano. Según [1], la inteligencia artificial se identifica generalmente como un generador de máquinas que se comportan como si fueran inteligentes. Asimismo, en una revisión bibliográfica sobre inteligencia artificial [2] se descubre que el problema de la IA es desarrollar un método que funcione exactamente igual que el cerebro humano. Debido al problema expuesto, la inteligencia artificial se clasifica en diferentes especializaciones, focalizándose en características propias de cada una de ellas. Una de estas es el Aprendizaje Automático (AA). El AA es una técnica que permite al computador aprender de la experiencia [3], [4]. Así también, en otro nivel de especialización, dentro del Aprendizaje Automático es el Aprendizaje Profundo (AP) [5]–[7]. En diferentes estudios, se plantea que el AP es la clave para entender como aprenden los humanos mediante la emulación de redes neuronales biológicas a través de redes neuronales artificiales, siendo un caso particular las Redes Neuronales Convolucionales (RNC). Además, se demuestra que este tipo de algoritmos realizan tareas complejas, tales como procesamiento de imágenes, textos, reconocimiento de objetos, entre otros [8]–[15]. Asimismo, los experimentos

que utilizan modelos de AP con RNC en el reconocimiento de imágenes han sugerido que el uso de estos modelos en nuevas tareas pueden alcanzar mejores resultados con la técnica llamada *Transfer Learning* [16]. Pero no solo RNC es una solución, sino también las Redes *Visio-Transformer* [17], que utilizan las redes *Transformer* focalizadas en imágenes o también *Self-Supervised Learning* [18], que surgen como alternativa cuando existen datos sin etiquetar.

Así también, con el aumento de la cantidad de imágenes disponibles, la tarea de categorización de las imágenes similares se ha vuelto compleja, las llamadas de **Grano Fino** (*Fine-grained*). Para [19] la categorización de imágenes de grano fino consiste en clasificar imágenes de una misma clase de objetos de gran tamaño y en comparación con la categorización tradicional de imágenes, representa una categorización más refinada y tiene mayores requisitos técnicos. Para [20] el objetivo de investigación del reconocimiento de imágenes de grano fino es convertir diferentes categorías a diferentes subcategorías y este reconocimiento desempeña un papel importante en el mundo real, tanto biológico como animal, transformando la tarea de reconocimiento de imágenes de grano fino en compleja. En [21] se destaca la complejidad de esta tarea, ya que los objetos con subcategorías similares pueden tener diferencias visuales marginales que incluso son difíciles de reconocer para los humanos y que el uso de RNC han sido una herramienta robusta para resolver la tarea de clasificación de grano fino. Adicionalmente, este tipo de tarea tiene como posible aplicación en la enseñanza de la geometría, especialmente de las figuras geométricas, las cuales tienen diferencias mínimas entre ellas, pudiendo ser una buena herramienta para la enseñanza. Por consiguiente y de acuerdo con [19] aquellos modelos que funcionan bien para categorizar imágenes a gran escala no siempre entregan buenos resultados al momento de categorizar imágenes de grano fino. De ahí que, resulta de interés por investigar en detalle las capacidades de los modelos de AP para lograr mejores exactitudes en este tipo de categorización de grano fino.

II. PLANTEAMIENTO DEL PROBLEMA

Para que los modelos de redes neuronales tengan buenos resultados, se necesitan de miles de datos etiquetados y no siempre se cuenta con ellos, por lo que una solución puede ser la utilización de la técnica de regularización propuesta por Srivastava [22]. Sin embargo, hasta ahora son varios los desafíos que plantea la utilización de técnicas de AP en la clasificación de imágenes de grano fino. El objetivo de esta clasificación es analizar objetos visuales pertenecientes

Jorge Diaz-Ramirez, Autor de correspondencia, profesor asistente. Facultad de Ingeniería, Universidad de Tarapacá - Iquique, e-mail:jdiazr@academicos.uta.cl

Fabrizio Alvarez-Alvarez, titulado. Facultad de Ingeniería, Universidad de Tarapacá - Iquique, e-mail:fabroaly@gmail.com

Ximena Badilla-Torrico, profesora asistente. Facultad de Ingeniería, Universidad de Tarapacá - Iquique, e-mail:xbadilla@academicos.uta.cl

a categorías subordinadas, como pueden ser diversas razas de perros, modelos de vehículos, un rectángulo con un cuadrado, pero con diferencias sutiles que no son fáciles de identificar para una máquina. Así, esta naturaleza de sutileza en las características de estas clases similares hace que el problema de clasificación sea más complejo y desafiante [23].

Son varios los trabajos donde se ha aplicado la clasificación de imágenes de grano fino, tanto en sistemas expertos, transporte inteligente y comercio electrónico [24]–[27], escenarios de la vida real [20] y de manera general en tareas de categorización [28]–[35] y clasificación [20], [36]–[41]. En dichos trabajos se utilizan conjuntos de datos creados en específico para este tipo de tareas (*CUB-200-2011*, *Stanford Dogs*, entre otras), no obstante, no se han encontrado trabajos relacionados con clasificación de figuras geométricas tales como cuadrados, rectángulos, círculos, elipses, triángulos y trapecios. Solamente se han creado conjuntos de datos con este tipo de figuras [42]. Por tanto, no queda claro como establecer modelos con mejor exactitud para el caso de clasificación de categorías finas de figuras geométricas, transformándose en un problema abierto para resolver con AP. Es por lo anterior que el objetivo de este trabajo es comparar modelos de aprendizaje profundo utilizando Redes Neuronales Convolucionales, con el fin de clasificar categorías finas en imágenes del tipo figuras geométricas.

III. SOLUCIÓN PROPUESTA

Particularmente para esta investigación se trabajó con clasificación, la cual consiste en asignar instancias de un dominio determinado, descritas por un conjunto de atributos de valor discreto o continuo, a un conjunto de clases, que pueden considerarse valores de un atributo objetivo [43]. Además, para este trabajo se consideran las siguientes clases similares (categorías finas): 1) Círculo – Elipse, 2) Cuadrado – Rectángulo – Trapecio y 3) Rombo – Triángulo. Así también para la reproducibilidad de los experimentos los códigos se encuentran disponibles en https://github.com/jdiazram/DEEP_GEOM.

A. Preprocesamiento

En el área de la clasificación de categorías finas, varios autores acuden a conjunto de datos tales como *CUB-200-2011*, *Stanford Dogs*, *Stanford Cars* y *Goldfinch* para entrenar sus modelos [44]–[46], ya que estas poseen subclases de una categoría en particular, como las razas de perros, modelos de vehículos, entre otros. En algunos casos, crean sus propios conjuntos de imágenes basándose en estas bases de imágenes ya mencionadas, con el propósito de obtener un volumen mayor de categorías y con mejor calidad [44], a su vez obteniendo resultados superiores a lo establecido. Para este proyecto se sigue este último ejemplo, creando un conjunto de datos propio, alimentado por imágenes del tipo figuras geométricas y no solo sintéticas, ni de un solo color o blanco y negro, sino también de imágenes que se usan en diversas situaciones. Es por lo anterior que el conjunto de datos (Fig. 1) a utilizar se creó de la siguiente forma:

- **Figuras generales:** La obtención de estas figuras se realizó mediante una búsqueda sistemática de imágenes en

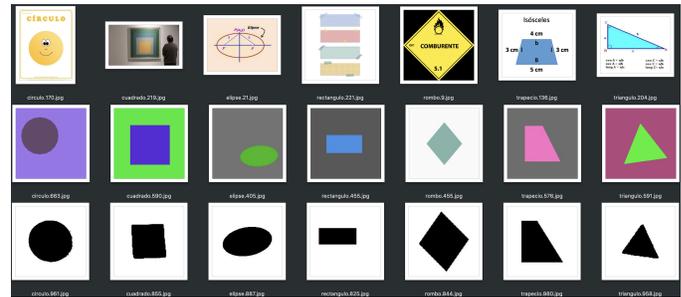


Fig. 1. Ejemplos del conjunto de datos usado. 1era fila imágenes generales, 2da fila imágenes sintéticas a color y 3ra fila imágenes en blanco y negro.

Internet según las clases definidas. Se logró obtener 3201 imágenes en total de las diferentes clases en cuestión.

- **Figuras sintéticas blanco y negro:** La obtención de estas figuras se realizó a través del sitio de Kaggle [47], donde se encuentran 16.000 imágenes de cuatro figuras, tales como triángulos, estrellas, círculos y cuadrados. El resto de clases se generaron para este trabajo.
- **Figuras sintéticas color:** La obtención de estas figuras se realizó a través del sitio de ScienceDirect [48], de Korchi [42], que cuentan con figuras de triángulos, cuadrados, pentágonos, hexágonos, heptágonos, octógonos, nonágonos, círculos y estrellas. Cada una de ellas con 10.000 ejemplos. El resto de clases se generaron para este trabajo.

Implícitamente al clasificar se quiere generalizar sin sesgar los resultados, por lo mismo para este trabajo se mantiene un equilibrio en la cantidad de las figuras, como se muestra en tabla I. Para entrenamiento se usaron 1000 imágenes por clase, para validación 125 imágenes en total por clase y para pruebas 125 imágenes en total por clase. Lo anterior para tener una separación adecuada (*Data Splitting*) con 80% para entrenamiento, 10% para validación y 10% para pruebas. Así también, esta separación permitió mantener el hecho que la cantidad de imágenes sintéticas en total no sobrepasaran las imágenes generales, salvo para Elipse y Rectángulo, ya que se tenía que completar la base de 1000 para entrenamiento, 125 para validación y 125 para pruebas. Finalmente, se realizó la normalización y escalado, para tener un conjunto de imágenes estándar, llegando a dimensiones de las imágenes de 150 por 150 píxeles y 3 canales. Luego, en la siguiente etapa, se consideran dos tipos de configuraciones generales, una para los **Modelos Base** a utilizar y la otra para modelos con técnicas de **Transferencia de Aprendizaje** (*Transfer Learning*), esto con el fin de realizar comparaciones en sus rendimientos.

B. Experimentos: Modelos Base

Para la generación de los modelos base en estos experimentos se utilizó Keras [49]. Particularmente se siguió una configuración estándar para arquitecturas de Redes Neuronales Convolucionales, con las siguientes capas: Capas convolucionales, Capas de Agrupación y Capas Completamente Conectadas. Así, se utilizaron y ejecutaron modelos base, con las configuraciones que se muestra en tabla II, donde

TABLA I
DISTRIBUCIÓN DE CANTIDAD DE FIGURAS GEOMÉTRICAS

Figura	Entrenamiento				Validación				Pruebas				Total	Total Gral
	Gral	Color	BN	Total	Gral	Color	BN	Total	Gral	Color	BN	Total		
Círculo	521	239	240	1000	65	30	30	125	65	30	30	125	1250	
Elipse	289	355	356	1000	36	44	45	125	36	44	45	125	1250	
Cuadrado	366	317	317	1000	46	39	40	125	45	40	40	125	1250	
Rectángulo	315	343	342	1000	40	42	43	125	39	43	43	125	1250	
Trapezio	346	326	328	1000	43	41	41	125	44	40	41	125	1250	
Rombo	338	331	331	1000	42	41	42	125	42	41	42	125	1250	
Triángulo	386	307	307	1000	48	39	38	125	49	38	38	125	1250	
Total	2561	2218	2221	7000	320	276	279	875	320	276	279	875	8750	

se observa desde el modelo 1 hasta el modelo 9. Para cada modelo, la configuración se fue incrementando en: cantidad de capas convolucionales, cantidad de filtros, tamaño del núcleo que variaron entre (3,3) y (5,5) por capa, número de capas de agrupamiento, unidades para las capas completamente conectadas, partiendo por la final de 7. Se fueron agregando más capas densas hasta llegar a tener 512 unidades (neuronas) por modelo. Además, cada uno de estos modelos, tienen componentes comunes entre ellos, por lo que no aparecen en la tabla, tales como: épocas, pasos por épocas, una función de activación para las capas convolucionales, la que fue ReLu (*Rectified Linear Unit*) [50], la capa de reducción utilizada fue *Maxpooling2D* y su tamaño de ventana (*Pool Size*) es de (2,2). También se utilizó la función *Flatten*, capa previa a la capa totalmente conectada. Para la capa totalmente conectada, las unidades de salida fueron de 7, esto dado que las clases eran 7. Finalmente, para la función de activación en esta capa se utilizó *Softmax* [51], entregando probabilidades categóricas de resultados esperados.

Para estos experimentos, se utilizaron y ejecutaron los mismos modelos de la tabla II, manteniendo como base 30 épocas y 30 pasos por épocas. Luego a cada capa densa, previamente se agregó la función *Dropout*, con un valor estándar de 0.3 para prevenir el sobreajuste que se pueda provocar por la cantidad de imágenes en los modelos. Después se utilizó la técnica de Aumento de Datos (*Data Augmentation*), con lo cual se generó una cantidad de datos ficticios desde las mismas imágenes de entrada, obteniendo una cantidad similar de forma aleatoria por época, esto es 1000 imágenes por cada clase en cada época. Además de los experimentos anteriores, se generaron otras configuraciones distintas para los 9 modelos base, modificando las épocas (30 y 100), pasos por épocas (30, 68 y 233), con regularización y aumento de datos, generando nuevos resultados, para un total de 8 experimentos generales. Estas configuraciones fueron las que mejor resultados nos entregaron los modelos. Al finalizar esta etapa se cuenta con 72 experimentos particulares, 8 por cada uno de los 9 modelos y configuraciones descritas anteriormente, donde cada experimento entregó diferentes resultados, los que se describirán más adelante, en la sección de resultados.

C. Experimentos: Modelos con Transferencia de Aprendizaje

Para aprovechar todo el potencial de los modelos pre-entrenados, tanto en su capacidad de aprendizaje adquirida

y cantidad de imágenes usadas en su entrenamiento [21], se decidió en esta etapa usar *Transfer Learning* para mejorar los resultados de los modelos base. Es así como el mejor modelo de los experimentos anteriores fue M6 con una exactitud de 80,5%, entonces para estos nuevos experimentos se utilizó la configuración del modelo anterior (M5) como base para la selección de modelos con técnicas de *Transfer Learning*, siendo esta la primera etapa en estos experimentos. Con esto se realizó un filtro de 12 modelos preentrenados con una configuración de: 233 pasos por épocas y 100 épocas y se seleccionaron los 6 modelos con exactitud superior a 80% (ResNet50V2, ResNet101V2, ResNet152V2, DenseNet121, DenseNet201, InceptionResNetV2)

En la segunda etapa también se procedió a realizar varios experimentos con distintas configuraciones descritas en la tabla III. En esta tabla se observa la configuración 0 base, donde Modelo Preentrenado fue el modelo para la técnica de *Transfer Learning* utilizada, luego una función *Flatten* y una capa densa (*Dense*) con 7 neuronas. Posterior a esto y a medida que se tenía una nueva configuración, se fueron agregando más capas, hasta llegar a la configuración 4.

D. Mediciones

Para los distintos experimentos, se utilizaron diferentes medidas de rendimiento [52]. Para el modelo en general se utilizó **Exactitud** (*Accuracy*) y para el análisis los distintos resultados obtenidos por cada clase se utilizaron las medidas: *Precision*, *Recall*, *F1-score* y *Support* (cantidad de instancias utilizadas).

IV. RESULTADOS Y DISCUSIÓN

A. Modelos Base

En la tabla IV se muestran los resultados obtenidos de los experimentos utilizando los modelos base. Cada columna representa una configuración distinta utilizada, -DO significa sin utilizar *Dropout*, +DO significa que se utilizó *Dropout*, +DA significa que se utilizó Aumento de Datos (*Data Augmentation*). También a modo de ejemplo para 30/30, el 1er valor (30) significa 30 pasos por épocas y el 2do valor (30) significa 30 épocas y esta nomenclatura se observa en las diferentes columnas. Además, como se puede apreciar en la tabla IV, se tiene que el modelo 6 logra la mayor exactitud con un 80,50%, esta se obtuvo utilizando técnicas de *Data Augmentation*, sin

TABLA II
RESUMEN DE CONFIGURACIONES DE LOS MODELOS BASE

Modelo	Capas Conv.	Filtros	Tamaño del núcleo	Capas de agrup.	Un. capa densa
M1	1	16	(3,3)	1	7
M2	2	16-32	(3,3)-(3,3)	2	7
M3	3	16-32-64	(3,3)-(3,3)-(3,3)	3	7
M4	4	16-32-64-128	(3,3)-(3,3)-(5,5)-(5,5)	4	7
M5	5	16-32-64-128-128	(3,3)-(3,3)-(5,5)-(5,5)-(5,5)	5	7
M6	5	16-32-64-128-128	(3,3)-(3,3)-(5,5)-(5,5)-(5,5)	5	128-7
M7	5	16-32-64-128-128	(3,3)-(3,3)-(5,5)-(5,5)-(5,5)	5	256-128-7
M8	5	16-32-64-128-128	(3,3)-(3,3)-(5,5)-(5,5)-(5,5)	5	512-256-128-7
M9	5	16-32-64-128-128	(3,3)-(3,3)-(5,5)-(5,5)-(5,5)	5	512-512-256-128-7

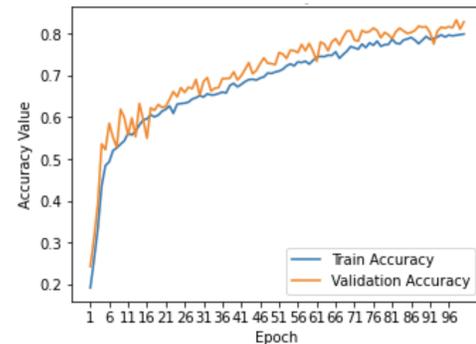
TABLA III
RESUMEN DE CONFIGURACIONES DE LAS TÉCNICAS PARA TRANSFER LEARNING

Etapa 1 Configuración 0	Etapa 2 Configuración 1	Etapa 2 Configuración 2	Etapa 2 Configuración 3	Etapa 2 Configuración 4
Modelo Preentrenado Flatten Dense(7)	Modelo Preentrenado Flatten Dense(128) Dense(7)	Modelo Preentrenado Flatten Dense(128) Dense(256) Dense(7)	Modelo Preentrenado Flatten Dense(128) Dense(256) Dense(512) Dense(7)	Modelo Preentrenado Flatten Dense(128) Dense(256) Dense(512) Dense(7)

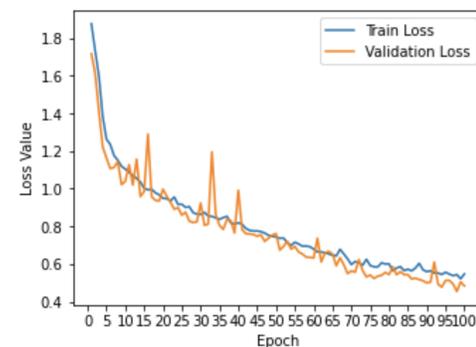
Dropout y con 233 pasos por época para un total de 100 épocas. En la Fig. 2 se pueden observar los gráficos que detallan el progreso del entrenamiento a medida que las épocas se incrementaban. La curva de exactitud en Fig. 2a muestra que no hay un sobreajuste en el modelo entrenado, dado que no hay diferencia significativa entre el conjunto de entrenamiento (*training set*) y conjunto de validación (*validation set*), a medida que avanzan las épocas. Mientras que, la curva de pérdida en Fig. 2b, el conjunto de validación es inferior a la del conjunto de entrenamiento, lo cual podría indicar un subajuste, pero no alcanza a ser significativo para considerarlo como tal.

En la tabla V, se observan los resultados de *Precision*, *Recall*, *F1-score* y *Support* para el modelo 6 con mejor Exactitud. Los resultados muestran que la **Elipse** es la figura con mayor *Precision* que el modelo puede clasificar con un 96,8%, mientras que el **Triángulo** es la que presenta la más baja puntuación con 69,0%. El resto de las figuras geométricas presentan valores dentro de los rangos de 75% a 95%, estando equilibrados sus resultados. En *Recall*, **Círculo** es la mejor clasificada con 96,8% y **Elipse** la más baja en un 73,6%. En *F1-score* la clase mejor clasificada es **Rectángulo** con un 84,4% y **Cuadrado** la más baja con un 74,8%. En términos generales el modelo reconoce y diferencia las figuras geométricas en el rango de 69,0% a 96,8%, siendo un resultado excelente, ya que *Precision* mide la habilidad del modelo de no etiquetar positiva una muestra negativa, *Recall* mide la habilidad del modelo de encontrar las muestras positivas y *F1-score* es una media armónica ponderada entre *Precision* y *Recall* mostrándose equilibrado en sus resultados.

Para las categorías finas establecidas:



(a) Exactitud



(b) Pérdida

Fig. 2. Exactitud y Función de Pérdida del modelo 6, utilizando aumento de datos, 233 pasos por época y 100 épocas.

- **Círculo – Elipse:** Se observa que el modelo clasifica mejor **Elipse** que **Círculo** por sobre un 21% de *Precision*, también para *Recall* **Círculo** esta por sobre **Elipse** en un

TABLA IV
RESULTADOS DE EXACTITUD CON MODELOS BASE

Modelo	-DO +DA 30/30	-DO +DA 30/100	-DO +DA 86/100	-DO +DA 233/100	+DO +DA 30/30	+DO +DA 30/100	+DO +DA 86/100	+DO +DA 233/100
M1	31,50%	28,70%	24,90%	27,20%	30,60%	28,80%	25,50%	25,90%
M2	32,80%	24,00%	30,60%	41,80%	32,50%	28,90%	31,30%	41,00%
M3	31,30%	43,10%	53,10%	65,40%	36,10%	35,50%	54,30%	63,90%
M4	50,50%	64,50%	77,00%	74,10%	55,00%	55,60%	74,90%	80,10%
M5	58,60%	61,40%	78,30%	79,50%	50,30%	67,10%	71,00%	78,10%
M6	57,40%	59,70%	70,20%	80,50%	36,10%	64,80%	71,50%	75,80%
M7	42,10%	61,60%	70,90%	78,30%	44,20%	63,10%	64,00%	72,10%
M8	39,90%	49,30%	68,60%	77,00%	34,70%	30,10%	48,20%	71,80%
M9	35,90%	46,50%	59,30%	69,10%	32,90%	19,70%	36,60%	61,30%

TABLA V
MEDICIONES DEL MODELO BASE 6 POR CLASE

Clase	Precision	Recall	F1-score	Support
Círculo	75,8%	92,8%	83,5%	125
Elipse	96,8%	73,6%	83,6%	125
Cuadrado	71,5%	78,4%	74,8%	125
Rectángulo	95,0%	76,0%	84,4%	125
Trapezio	86,2%	75,2%	80,3%	125
Rombo	81,3%	80,0%	80,6%	125
Triángulo	69,0%	87,2%	77,0%	125

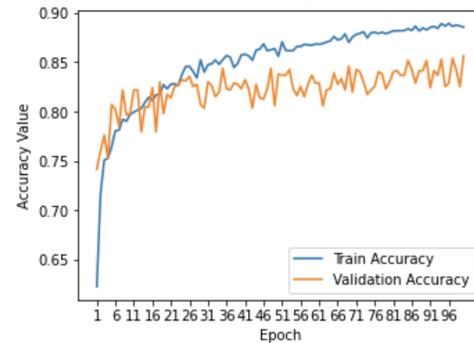
19,2% y *F1-score* se mantienen casi constantes en un 83,5%.

- **Cuadrado – Rectángulo – Trapecio:** Se observa que el modelo clasifica de mejor manera a **Rectángulo** por sobre **Cuadrado** en un 23,5% de *Precision* y **Trapezio** por sobre un 8,8%, también para *Recall* se mantienen equilibrados entre 75,2% y 78,4%. Para *F1-score* **Rectángulo** es superior a sus clases similares.
- **Rombo – Triángulo:** Se observa que el modelo clasifica a **Rombo** por sobre **Triángulo** en un 12,3% de *Precision*. Para *Recall* **Triángulo** esta por sobre **Rombo** en un 7,2% y en *F1-score* **Rombo** sobrepasa a **Triángulo** en un 3,6%.

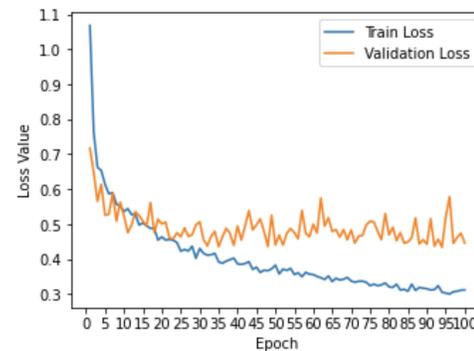
B. Modelos con Transferencia de Aprendizaje

Teniendo en cuenta los resultados obtenidos en los experimentos que se muestran en la tabla IV, a continuación, se buscó determinar qué modelo presentaba mayor exactitud en cada una de las configuraciones. En la tabla VI muestra la exactitud de cada uno de los siete modelos de *Transfer Learning* para cada una de las cinco configuraciones, una vez aplicados al conjunto de datos considerados en este trabajo. En esta tabla también se observa el modelo con mejor exactitud, siendo *InceptionResnet* versión 2 con **87,10%**. Estos experimentos tuvieron 233 pasos por épocas y 100 épocas, la diferencia entre ellos fue la configuración descrita en tabla III. Así también, se realizaron otros experimentos con diferentes épocas y pasos por épocas, pero estos obtuvieron bajas exactitudes.

En la tabla VII se observa el detalle del modelo *InceptionResNetV2* por cada clase. Para *Precision* la clase mejor clasificada fue **Rectángulo** con un 93,5% y **Rombo** la más baja con un 82,3%. Para *Recall* la mejor clase fue **Círculo** con un 93,6% y la más baja fue **Cuadrado** con un 79,2%. Así también para *F1-score* **Círculo** es la más alta clase clasificada y **Cuadrado** la más baja. Estos resultados demuestran lo buen clasificador que es el modelo de *InceptionResNetV2* para el conjunto de datos utilizado.



(a) Exactitud



(b) Pérdida

Fig. 3. Resultados de Exactitud y Función de Pérdida en Inception-ResNetV2

Otra característica importante a analizar es el comportamiento del modelo a medida que aprende a través de las épocas. En Fig. 3 se observa la curva de Exactitud y Función de Pérdida. Para Exactitud, Fig. 3a, la curva se comporta de manera incremental su aprendizaje, llegando a un techo por bajo el 90% y luego de eso su comportamiento es constante, lo que quiere decir que ya en tempranas épocas pudo entender el modelo como clasificar las diferentes clases similares propuestas. Para la Función de Pérdida, Fig. 3b, se observa un decaimiento adecuado en relación que avanzan las épocas, mas pronunciado para entrenamiento que validación, pero con tendencia a mantenerse constante al final de las 100 épocas. Con lo anterior se muestra que aún se puede mejorar el conjunto de datos para la representatividad de las características de las clases similares.

TABLA VI
PORCENTAJES DE EXACTITUD CON MODELOS PRE-ENTRENADOS

Modelo	Exactitud conf. 0	Exactitud conf. 1	Exactitud conf. 2	Exactitud conf. 3	Exactitud conf. 4
InceptionResNetV2 [53]	81,00%	86,30%	87,10%	85,70%	85,70%
DenseNet201 [54]	81,50%	80,10%	80,20%	81,80%	81,00%
DenseNet121 [54]	81,80%	84,00%	84,80%	83,30%	85,60%
ResNet152V2 [55]	83,20%	86,30%	86,10%	85,80%	84,80%
ResNet101V2 [55]	84,20%	85,60%	85,50%	85,00%	85,40%
ResNet50V2 [55]	84,60%	85,50%	83,30%	84,60%	85,50%

TABLA VII
MEDICIONES DE INCEPTIONRESNETV2 POR CLASE

Clase	Precision	Recall	F1-score	Support
Círculo	85,4%	93,6%	89,3%	125
Cuadrado	87,6%	79,2%	83,2%	125
Elipse	86,5%	87,2%	86,9%	125
Rectángulo	93,5%	80,0%	86,2%	125
Rombo	82,3%	92,8%	87,2%	125
Trapezio	88,8%	88,8%	88,8%	125
Triángulo	87,3%	88,0%	87,6%	125

V. CONCLUSIONES

La cantidad de datos para usar en este tipo de técnicas de *Aprendizaje Profundo* juega un rol importante para obtener buenos resultados de clasificación, especialmente cuando se trata de categorías finas como las analizadas en este trabajo. Al intentar categorizar clases tan similares, los modelos suelen luchar por alcanzar niveles de precisión aceptables, ya que las características que suelen diferenciar una clase de otra son muy sutiles, haciendo que la labor de clasificación sea mucho más compleja. Se logra comparar modelos de aprendizaje profundo utilizando redes neuronales convolucionales, con el fin de clasificar categorías finas en imágenes del tipo figuras geométricas, obteniendo resultados de 80,50% para M6 y 87,10% de exactitud para InceptionResNetV2. De acuerdo con las referencias utilizadas en este estudio, los resultados se consideran aceptables de acuerdo con la configuración de modelos utilizada, puesto que al aplicar la técnica de *transferencia de aprendizaje* se espera tener mejoras con *finetuning* (modificación del modelo para *Transfer Learning*), lo que representa una mejora del 7% en la exactitud del modelo. Es así como, tanto la configuración del modelo como así también factores como la calidad y cantidad de imágenes disponibles juegan un rol importante en la obtención de mejores resultados. Seguir mejorando las técnicas y arquitecturas de estos modelos, como también el conjunto de datos, es un trabajo futuro a seguir investigando para obtener mejores resultados en el ámbito de clasificación de imágenes de categorías finas en el tipo de figuras geométricas.

REFERENCES

- [1] S. Russel and P. Norvig, *Inteligencia Artificial Un enfoque Moderno*, vol. 1. 2da ed., 2004.
- [2] C. V. Krishna, H. R. Rohit, and Mohana, "A review of artificial intelligence methods for data science and data analytics: Applications and research challenges," in *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018*, pp. 591–594, Institute of Electrical and Electronics Engineers Inc., 2019.
- [3] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, pp. 210–229, 7 1959.
- [4] T. M. Mitchell, "Does Machine Learning Really Work?," *AI Magazine*, vol. 18, no. 3, pp. 11–20, 1997.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [6] J. Salas, F. De Barros Vidal, and F. Martinez-Trinidad, "Deep Learning: Current State," *IEEE Latin America Transactions*, vol. 17, pp. 1925–1945, 12 2019.
- [7] L. C. Soto-Ayala and J. A. Cantoral-Ceballos, "Automatic Blood-Cell Classification via Convolutional Neural Networks and Transfer Learning," *IEEE Latin America Transactions*, vol. 19, pp. 2028–2036, 5 2021.
- [8] Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, "Handwritten digit recognition: applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, 1989.
- [9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.
- [11] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, 10 2014.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.
- [13] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.
- [14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.)*, vol. 28, pp. 91–99, Curran Associates, Inc., 2015.
- [15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.
- [16] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *Advances in Computational Intelligence Systems (A. Lotfi, H. Bouchachia, A. Gegov, C. Langensiepen, and M. McGinnity, eds.)*, (Cham), pp. 191–202, Springer International Publishing, 2019.
- [17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.
- [18] F. A. Breiki, M. Ridzuan, and R. Grandhe, "Self-supervised learning for fine-grained image classification," 2021.
- [19] Z. Lv, L. Qiao, A. K. Singh, and Q. Wang, "Fine-Grained Visual Computing Based on Deep Learning," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, 4 2021.
- [20] J. Luo, Y. Jiang, and J. Qiu, "Fine-Grained Object Recognition Based on Multi-Scale Destruction and Construction Learning," in *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering, EITCE 2021*, (New York, NY, USA), pp. 1227–1232, Association for Computing Machinery, 2021.

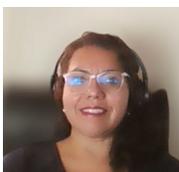
- [21] B. Zhao, J. Feng, X. Wu, and S. Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," *International Journal of Automation and Computing*, vol. 14, pp. 119–135, Apr 2017.
- [22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [23] X.-S. Wei, J. Wu, and Q. Cui, "Deep Learning for Fine-Grained Image Analysis: A Survey," 2019.
- [24] Y. Huang, J. Chen, W. Ouyang, W. Wan, and Y. Xue, "Image Captioning With End-to-End Attribute Detection and Subsequent Attributes Prediction," *IEEE Transactions on Image Processing*, vol. 29, pp. 4013–4026, 2020.
- [25] T. Soenen, W. Tavernier, M. Peuster, F. Vicens, G. Xilouris, S. Kolometsov, M.-A. Kourtis, and D. Colle, "Empowering Network Service Developers: Enhanced NFV DevOps and Programmable MANO," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 89–95, 2019.
- [26] A. Wentzel, P. Hanula, T. Luciani, B. Elgohari, H. Elhalawani, G. Canahuate, D. Vock, C. D. Fuller, and G. E. Marai, "Cohort-based T-SSIM Visual Computing for Radiation Therapy Prediction and Exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 949–959, 2020.
- [27] X. Shu, J. Tang, G.-J. Qi, Z. Li, Y.-G. Jiang, and S. Yan, "Image Classification With Tailored Fine-Grained Dictionaries," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 454–467, 2018.
- [28] Y. Tan, M. M. Rahman, Y. Yan, J. Xue, L. Shao, and K. Lu, "Fine-Grained Categorization From RGB-D Images," *IEEE Transactions on Multimedia*, vol. 24, pp. 917–928, 2022.
- [29] Y. Hu, X. Jiang, X. Liu, X. Luo, Y. Hu, X. Cao, B. Zhang, and J. Zhang, "Hierarchical Self-Distilled Feature Learning for Fine-Grained Visual Categorization," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.
- [30] Z. Pan, X. Yu, M. Zhang, and Y. Gao, "Mask-Guided Feature Extraction and Augmentation for Ultra-Fine-Grained Visual Categorization," in *2021 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2021.
- [31] Q. Wang, K. Zhang, J. Fan, S. Huang, and L. Zhang, "Multi-Order Feature Statistical Model for Fine-Grained Visual Categorization," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7379–7386, 2021.
- [32] K. Xu, R. Lai, L. Gu, and Y. Li, "Multiresolution Discriminative Mixup Network for Fine-Grained Visual Categorization," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.
- [33] J. Han, X. Yao, G. Cheng, X. Feng, and D. Xu, "P-CNN: Part-Based Convolutional Neural Networks for Fine-Grained Visual Categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 579–590, 2022.
- [34] M. Gwilliam, A. Teuscher, C. Anderson, and R. Farrell, "Fair Comparison: Quantifying Variance in Results for Fine-grained Visual Categorization," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 3308–3317, 2021.
- [35] M. Zhu, S. Wan, P. Jin, and Q. Tian, "A Feature Fusion Method Based on Multi-Classification Losses for Fine-Grained Visual Categorization," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 6072–6074, 2021.
- [36] M. Li, L. He, C. Lei, and Y. Gong, "Fine-grained image classification model based on improved SqueezeNet," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 393–399, 2021.
- [37] Y. Seo and K.-s. Shin, "Image classification of fine-grained fashion image based on style using pre-trained convolutional neural network," in *2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)*, pp. 387–390, 2018.
- [38] C. Qiu, J. Cui, S. Zhang, C. Wang, Z. Gu, H. Zheng, and B. Zheng, "Transfer Learning for Small-Scale Fish Image Classification," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–5, 2018.
- [39] R. Sipes and D. Li, "Using Convolutional Neural Networks for Automated Fine Grained Image Classification of Acute Lymphoblastic Leukemia," in *2018 3rd International Conference on Computational Intelligence and Applications (ICCIA)*, pp. 157–161, 2018.
- [40] Y.-H. Chen and M.-C. Yeh, "Text-Enhanced Attribute-Based Attention for Generalized Zero-Shot Fine-Grained Image Classification," in *Proceedings of the 2021 International Conference on Multimedia Retrieval, ICMR '21*, (New York, NY, USA), pp. 447–450, Association for Computing Machinery, 2021.
- [41] X. Yang, J. Hu, Z. Wang, F. Xu, and L. Zhu, "Self-Supervised Fine-Grained Image Classification via Progressive Global Disturbance," in *2021 4th International Conference on Computer Science and Software Engineering (CSSE 2021)*, CSSE 2021, (New York, NY, USA), pp. 119–125, Association for Computing Machinery, 2021.
- [42] A. E. Korchi and Y. Ghanou, "2D geometric shapes dataset – for machine learning and pattern recognition," *Data in Brief*, vol. 32, p. 106090, 2020.
- [43] P. Cichosz, *Data mining algorithms: explained using R*. New York: Wiley, 2015.
- [44] S. Ming, Y. Yuan, Z. Feng, and D. Errui, "Multi-Attention Multi-Class Constraint for Fine-grained Image Recognition," in *Computer Vision – ECCV 2018* (F. Vittorio, M. Hebert, S. Cristian, and W. Yair, eds.), (Cham), pp. 834–850, Springer International Publishing, 2018.
- [45] Z. Ning, J. Donahue, G. Ross, and D. Trevor, "Part-Based R-CNNs for Fine-Grained Category Detection," in *Computer Vision – ECCV 2014* (F. David, T. Pajdla, S. Bernt, and T. Tinne, eds.), (Cham), pp. 834–849, Springer International Publishing, 2014.
- [46] Y. Ze, T. Luo, W. Dong, H. Zhiqiang, G. Jun, and W. Liwei, "Learning to Navigate for Fine-Grained Classification," in *Computer Vision – ECCV 2018* (F. Vittorio, M. Hebert, S. Cristian, and W. Yair, eds.), (Cham), pp. 438–454, Springer International Publishing, 2018.
- [47] "https://www.kaggle.com/smeschke/four-shapes,"
- [48] "https://www.sciencedirect.com/science/article/pii/S2352340920309847,"
- [49] F. Chollet, "Keras: the Python deep learning API."
- [50] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, (Madison, WI, USA), pp. 807–814, Omnipress, 2010.
- [51] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," 11 2018.
- [52] Sklearn, "Metrics precision, recall, fscore and support."
- [53] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.
- [54] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.



Jorge Díaz-Ramírez graduated in computer engineering from Universidad de Tarapacá, Chile, in 2009 and obtained a Master's degree in Information Technology from Universidad Técnica Federico Santa María, Chile, in 2015. He is currently pursuing the Ph.D. degree in computer science at the Universidad Católica de Chile, PUC. He has been a professor of Computer Engineering at Universidad de Tarapacá since 2012. His research interests include Machine Learning, Deep Learning and Visual Navigation.



Fabrizio Alvarez-Alvarez is Computer Engineer graduated from Universidad de Tarapacá, Chile. He currently works as a computer engineer in a private certification company. His interests are Machine Learning, deep learning and data analysis.



Ximena Badilla-Torrico is a professor in the Engineering Faculty of Universidad de Tarapacá, she has a career in computer engineering and an MBA grade. Her interests include Information Systems, Software Engineering and Machine Learning.