

# Aprendizaje por Transferencia

J. E. Hernández, *Estudiante MCIA, UAQ*, C. G. Rodríguez, *Estudiante MCIA, UAQ*.  
Deep Learning

**Resumen**—El aprendizaje por transferencia tiene como objetivo mejorar el rendimiento de los métodos de aprendizaje mediante la transferencia de conocimiento contenido en modelos de aprendizaje, distintos, pero relacionados. De esa forma es posible reducir el tiempo de entrenamiento y la necesidad de bases de datos muy grandes para el entrenamiento de estos.

El aprendizaje por transferencia se ha vuelto una herramienta popular y prometedora en el campo de la inteligencia artificial, con una rápida expansión de conocimiento y constantes desarrollos.

En el presente trabajo se probaron 5 métodos distintos aplicados a un clasificador de perros y gatos, mediante aprendizaje por refuerzo, los cuales son Inception V3, VGG 16, Xception, Inception ResNet V2, ResNet-50 y EfficientNetB0.

**Temas claves**—Aprendizaje por transferencia, Machine Learning, Clasificación de imágenes, Keras, TensorFlow, ImageNet, API.

## I. INTRODUCCIÓN

IDEALMENTE, los métodos de aprendizaje automático requieren un contexto con un conjunto etiquetado y abundante de datos de entrenamiento, con una distribución similar a los datos de prueba. Sin embargo, en el caso real, es difícil conseguir un escenario cercano al ideal, ya sea por tiempo, por costo, o porque simplemente es poco realista. Esto presenta una limitación actual de los métodos de aprendizaje automático.

Con el aprendizaje semi-supervisado, es posible atenuar parte del problema al requerir un número limitado de datos etiquetados, y un gran número de datos no etiquetados para la mejora de aprendizaje. Sin embargo, así como en algunos escenarios la recopilación de grandes cantidades de datos etiquetados en un problema también llega a suceder con los datos no etiquetados.

Gracias a un concepto, proveniente de la psicología educativa, es posible dar solución a dicho problema. El aprendizaje por transferencia se centra en la transferencia del conocimiento entre dominios y es una herramienta actual que sirve de apoyo a los métodos de aprendizaje automático. El psicólogo C. H. Judd, en su teoría de la generalización de la transferencia, nos dice que aprender a transferir es el resultado de la generalización de la experiencia.

J. E. Hernández estudia la Maestría en Ciencias en Inteligencia Artificial en la Universidad Autónoma de Querétaro (e-mail: [zeedhermx@gmail.com](mailto:zeedhermx@gmail.com))

C. G. Rodríguez estudia la Maestría en Ciencias en Inteligencia Artificial en la Universidad Autónoma de Querétaro(e-mail: [gaby96.flores@gmail.com](mailto:gaby96.flores@gmail.com))

Así, es posible realizar una transferencia de conocimiento de un dominio a otro, siempre y cuando se generalice la experiencia a transferir, y apoyándose de una conexión existente entre los dos dominios.

Como un ejemplo sencillo, una persona que ha aprendido el violín puede aprender el piano más rápido que otros, pues tanto el violín como el piano son instrumentos musicales y pueden compartir conocimientos en común. Como caso contrario, de nada sirve a una persona el saber andar en bicicleta a aprender a tocar el violín, pues no existe una conexión de conocimiento entre ambos dominios. Así mismo, aunque los dominios sean similares, la conexión entre estos puede resultar engañosa, como el caso de una persona que sabe hablar español, pero la experiencia adquirida no le facilita el aprender francés, si no que puede dificultarlo (por ejemplo, debido a que la experiencia exitosa en el aprendizaje del español, podría dificultar el aprendizaje del francés respecto a la fonética, conjugación, etc), a pesar de que ambos tienen una estrecha relación entre sí y forman parte del conjunto de lenguas romances.

A este fenómeno, en que una experiencia previa tiene un efecto negativo en el aprendizaje se le conoce en la psicología como aprendizaje negativo, y es aplicado al aprendizaje automático, cuando la experiencia trasferida de un método a otro ocasiona un efecto negativo en el método objetivo. En la figura 1 se muestran algunos ejemplos sencillos sobre aprendizaje por transferencia.

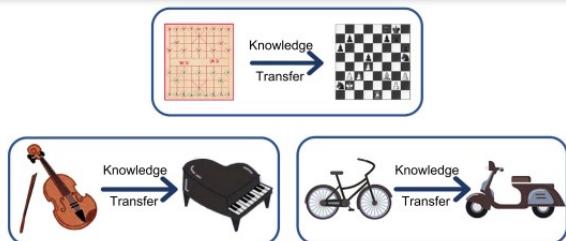


Fig. 1. Ejemplos sencillos sobre el aprendizaje por transferencia.

El estar de un lado o del otro de esta diferencia en el aprendizaje, depende en gran medida de la expertiza de la personal al decidir qué parte del dominio a transferir es importante para el dominio objetivo.

El aprendizaje por transferencia se divide en transferencia homogénea, y heterogénea. Una transferencia es homogénea cuando los dominios son del mismo espacio de características,

y heterogénea si los dominios tienen diferentes espacios de características.

En el presente trabajo se realiza un comparativo entre 5 métodos distintos a transferir, mediante aprendizaje por transferencia, y los resultados obtenidos al transferirlos a un modelo de clasificación de perros y gatos.

## II. DESARROLLO

Para realizar la implementación de las diferentes arquitecturas en la interfaz de Google Colab fue necesario primeramente habilitar la GPU, con el fin de acelerar el tiempo de ejecución del programa en cuestión.

De igual forma, fue necesario realizar la declaración de las librerías, en donde se importaron los modelos a utilizar mediante la instrucción que se muestra en la Figura 2; y el montado de Drive para su posterior utilización.

```
from tensorflow.keras.applications import InceptionV3, VGG16, Xception, InceptionResNetV2, EfficientNetB0, ResNet50
```

Fig. 2. Declaración de los modelos mediante la librería de keras.

Así bien, se reutilizaron algunas líneas de la tarea anterior para la realización de ciertas secciones del código como son: la sección de los datos, la cual se enfoca en descargar y descomprimir el archivo.zip; y la sección del pre procesamiento, en donde se realiza la lectura de las imágenes, la asignación de las variables: X y Y, la escritura y lectura de archivo HDF5, Normalización de las variables, así como la división del conjunto de datos por medio de la regla 60/20/20.

### A. Arquitecturas.

A continuación, se comentará brevemente sobre cada una de los modelos escogidos.

La red ‘Efficient Network’ (EfficientNet), es un nuevo modelo de red neuronal desarrollado por Google, el cual no solo se centra en mejorar la precisión del modelo, sino también en mejorar la eficiencia del mismo.

La red Inception es una arquitectura de red convolucional profunda propuesta por Google. Entre los modelos de Inception se encuentran: InceptionV3 y InceptionResNet V2.

Las redes neuronales residuales (ResNet) utilizando de conexiones de salto para moverse sobre varias capas. Así bien, permite entrenar cientos, si no miles de capas, consiguiendo un gran rendimiento

Las redes VGG siguen el esquema arquetípico de las redes convolucionales clásicas: una serie de convoluciones, max pooling, capas de activación y finalmente algunas capas de clasificación completamente conectadas.

La red Xception supera el rendimiento de Inception v3 en los dataset de ImageNet, y lo hace igualmente bien con los dataset para la clasificación de imágenes de 17,000 clases. Además, tiene la misma cantidad de parámetros del modelo Inception, lo que permite contar una eficiencia.

En la Tabla 1. se puede visualizar algunas de las características más importantes sobre las distintas arquitecturas a utilizar, como son: el tamaño, el número de parámetros, la profundidad del modelo, etc.

TABLA 1  
CARACTERÍSTICAS DE LOS DISTINTOS MÉTODOS [2]

Modelo	Tam. (MB)	Parámetros	Prof,	Tiempo (ms) por paso de inferencia (GPU)
Efficient NetB0	29	5.3 M	132	4.9
InceptionV3	92	23.9 M	189	6.9
Inception ResNet V2	215	55.9 M	449	10.0
ResNet50	98	25.6 M	107	4.6
VGG-16	528	138.4 M	16	4.2
Xception	88	22.9 M	81	8.1

Para una mejor visualización de cada una de las arquitecturas de los modelos mencionados anteriormente, ver el anexo 1.

### B. Ajuste Fino (Fine Tuning)

El ajuste fino es una herramienta ampliamente utilizada en aprendizaje por transferencia, el cual consiste en descongelar capas a selección del usuario del modelo transferido, con el objetivo de permitir que los pesos se adapten mejor al problema particular y se obtenga en la capa de salida mejores resultados. De esta manera se consigue una mejor integración entre el modelo base y el modelo propuesto para la clasificación deseada.

En la siguiente tabla se puede visualizar las capas activas e inactivas de cada uno de los modelos.

TABLA 2  
CAPAS ACTIVAS E INACTIVAS DE CADA MODELO

	Capas Inactivas	Capas Activas
EfficientNetB0	0:159	160:236
Inception V3	0:247	248:311
Inception ResNet V2	0:629	630:779
ResNet50	0:142	143:174
VGG 16	0:14	15:18
Xception	0:125	126:131

### C. Modelos propuestos

En la tabla 3. se puede observar los elementos que conforman la sección de las conexiones completas para cada uno de los modelos. Esta sección se encuentra conectada posterior a la arquitectura pre-entrenada.

TABLA 3

CAPAS AÑADIDAS POSTERIOR A LA TRANSFERENCIA

	Capas Añadidas
EfficientNetB0	BatchNormalization Flatten Dense(512, activation =’relu’) Dropout(0.5) Dense(256, activation =’relu’) Dense(2 activation =’softmax’)
Inception V3	BatchNormalization Flatten Dense(512, activation =’relu’) Dropout(0.5) Dense(256, activation =’relu’) Dense(2, activation =’softmax’)
Inception ResNet V2	GlobalAveragePooling2D() Dense(2 activation =’softmax’)
ResNet50	Flatten BatchNormalization Dense(2 activation =’softmax’)
VGG 16	Flatten Dense(512, activation =’relu’) Dropout(0.5) Dense(256, activation =’relu’) Desne(2 activation =’softmax’)
Xception	GlobalAveragePooling2D() Dense(2 activation =’softmax’)

### D. Hiperparámetros

En la tabla 4. se muestra la comparación de los diferentes hiperparámetros para cada uno de los modelos.

TABLA 4

COMPARATIVA DE HIPERPARÁMETROS EN LOS DISTINTOS MÉTODOS.

	LR	Opt.	Loss	Métrica	Batch size	Num. Épocas
Efficient NetB0	5e-5	Adagrad	Cross	Acc.	32	10
Inception V3	5e-5	Adagrad	Cross	Acc.	32	10
Inception ResNet V2	5e-5	Adam	Cross	Acc.	32	10
ResNet 50	5e-5	Adam	Cross	Acc.	32	10
VGG 16	5e-5	Adam	Cross	Acc.	32	10
Xception	5e-5	Adam	Cross	Acc.	32	10

### E. Resultados del entrenamiento y la evaluación.

En las siguientes figuras se puede visualizar los resultados obtenidos en la etapa de entrenamiento y la evaluación de cada uno de los modelos.

#### INCEPTION V3

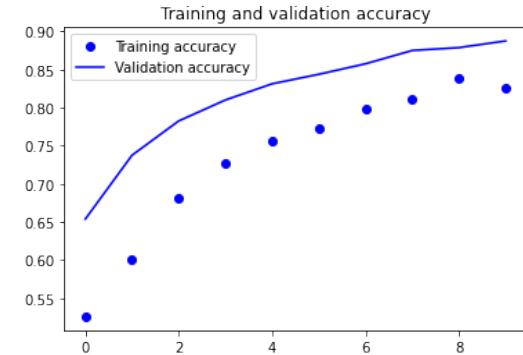


Fig. 3. Comparativa de precisión en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia Inception V3.

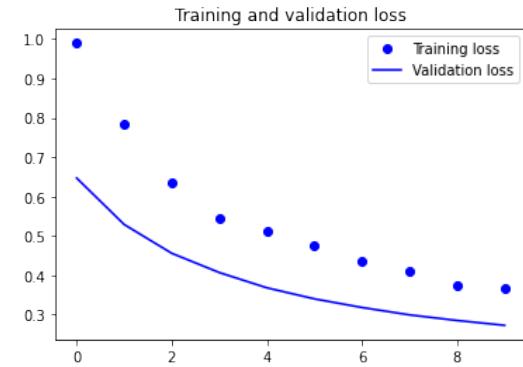


Fig. 4. Comparativa de perdida en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia Inception V3

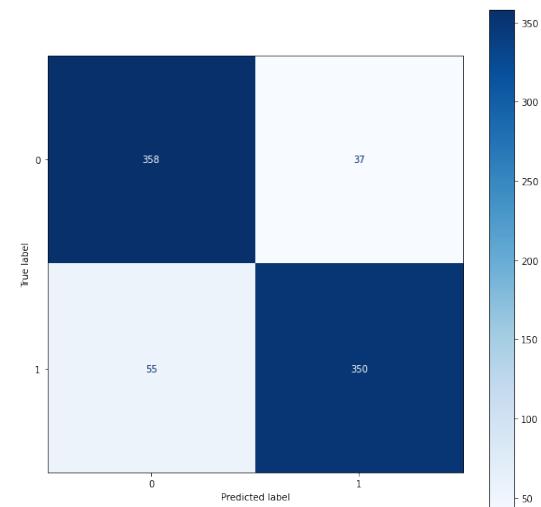


Fig. 5. Matriz de confusión del modelo propuesto con transferencia Inception V3.

TABLA 5

REPORTE DE RESULTADOS DEL MODELO PROUESTO CON TRANSFERENCIA  
INCEPTION V3

	Precisión	Recall	F1-Score	Support
0	0.87	0.91	0.89	395
1	0.90	0.86	0.88	405
Accuracy			0.89	800
Macro avg	0.89	0.89	0.88	800
Weighted av	0.89	0.89	0.88	800

VGG-16

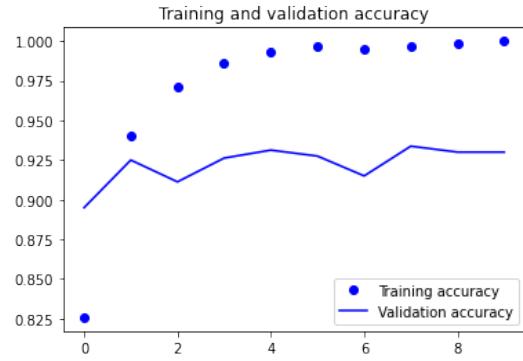


Fig. 6. Comparativa de precisión en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia VGG-16.

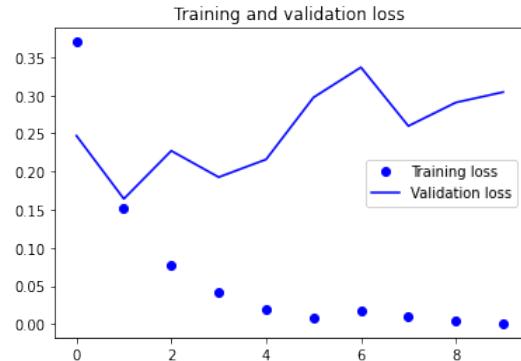


Fig. 7. Comparativa de perdida en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia VGG-16.

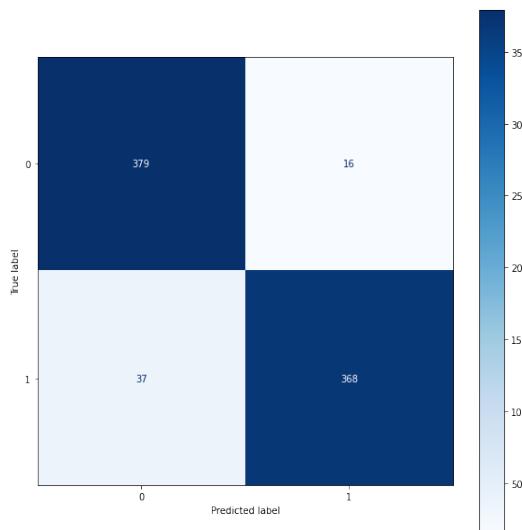


Fig. 8. Matriz de confusión del modelo propuesto con transferencia VGG-16.

TABLA 6

REPORTE DE RESULTADOS DEL MODELO PROUESTO CON TRANSFERENCIA  
VGG-16

	Precisión	Recall	F1-Score	Support
0	0.91	0.96	0.93	395
1	0.96	0.91	0.93	405
Accuracy			0.93	800
Macro avg	0.93	0.93	0.93	800
Weighted av	0.93	0.93	0.93	800

XCEPTION

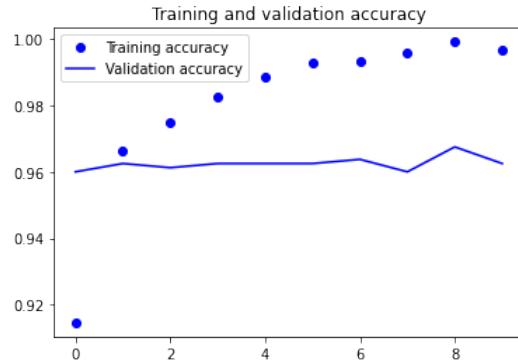


Fig. 9. Comparativa de precisión en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia Xception.

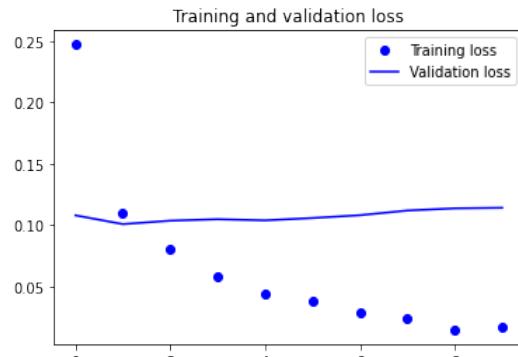


Fig. 10. Comparativa de perdida en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia Xception.

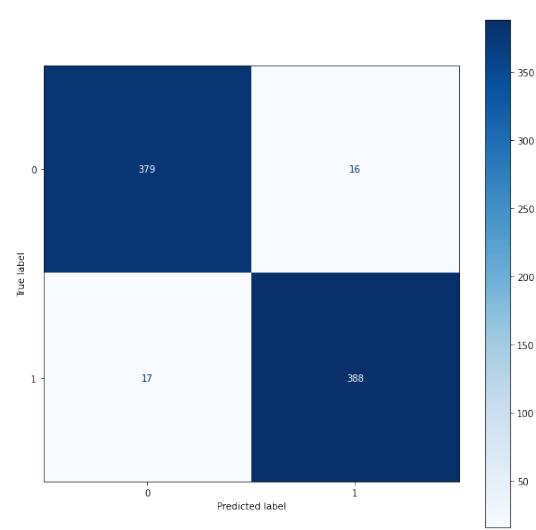


Fig. 11. Matriz de confusión del modelo propuesto con transferencia Xception.

**TABLA 7**  
REPORTE DE RESULTADOS DEL MODELO PROUESTO CON TRANSFERENCIA  
INCEPTION

	Precisión	Recall	F1-Score	Support
0	0.96	0.96	0.96	395
1	0.96	0.96	0.96	405
Accuracy			0.96	800
Macro avg	0.96	0.96	0.96	800
Weighted av	0.96	0.96	0.96	800

### INCEPTION RESNET V2

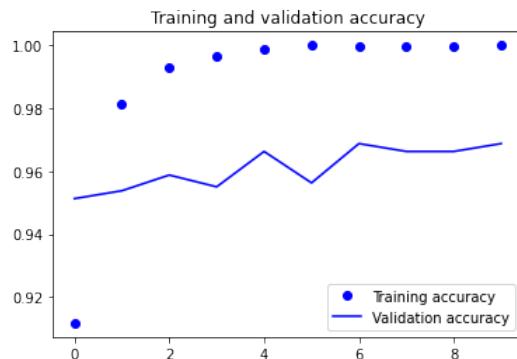


Fig. 12. Comparativa de precisión en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia Inception ResNetV2.

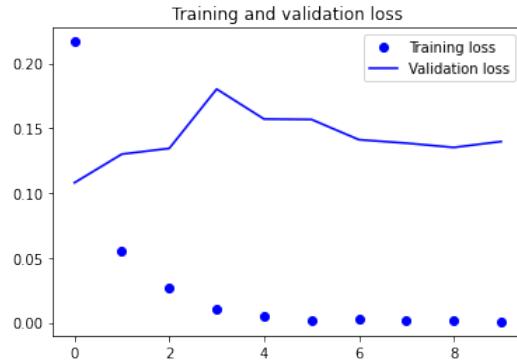


Fig. 13. Comparativa de perdida en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia Inception ResNetV2.

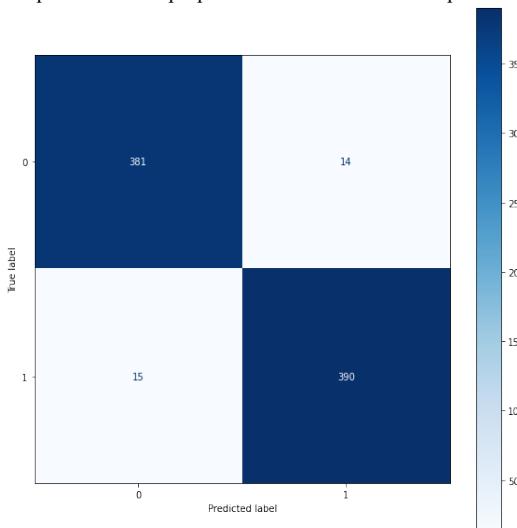


Fig. 14. Matriz de confusión del modelo propuesto con transferencia Inception ResNetV2.

**TABLA 8**  
REPORTE DE RESULTADOS DEL MODELO PROUESTO CON TRANSFERENCIA  
INCEPTION RESNET V2

	Precisión	Recall	F1-Score	Support
0	0.96	0.96	0.96	395
1	0.97	0.96	0.96	405
Accuracy			0.96	800
Macro avg	0.96	0.96	0.96	800
Weighted av	0.96	0.96	0.96	800

### RESNET-50

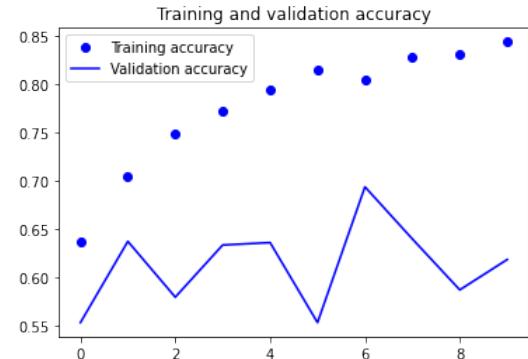


Fig. 15. Comparativa de precisión en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia ResNet-50.

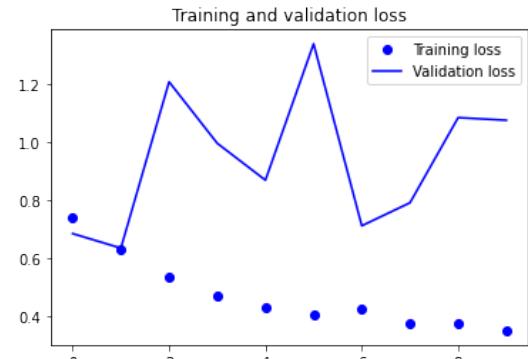


Fig. 16. Comparativa de perdida en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia ResNet-50.

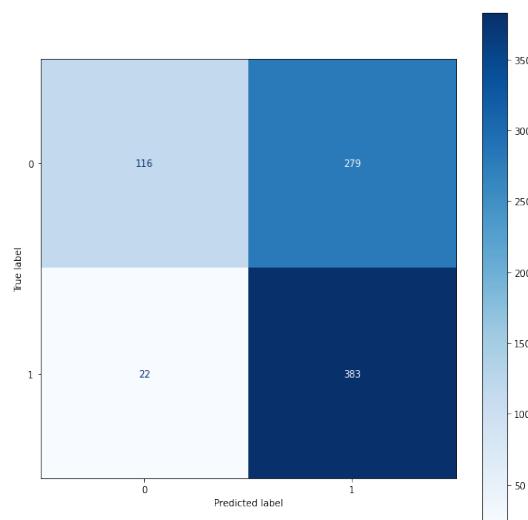


Fig. 17. Matriz de confusión del modelo propuesto con transferencia ResNet-50.

TABLA 9

REPORTE DE RESULTADOS DEL MODELO PROUESTO CON TRANSFERENCIA  
RESNET 50

	Precisión	Recall	F1-Score	Support
0	0.84	0.29	0.44	395
1	0.58	0.95	0.72	405
Accuracy			0.62	800
Macro avg	0.71	0.62	0.58	800
Weighted av	0.71	0.62	0.58	800

TABLA 10

REPORTE DE RESULTADOS DEL MODELO PROUESTO CON TRANSFERENCIA  
EFFICIENTNET B0

	Precisión	Recall	F1-Score	Support
0	0.49	1.00	0.66	395
1	0.00	0.00	0.00	405
Accuracy			0.49	800
Macro avg	0.25	0.50	0.33	800
Weighted av	0.24	0.49	0.33	800

## EFFICIENTNETB0

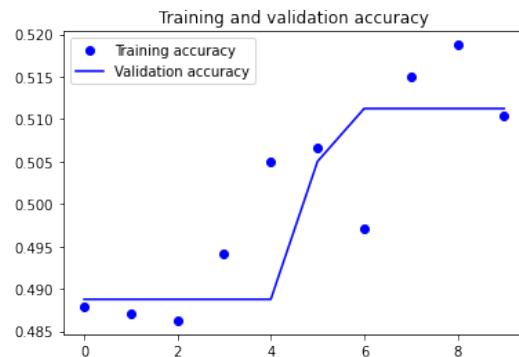


Fig. 18. Comparativa de precisión en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia EfficientNetB0.

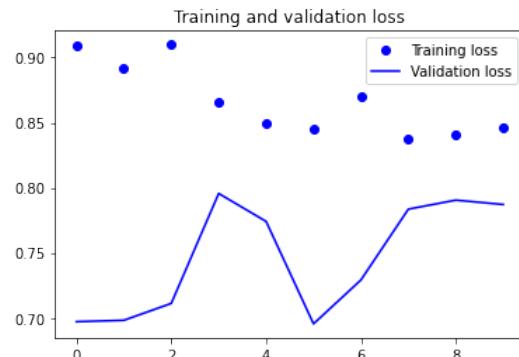


Fig. 19. Comparativa de perdida en grupo de datos de entrenamiento y validación para el modelo propuesto con transferencia EfficientNetB0.

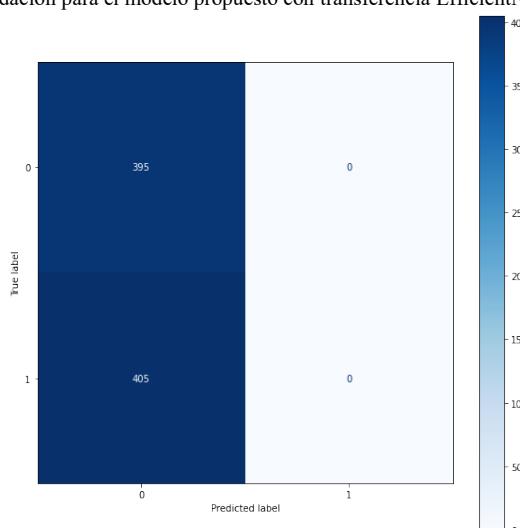


Fig. 20. Matriz de confusión del modelo propuesto con transferencia EfficientNetB0.

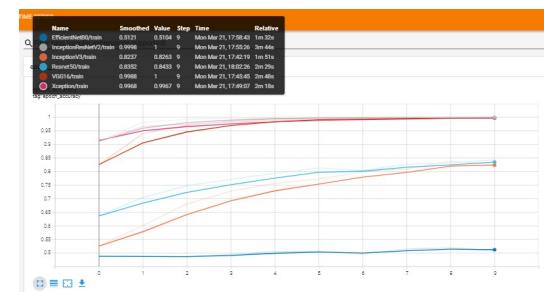


Fig. 21. Comparativa de precisión en grupo de datos de entrenamiento de los seis modelos propuestos.

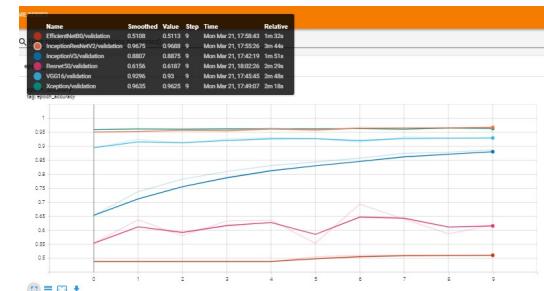


Fig. 22. Comparativa de precisión en grupo de datos de validación de los seis modelos propuestos.



Fig. 23. Comparativa de perdida en grupo de datos de entrenamiento de los seis modelos propuestos.

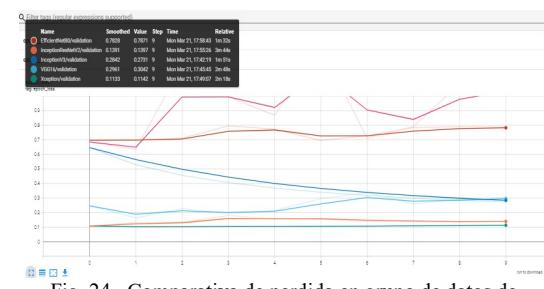


Fig. 24. Comparativa de perdida en grupo de datos de validación de los seis modelos propuestos.

#### F. Análisis de los resultados

La tabla 11, muestra los resultados obtenidos conforme a la evaluación de los modelos, en donde se puede realizar una comparación entre la precisión obtenida.

TABLA 11

COMPARATIVA DE PRECISIÓN OBTENIDA EN LOS DISTINTOS MÉTODOS.

Método	Precisión
EfficientNetB0	0.4937500059604645
Inception V3	0.8849999904632568
Inception ResNet V2	0.9637500047683716
ResNet50	0.6237499713897705
VGG 16	0.9337499737739563
Xception	0.9587500095367432

#### G. Ensamble

A continuación, se muestra el modelo del método ensamble, el cual se compone de las dos arquitecturas con los mejores resultados obtenidos anteriormente. El presente método consiste en devolver el promedio de la predicción de la combinación de los modelos.

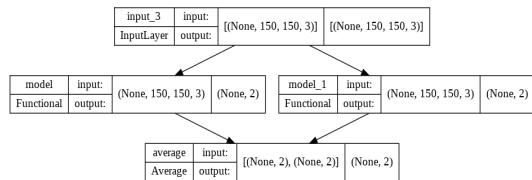


Fig. 25. Diagrama del modelo ensamble.

Así bien, en las siguientes figuras se puede visualizar los resultados obtenidos en la etapa de entrenamiento y la evaluación del modelo ensamble.

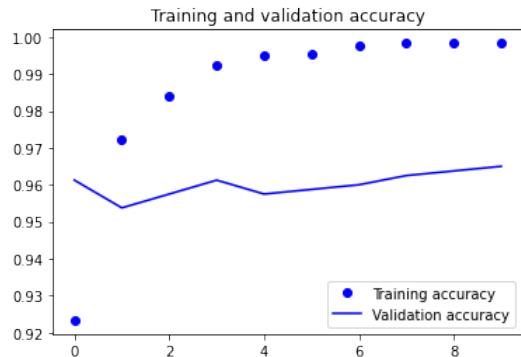


Fig. 26. Comparativa de precisión en grupo de datos de entrenamiento y validación para el modelo ensamble.

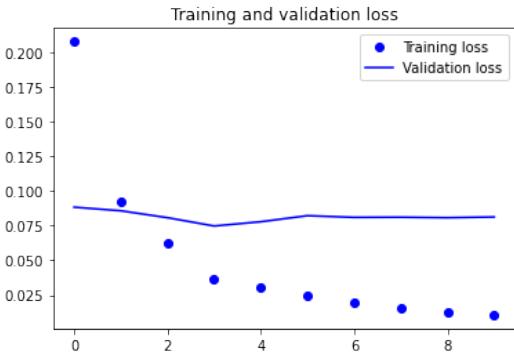


Fig. 27. Comparativa de perdida en grupo de datos de entrenamiento y validación para el modelo ensamble.

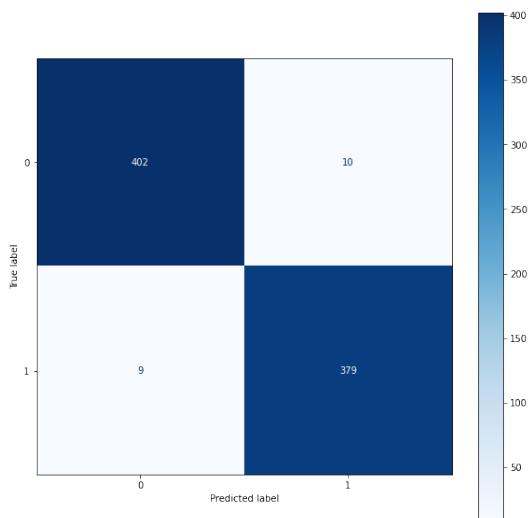


Fig. 28. Matriz de confusión del modelo propuesto ensamble.

TABLA 12

REPORTE DE RESULTADOS DEL MODELO PROPUESTO ENSAMBLE

	Precisión	Recall	F1-Score	Support
0	0.98	0.98	0.98	412
1	0.97	0.98	0.98	388
Accuracy			0.98	800
Macro avg	0.98	0.98	0.98	800
Weighted av	0.98	0.98	0.98	800

La tabla 13, muestra la comparación entre los resultados obtenidos conforme a la evaluación de los modelos, en donde se puede realizar una comparación entre la precisión obtenida.

TABLA 13  
COMPARATIVA DE PRECISIÓN OBTENIDA EN LOS DISTINTOS MÉTODOS.

Método	Precisión
EfficientNetB0	0.4937500059604645
Inception V3	0.8849999904632568
Inception ResNet V2	0.9637500047683716
ResNet50	0.6237499713897705
VGG 16	0.9337499737739563
Xception	0.9587500095367432
Ensamble	0.9762499928474426

## H. Conclusiones

La realización de esta tarea se enfocó en el reconocimiento de animales en específico de perros y gatos basado en imágenes por medio de la implementación del método de aprendizaje por transferencia, así como de la utilización del ajuste fino.

Para ello, se escogieron 6 modelos de redes neuronales convoluciones de la librería de Keras [2]: Inception V3, VGG-16, Xception, Inception ResNet V2, ResNet-50 y EfficientNetB0.

Los resultados obtenidos muestran que el enfoque de aprendizaje por transferencia puede conseguir una precisión de clasificación razonable a partir de una base de datos pequeña. Así bien, se pudo comprobar que no todas las redes pueden brindar resultados relativamente buenos, como fue el caso del modelo ResNet-50 y el modelo EfficientNetB0, en donde se obtuvieron precisiones menores 0.74.

Esto, debido posiblemente a que el modelo encuentra características que no son totalmente compatibles con el problema de clasificación de perros y gatos que buscamos resolver, lo que es prueba de que el éxito del aprendizaje por transferencia no solo depende de transferir un buen modelo, si no de saber identificar si las características de dicho modelo son compatibles con nuestro problema a solucionar.

Fue destacable la importancia del uso de GPU para el desarrollo del trabajo actual, pues reduce los tiempos de ejecución notablemente. Aún cuando gran parte del modelo transferido no fue reentrenado, lo que da prueba de la gran importancia de la técnica de aprendizaje por transferencia en las técnicas de aprendizaje automático.

Así bien, otro punto importante a destacar es el método de ensamble, el cual se compone de un grupo de elementos vistos como un todo y no desde una perspectiva individual. En general, el resultado obtenido es mejor que el resultado individual de cada modelo. Esto se puede comprobar por medio de la tabla 13, en donde se puede observar que se obtuvo una mejor precisión por medio del modelo ensamble.

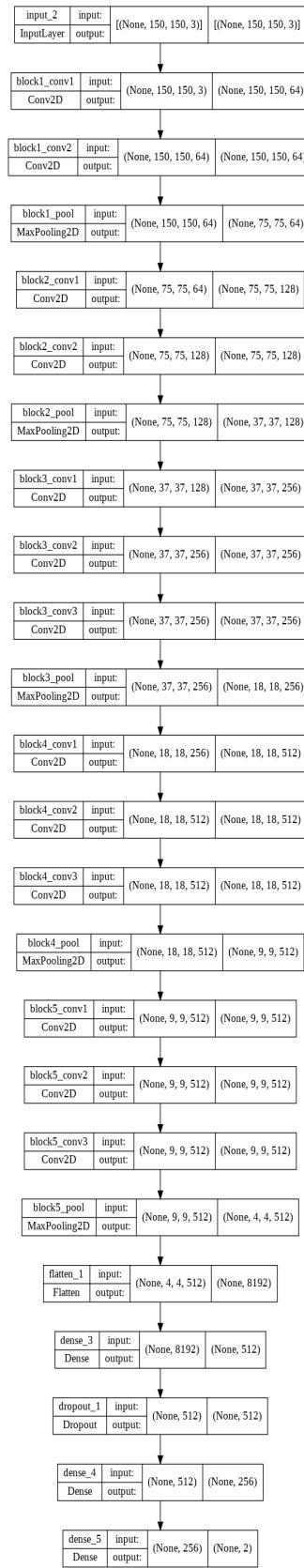
## III. REFERENCIAS

- [1] DataFlair.(s.f) Cats vs dogs Classification (with 98% Accuracy) using CNN Keras – Deep Learning Project for Begineers. Recuperado el día 19 de Marzo de 2022 de <https://data-flair.training/blogs/cats-dogs-classification-deep-learning-project-beginners/>
- [2] Keras (s.f) Keras Applications. Recuperado el día 17 de Marzo de 2022 de <https://keras.io/api/applications/inceptionresnet2/>
- [3] Acevedo, A. (2017) Transfer Learning:Keras Xception CNN. Recuperado el día 19 de Marzo de 2022 de <https://www.kaggle.com/abnera/transfer-learning-keras-xception-cnn>
- [4] Keras (s.f) Transfer learning & fine-tuning. Recuperado el día 17 de Marzo de 2022 de [https://keras.io/guides/transfer\\_learning/](https://keras.io/guides/transfer_learning/)
- [5] Brownlee J.(2019) Transfer Learning in Keras with Computer Vision Models. Recuperado el día 19 de Marzo de 2022 de <https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/>
- [6] Keras (s.f) EfficientNet B0 to B7. Recuperado el día 17 de Marzo de 2022 de <https://keras.io/api/applications/efficientnet/#efficientnetb7-function>
- [7] Rivera, M. (2019) Reuso de Redes Preentrenadas: Transferencia del conocimiento. Recuperado el día 17 de Marzo de [http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje\\_profundo/preentrenadas/preentrenadas.html](http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/preentrenadas/preentrenadas.html)
- [8] Keras (s.f) InceptionResNetV2. Recuperado el día 17 de Marzo de 2022 <https://keras.io/api/applications/inceptionresnetv2/>
- [9] Mwiti, D. (2022) Transfer Learning Guide: A Practical Tutorial with Examples for Images and Text in Keras. Recuperado el día 19 de Marzo de 2022 de <https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras>
- [10] Lendave, V.(2021) A comparison of 4 popular Transfer Learning Models. Recuperado el día 19 de Marzo de 2022 de <https://analyticsindiamag.com/a-comparison-of-4-popular-transfer-learning-models/>
- [11] Landup, D. (s.f) Image Classification with Transfer Learning in Keras-CREATE CUTTING EDGE CNN MODELS. Recuperado el día 19 de Marzo de 2022 de <https://stackabuse.com/image-classification-with-transfer-learning-in-keras-create-cutting-edge-cnn-models/>
- [12] TensorFlow (2019) Tensorboard in notebooks. Recuperado del día 19 de Marzo de 2022 de [https://colab.research.google.com/github/tensorflow/tensorboard/blob/master/docs/tensorboard\\_in\\_notebooks.ipynb#scrollTo=lpUO9HqUKP6z](https://colab.research.google.com/github/tensorflow/tensorboard/blob/master/docs/tensorboard_in_notebooks.ipynb#scrollTo=lpUO9HqUKP6z)
- [13] Keras (s.f) Optimizers. Recuperado 17 el día de Marzo de 2022 de <https://keras.io/api/optimizers/>.

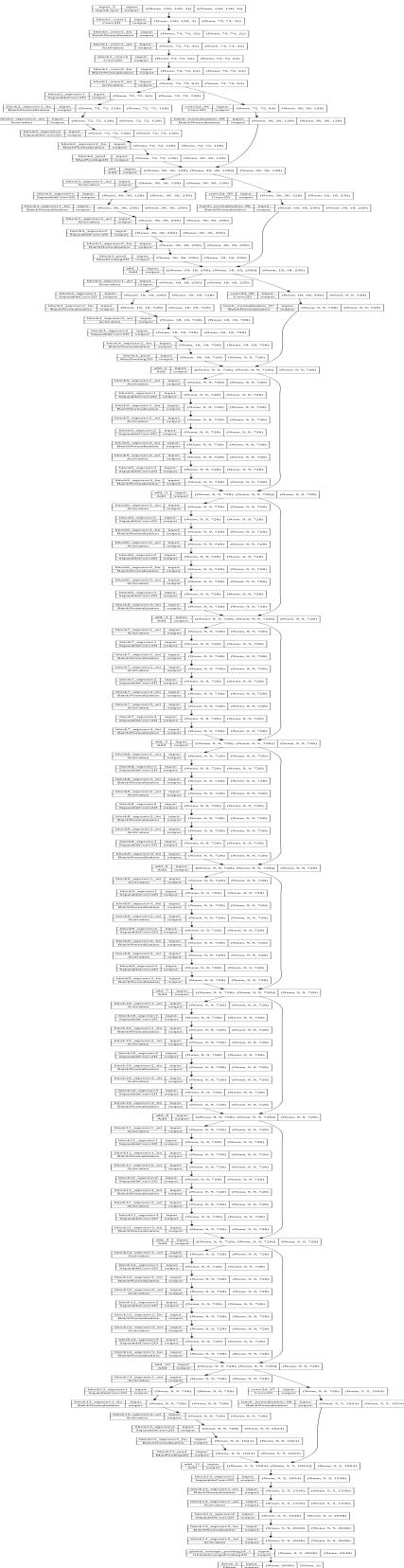
#### IV. ANEXO I

#### Modelo VGG-16

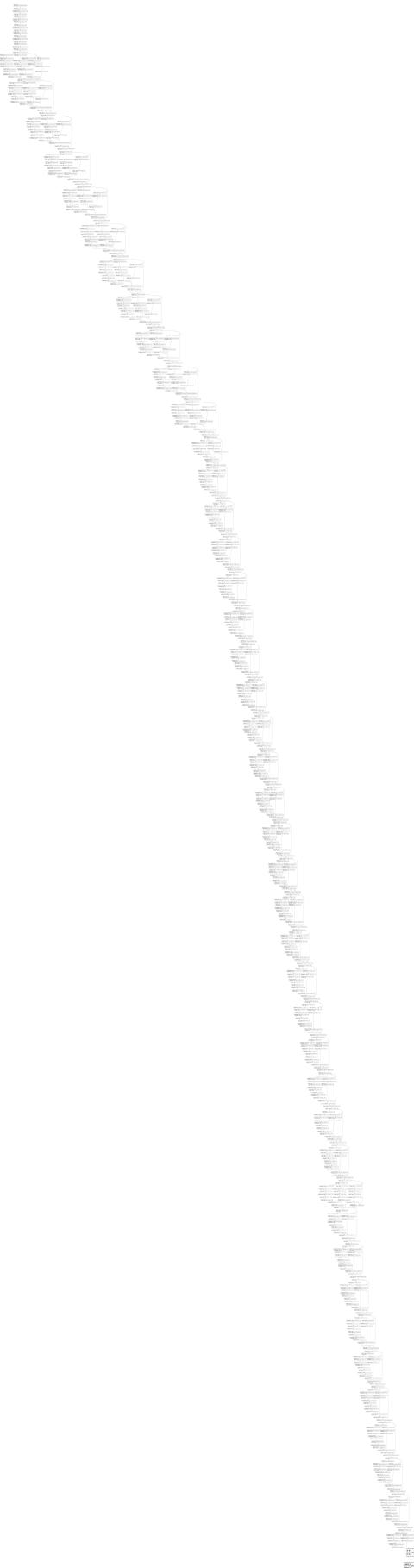
##### Modelo Inception V3



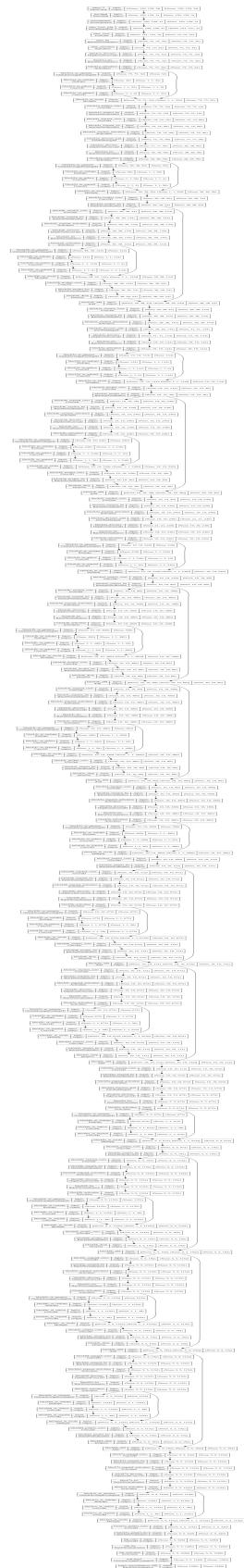
## Modelo Xception



## Modelo InceptionResNetV2



## Modelo EfficientNetB0



## Modelo ResNet-50

