

24GIIN – Inteligencia Artificial - UC3

Miguel Gagliardo¹

¹ Universidad Internacional de Valencia, Valencia 46002, España
mgagliardo@universidadviu.com

Abstract. En el siguiente artículo veremos cómo se ha entrenado un modelo para, dada la imagen de un animal, determinar si se trata de un gato, un perro u otro animal diferente.

Keywords: Inteligencia, Artificial, IA, Algoritmo

1 Datos

Para los datos de entrada, se utilizó el modelo de “Dogs vs Cats” de Kaggle¹, el mismo se puede descargar de forma gratuita desde el sitio web y contiene 25,000 archivos en formato **.jpg** de perros y gatos. Las fotos están etiquetadas por su nombre de archivo, con la palabra “dog” o “cat”. La convención de nomenclatura de los archivos es la siguiente:

```
cat.0.jpg
...
cat.124999.jpg
dog.0.jpg
dog.124999.jpg
```

Dado que en el dataset se encuentran miles de archivos, para entrenar el modelo solo se utilizaron 500 imágenes de gatos y de perros, respectivamente.

Ahora bien, como podemos ver el ejercicio solicita que nuestro modelo determine si es un gato, un perro **u otro animal**. El problema que esto conlleva es que los modelos de inteligencia artificial están entrenados para dar Outputs y, a la hora de predecir, le asignan una probabilidad a cada clase que tiene una suma de 1. Por tanto para este caso no tendremos solo 2 clases (perro y gato) si no una tercera, a la cual le asignaremos el label **“Otros”** y lo entrenaremos con imágenes azarosas de animales que no son perros ni gatos ^{2, 3, 4, 5} (*Figura 1*).

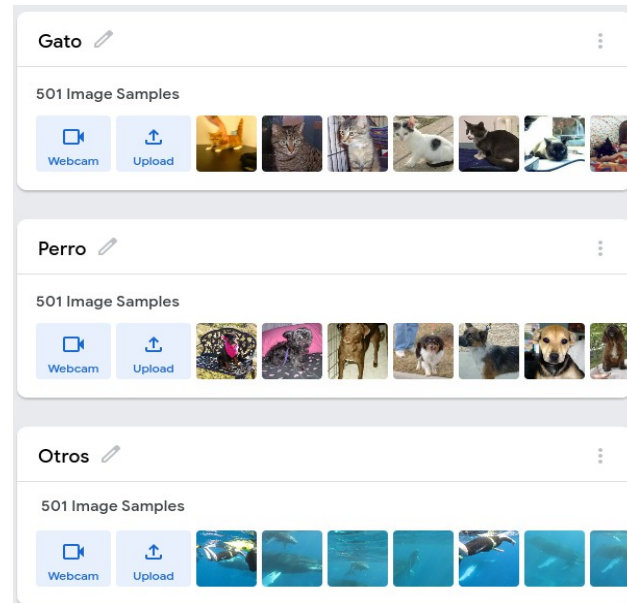


Figura1. Clases para las cuales el modelo fue entrenado

Este problema es también abordado en el paper *Unseen Class Discovery in Open-world Classification*⁶, el cual aborda el tema de la clasificación en un mundo abierto, donde un clasificador no solo clasifica ejemplos de prueba en **clases vistas durante el entrenamiento, sino que también rechaza ejemplos de clases no vistas o novedosas que no han aparecido durante el entrenamiento**. El enfoque se centra en descubrir clases ocultas no vistas dentro de los ejemplos rechazados. En el paper, los autores proponen un modelo conjunto de clasificación abierta con un submodelo para clasificar si un par de ejemplos pertenecen a las mismas o diferentes clases. Este submodelo sirve como una función de distancia para el agrupamiento con el objetivo de descubrir clases ocultas en los ejemplos rechazados. Los resultados experimentales sugieren que el modelo propuesto es altamente prometedor.

2 Organización y Preparación de los Datos

Tal y como se ha comentado, al descargar los datos de Kaggle, los mismos ya se encuentran organizados, para el caso de los perros todas las fotos están etiquetadas por su nombre de archivo, con la palabra “dog” o “cat” y un número, con lo cual el etiquetado ya está resuelto.

3 Creacion del Modelo

Al revisar los datos del dataset de perros y gatos, podemos ver que algunas fotos tienen formato horizontal, otras tienen formato vertical y otras son cuadradas. También se visualizan fotos donde apenas se ve al animal (*Figura 2.* en la esquina inferior izquierda) y otra que tiene dos gatos (*Figura 2.* esquina inferior derecha). Esto sugiere que cualquier clasificador que se ajuste a este problema tendrá que ser robusto.

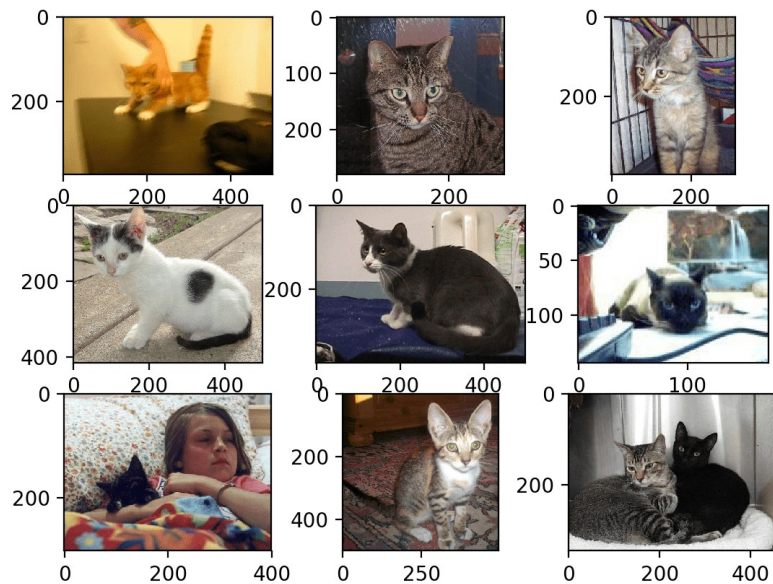


Figura 2. Ejemplos del dataset de gatos

Por tanto, las fotografías deberán remodelarse antes del modelado para que todas las imágenes tengan la misma forma. Hay muchas maneras de lograr esto, aunque la más común es una simple operación de cambio de tamaño que estirará y deformará la relación de aspecto de cada imagen y la forzará a adoptar la nueva forma.

Podríamos cargar todas las fotos y observar la distribución de los anchos y altos de las fotos, luego diseñar un nuevo tamaño de foto que refleje mejor lo que es más probable que veamos en la práctica. Las entradas más pequeñas significan un modelo que es

más rápido de entrenar. En este caso, seguiremos este ultimo enfoque y elegiremos un tamaño fijo de 200×200 píxeles.

Una vez fijado el tamaño de las 500 imagenes elegidas, se proceden a cargar en el modelo de Teachable creando 3 clases: Gato, Perro y Otros. En este ultimo cargaremos imagenes de animales que, como bien define su nombre, no son ni gatos ni perros (*Figura 1*).

4 Prueba del modelo

A la hora de probar el modelo tenemos 3 vias, o bien con imagenes de perros (*Figura 4*), o bien con imagenes de gatos (*Figura 5*). Para esto, tomamos imagenes del dataset que no hayan sido utilizadas para entrenar el modelo y las ubicamos en la UI de teachable. Como podemos ver, nuestro modelo reconoce con un 100% de confianza las imagenes ofrecidas.

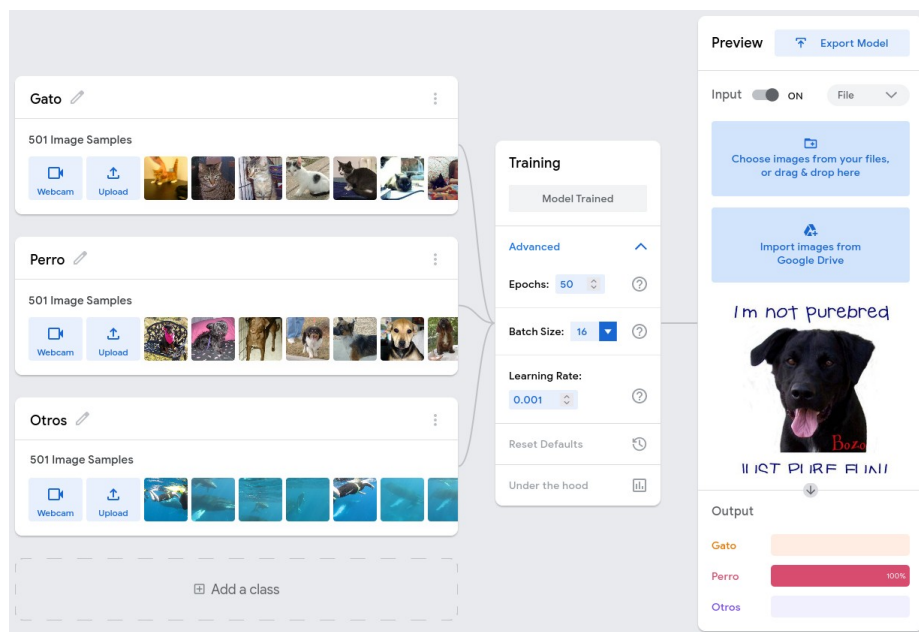


Figura 4. Prueba del modelo con la imagen de un perro

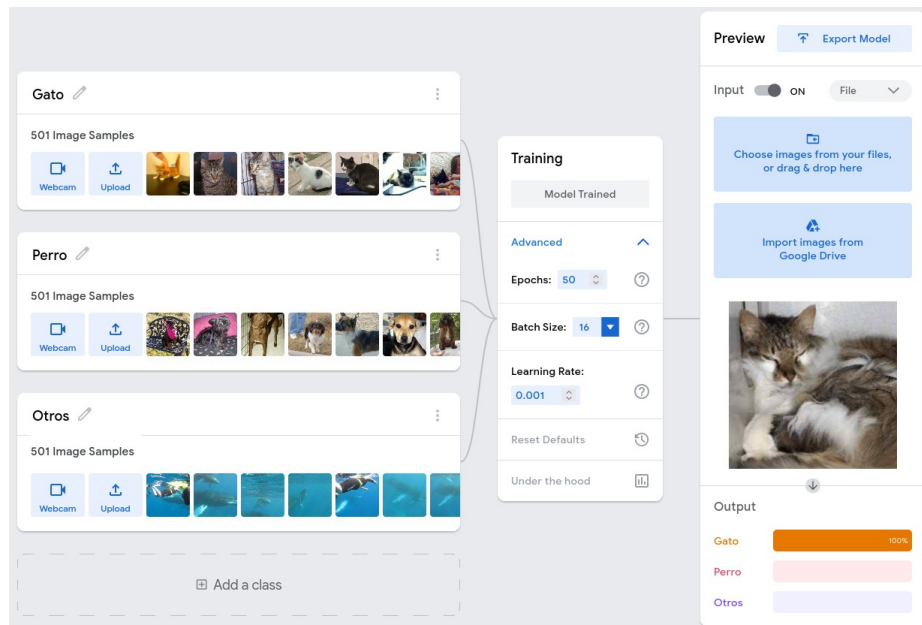


Figura 5. Prueba del modelo con la imagen de un gato

Ahora bien, para el caso de “**otros**” tenemos pruebas bastante sencillas como un mono (*Figura 6*) pero tambien tenemos casos de animales que son, o bien por linea genetica o bien por similitudes fenotipicas, muy similares a un perro o a un gato.

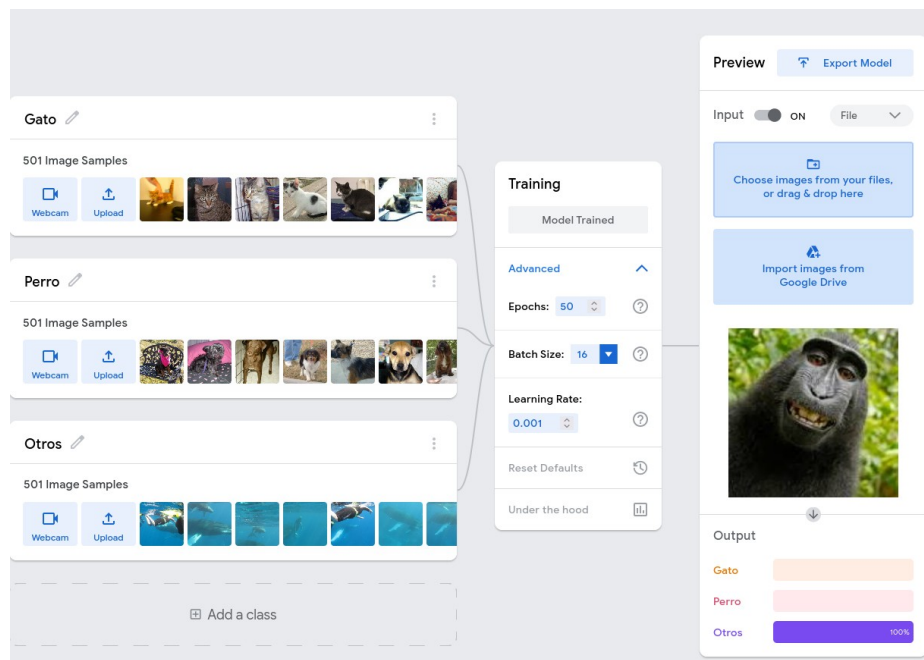


Figura 6. Probando el modelo con la imagen de un mono

Los casos mas claros de esto son el lobo (antecesor del perro) y el zorro, que tiene una linea muy similar a ambos por sus rasgos. . Para estos **edge cases** (*Figuras 7 y 8*), vemos que nuestro modelo no es tan bueno prediciendo estos casos, por lo cual podriamos reentrenarlo asignandole imagenes de lobos y zorros a la clase **otros**, pero nos encontramos con otro inconveniente mas, y es que existen razas de perros que por linea genetica son descendientes directos del lobo y comparten muchos rasgos fenotipicos con el mismo, siendo los casos mas iconicos el **Shiba Inu**, el **Akita Inu**, el **Husky**, el **Alaskan Malamud**, entre otras tantas, por lo cual tendriamos que reentrenar el modelo para dichas razas y aumentar su robustez.

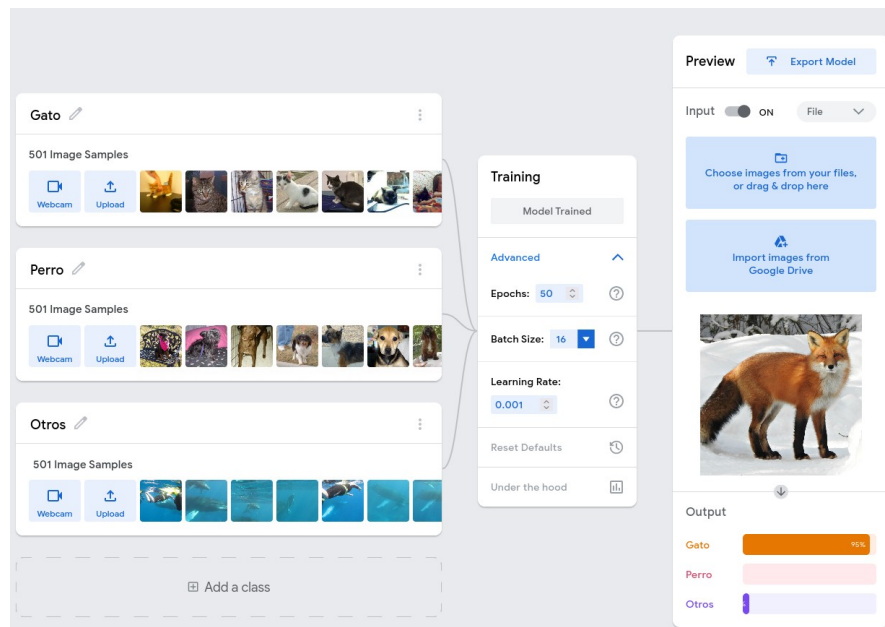


Figura 7. Probando el modelo con la imagen de un Zorro

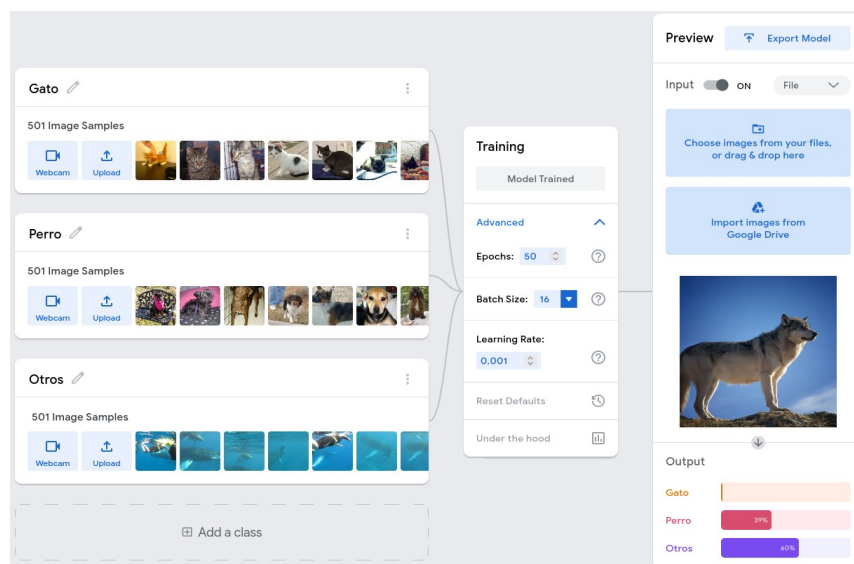


Figura 8. Probando el modelo con la imagen de un Lobo

5 Validacion del modelo

Validar un modelo de machine learning utilizando una matriz de confusión es una práctica común para evaluar su desempeño, especialmente en problemas de clasificación.

Para esto, hemos dividido el conjunto de datos en entrenamiento y prueba. Para el caso de entrenamiento hemos utilizado 1500 imagenes en total (500 perros, 500 gatos y 500 “otros” respectivamente).

Luego de entrenar el modelo con el dataset antes mencionado, se procede a realizar predicciones con 125 imagenes de cada clase o 375 en total para nuestro conjunto de prueba.

Una vez realizadas las predicciones, se procede a comparar los resultados reales con las predicciones y se crea la matriz de confusión. La matriz de confusión es una tabla que muestra las predicciones del modelo en comparación con los resultados reales.

En términos simples:

- **Verdadero Positivo (TP):** Instancias positivas correctamente clasificadas
- **Falso Negativo (FN):** Instancias positivas incorrectamente clasificadas.
- **Falso Positivo (FP):** Instancias negativas incorrectamente clasificadas.
- **Verdadero Negativo (TN):** Instancias negativas correctamente clasificadas.

Dado que estamos trabajando en un problema de clasificación multiclase, donde el modelo detecta si una imagen es de un perro, un gato o cualquier otro animal, hay varias métricas útiles a considerar para evaluar el rendimiento del modelo y construir una matriz de confusión. En nuestro caso hemos elegido:

- **Exactitud:** Mide la proporción de predicciones correctas respecto al total de predicciones.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** Mide la proporción de instancias positivas correctamente clasificadas respecto a todas las instancias clasificadas como positivas.

$$precision = \frac{TP}{TP + FP}$$

- **Recall (Sensibilidad):** Mide la proporción de instancias positivas correctamente clasificadas respecto a todas las instancias positivas reales

$$recall = \frac{TP}{TP + FN}$$

- **F1-Score:** Es la media armónica de precisión y recall, proporciona un equilibrio entre ambas métricas.

$$F1 - Score = 2 * \left(\frac{precision * recall}{precision + recall} \right)$$

- **Matriz de Confusion:** Representa visualmente la actuación del modelo. Muestra el número de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

Para nuestro conjunto de prueba de 375 imagenes, nuestros valores han sido:

Verdaderos Positivos (TP):

- Perro: 112
- Gato: 94
- Otro Animal: 56

Falsos Positivos (FP):

- Perro: 18
- Gato: 30
- Otro Animal: 6

Verdaderos Negativos (TN):

- Perro: 168
- Gato: 188
- Otro Animal: 301

Falsos Negativos (FN):

- Perro: 18
- Gato: 56
- Otro Animal: 9

Por tanto, nuestra matriz de confusión se ve así:

	Prediccion Perro	Prediccion Gato	Prediccion Otro Animal
Verdadero Perro	112	18	18
Verdadero Gato	30	94	56
Verdadero Otro Animal	6	6	301

Tabla 1. Resultados de la matriz de confusion

Ahora calculamos los datos con un posterior analisis de los mismos:

- **Accuracy (Exactitud)** ≈ 0.952
 - El modelo tiene una exactitud del 95.2%, lo que significa que el 95.2% de las clasificaciones son correctas en el conjunto de prueba de 375 imágenes.
- **Precision**
 - Para perros ≈ 0.862
 - Para gatos ≈ 0.758
 - Para otros animales ≈ 0.980
 - La precisión mide la proporción de instancias positivas correctamente clasificadas entre todas las instancias clasificadas como positivas. En nuestro caso, el modelo tiene una precisión alta para todas las clases, siendo particularmente alta para "Otro Animal" (98%).
- **Recall (Sensibilidad)**
 - Para perros ≈ 0.862
 - Para gatos ≈ 0.627
 - Para otros animales ≈ 0.971
 - El recall mide la proporción de instancias positivas correctamente clasificadas respecto a todas las instancias positivas reales. Nuestro modelo tiene un recall alto para todas las clases, siendo ligeramente más bajo para "Gatos" en comparación con las otras clases.

- **F1-Score:**
 - Para perros ≈ 0.862
 - Para gatos ≈ 0.686
 - Para otros animales ≈ 0.975
 - El F1-Score es una métrica que proporciona un equilibrio entre precisión y recall. Nuestro modelo tiene un F1-Score alto para todas las clases, indicando un buen equilibrio entre precisión y recall.

En resumen, el modelo parece tener un rendimiento sólido en la clasificación de animales, con altas precisiones, recalls y F1-Scores para todas las clases en el conjunto de prueba ampliado. **Sin embargo**, es importante considerar el contexto específico del problema y la importancia relativa de falsos positivos y falsos negativos según las necesidades del usuario o la aplicación.

6 Conclusion

La plataforma Teachable Machine de Google ofrece una interfaz intuitiva y accesible para adentrarse en el aprendizaje automático, eliminando la necesidad de habilidades de programación avanzadas. En la tarea específica de clasificación de animales, se destaca la importancia de la recolección y organización adecuada de datos. Aunque se sugiere un conjunto pequeño de imágenes (alrededor de 30 por clase), hemos experimentado con 500 por clase para evaluar su impacto en el rendimiento del modelo.

La fase de etiquetado de datos se presenta como un paso crítico, ya que la correcta asignación de etiquetas a cada clase es esencial para el entrenamiento y evaluación del modelo. La preparación de datos aborda la normalización de imágenes, la gestión de clases desequilibradas y la división efectiva entre conjuntos de entrenamiento y prueba, asegurando que el modelo pueda procesar y aprender de manera efectiva.

La creación del modelo se simplifica mediante la plataforma Teachable Machine y TensorFlow, destacando el enfoque supervisado y su aplicabilidad en la clasificación de imágenes de animales. Posteriormente, se enfatiza la importancia de probar y validar el modelo, utilizando técnicas como la matriz de confusión y métricas relevantes para evaluar su rendimiento en la clasificación de las tres clases: gato, perro y otro animal.

El análisis de resultados se basa en la interpretación de métricas como precisión, recall y F1-Score, proporcionando una comprensión más profunda de cómo el modelo realiza la clasificación en cada clase. Este análisis es crucial para ajustar y mejorar el modelo según las necesidades específicas del problema.

7 Referencias

1. **Dogs vs Cats - Kaggle:** <https://www.kaggle.com/c/dogs-vs-cats/data>
2. **Birds Species (tiny) Dataset - Kaggle:** <https://www.kaggle.com/datasets/sapal6/bird-speciestiny>
3. **Wild Animals Dataset - Kaggle:** <https://www.kaggle.com/datasets/vishweshsalodkar/wild-animals>
4. **Zoo Animals Extended Dataset - Kaggle:** <https://www.kaggle.com/datasets/agajorte/zoo-animals-extended-dataset>
5. **Sea Animals Dataset - Kaggle:** <https://www.kaggle.com/datasets/vencerlanz09/sea-animals-image-dataset>
6. **Unseen Class Discovery in Open-world Classification [2018] - Lei Shu, Hu Xu, Bing Liu:** <https://arxiv.org/pdf/1801.05609.pdf>
7. **24GIIN, Inteligencia Artificial:** Manual de la Asignatura, 2023