

Modelos de Machine Learning aplicados a la clasificación de imágenes de vehículos

J. Salas Flórez¹, M. Ramírez López¹, S. Suárez Ramírez¹

¹Universidad de Antioquia, departamento de ingeniería de sistemas

ABSTRACT En este documento se presenta el proceso de desarrollo y evaluación de algunos modelos de aprendizaje automático para la clasificación de imágenes de vehículos. La clasificación precisa de vehículos a partir de imágenes es una tarea fundamental en diversos campos, como lo son la seguridad vial y la gestión del tráfico. El enfoque propuesto utiliza técnicas de aprendizaje automático para extraer los modelos incluye la recolección de un conjunto de datos con imágenes de vehículos, las cuales se encuentran previamente etiquetadas, así como la limpieza y preparación de dichos datos, la elección de los algoritmos de aprendizaje automático más adecuados y el entrenamiento de los modelos. Se exploran diferentes enfoques, como redes neuronales convolucionales (CNN). Se realiza la evaluación de los modelos y se hace uso de métricas de desempeño. Los resultados muestran que los modelos de aprendizaje automático desarrollados logran un rendimiento satisfactorio en la clasificación de los datos, mostrando un potencial para su aplicación en situaciones del mundo real. Este estudio ofrece una base para futuras investigaciones en el campo del desarrollo de modelos de aprendizaje automático para la clasificación de imágenes de vehículos.

PALABRAS CLAVE Aprendizaje automático, clasificación de imágenes, reconocimiento de vehículos, redes neuronales artificiales, machine learning, entrenamiento, modelos, validación,

I. INTRODUCCIÓN

En los últimos años, los avances en el campo del aprendizaje automático, especialmente en técnicas de Deep learning, han permitido alcanzar resultados sobresalientes en tareas de clasificación de imágenes. Estos modelos son capaces de extraer características complejas y discriminativas de las imágenes, lo que ha impulsado el desarrollo de aplicaciones de clasificación en diversos dominios.

Sin embargo, la clasificación precisa de vehículos en imágenes aún presenta desafíos debido a la variabilidad en la apariencia de los vehículos, las diferentes perspectivas en las imágenes, los cambios de iluminación y la presencia de otros objetos en el entorno. Por lo tanto, se requiere un enfoque adecuado para abordar este problema y mejorar la precisión de la clasificación.

En este contexto, este artículo propone explorar y comparar varios modelos de machine learning aplicados a la clasificación de imágenes de vehículos. Se evaluarán diferentes enfoques, como redes neuronales convolucionales (CNN) y modelos pre-entrenados, con el

objetivo de determinar cual es el más eficiente y preciso para este problema específico.

II. EXPERIMENTOS

En esta sección se describe la metodología de validación utilizada y la base de datos empleada en el proyecto. También se proporcionan detalles sobre las clases, el número de muestras, la distribución de muestras y las técnicas de balanceo inteligente

A. METODOLOGÍA DE VALIDACIÓN

La metodología de validación utilizada es la validación cruzada estratificada. En este enfoque, se divide automáticamente el conjunto de datos en un conjunto de entrenamiento y un conjunto de validación. Al definir el generador de datos con *ImageDataGenerator* y establecer *validation_split=0.2*, se realiza una partición en la que el 80% de los datos se destina al conjunto de entrenamiento y el 20% al conjunto de validación. Esta estrategia garantiza que ambas divisiones mantengan una proporción equilibrada de muestras de cada clase, evitando así sesgos y asegurando una evaluación imparcial del modelo

B. BASE DE DATOS

La base de datos *TAU Vehicle Type Recognition* consiste en datos de entrenamiento con etiquetas de clase y datos de prueba sin etiquetas. Esta base de datos cuenta con un total de 17 clases diferentes, que son las siguientes: Ambulance, Boat, Cart, Limousine, Snowmobile, Truck, Barge, Bus, Caterpillar, Motorcycle, Tank, Van, Bicycle, Car, Helicopter, Segway y Taxi. En la siguiente figura se ve la distribución de muestras por cada clase

| Clase | Muestras | Clase | Muestras |
|------------|----------|-------------|----------|
| Barge | 202 | Caterpillar | 331 |
| Van | 1,111 | Snowmobile | 123 |
| Motorcycle | 2,986 | Car | 6,781 |
| Ambulance | 132 | Boat | 8,695 |
| Cart | 51 | Segway | 153 |
| Bus | 2,133 | Bicycle | 1,618 |
| Tank | 206 | Taxi | 748 |
| Truck | 2,033 | Limousine | 74 |
| Helicopter | 668 | | |

Figura 1: distribución de las clases

C. BALANCEO INTELIGENTE

Para abordar el desbalance en la distribución de muestras, se implementaron técnicas de submuestreo y sobremuestreo de manera inteligente. El enfoque consiste en utilizar exclusivamente muestras que se encuentren en el rango de 2000 a 2200 para cada clase, lo que permite trabajar con un conjunto de datos equilibrado. Para llegar al rango deseado se descartaron las clases que contaban con menos de 1000 muestras.

En primer lugar, se aplicó sobremuestreo a las clases 'Bicycle' y 'Van' con el propósito de aumentar las muestras. Se emplearon técnicas de rotación, zoom, distorsión y otras transformaciones para generar nuevas imágenes a partir de las muestras actuales. Dando este resultado:

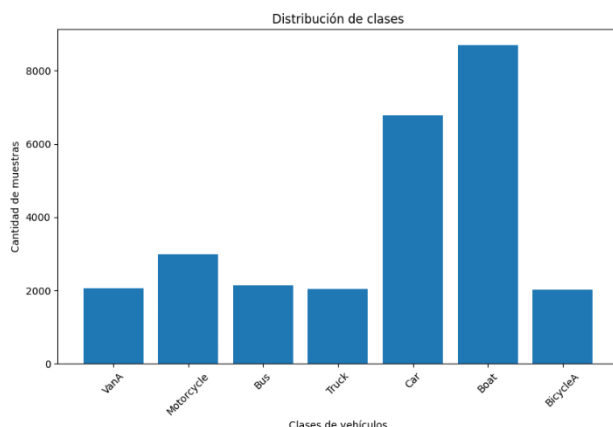


Figura 2: distribución después del sobremuestreo

Posteriormente, se llevó a cabo un submuestreo en las clases 'Boat', 'Car' y 'Motorcycle' con el fin de reducir la cantidad de imágenes. Mediante una eliminación aleatoria de muestras, se ajustó el número total de imágenes a 2200 para las clases seleccionadas. Este proceso permitió adecuar la distribución de las muestras y lograr un equilibrio en el conjunto de datos.

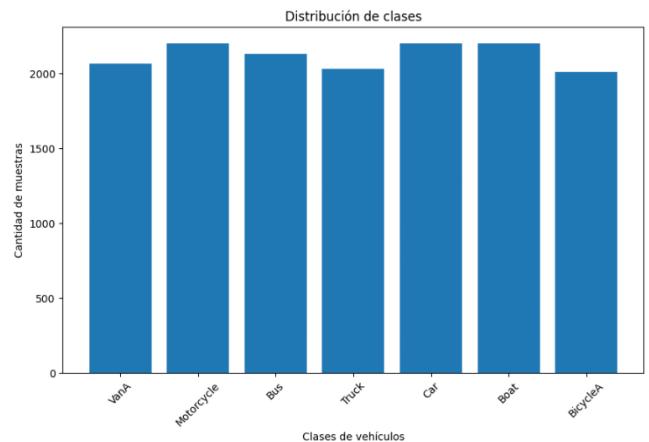


Figura 3: distribución después del submuestreo

La implementación inteligente de estas técnicas de submuestreo y sobremuestreo permite abordar de manera efectiva el desbalance en la distribución de muestras y obtener un conjunto de datos equilibrado para el proyecto. Esto brinda una base sólida para el entrenamiento y la evaluación de los modelos de reconocimiento de tipos de vehículos, con resultados más precisos y confiables.

III. MEDIDAS DE DESEMPEÑO

A. MATRIZ DE CONFUSIÓN

En problemas multiclase, la matriz de confusión es una representación tabular que muestra la relación entre las clases reales y las predicciones realizadas por el modelo para cada clase. A diferencia del caso binario, en un problema multiclase, la matriz de confusión tiene una estructura más compleja.

| | | Matriz de confusión: Modelo SVM | | | | | |
|--------|-----------|---------------------------------|----------|------|-----------|----------|-------|
| actual | ira | 37 | 0 | 1 | 2 | 5 | 0 |
| | tristeza | 0 | 59 | 3 | 0 | 0 | 4 |
| | asco | 0 | 3 | 47 | 4 | 4 | 0 |
| | felicidad | 2 | 0 | 3 | 45 | 3 | 0 |
| | sorpresa | 9 | 0 | 3 | 7 | 45 | 0 |
| | miedo | 0 | 1 | 1 | 1 | 0 | 48 |
| | | ira | tristeza | asco | felicidad | sorpresa | miedo |

Figura 4: ejemplo matriz de confusión

La matriz cuenta con dimensiones $n \times n$ donde n es el número de clases en el problema. Cada fila de la matriz representa la clase real, mientras que cada columna representa la clase predicha por el modelo.

Cada celda de la matriz de confusión contiene el recuento de instancias que pertenecen a la clase real correspondiente (fila) y que fueron clasificadas en la clase predicha correspondiente (columna). En el caso ideal, todos los valores estarían en la diagonal principal de la matriz, lo que indicaría una clasificación perfecta para todas las clases.

Sin embargo, en la práctica, puede haber errores de clasificación en forma de falsos positivos y falsos negativos para cada clase. Por lo tanto, la matriz de confusión multiclase permite analizar los aciertos y errores específicos para cada clase, lo que brinda información más detallada sobre el rendimiento del modelo en cada categoría.

B. CURVA ROC

La curva ROC (Receiver Operating Characteristic) es una herramienta gráfica utilizada para evaluar el rendimiento de un modelo de clasificación, especialmente en problemas de clasificación binaria. Sin embargo, también se puede aplicar a problemas de clasificación multiclase mediante la generación de curvas ROC para cada clase frente al resto.

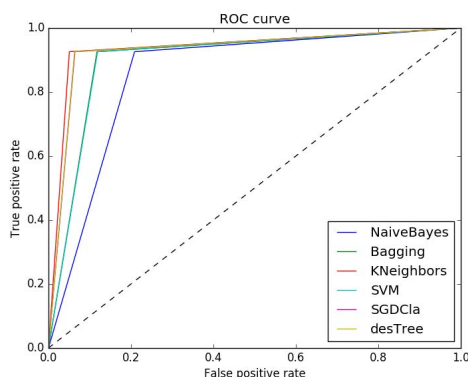


Figura 5: ejemplo curva ROC

IV. PREPARACIÓN DEL DATASET

Antes de iniciar con la creación del modelo, se deben preparar los datos de muestra para trabajar de una forma más cómoda, para esto, se muestran los datos actuales:



Figura 6: imágenes originales

Para garantizar un procesamiento eficiente, estabilidad del modelo y una mejor capacidad de generalización, se normalizarán las imágenes para que tengan el mismo tamaño y escala de colores RGB, dando como resultado un dataset normalizado:

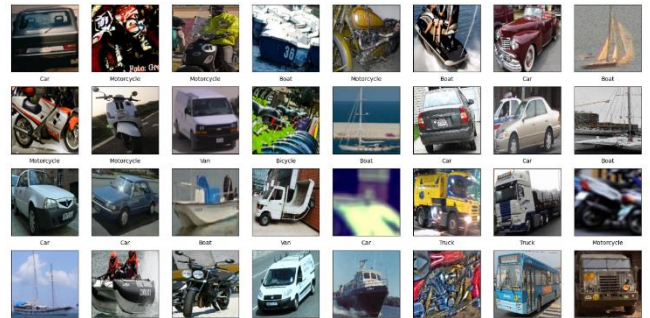


Figura 7: imágenes reescaladas

Este reescalado se realiza al mismo tiempo que la validación, es decir, el método que se encarga de reescalar las imágenes también se encarga de dividir las muestras en un conjunto de entrenamiento (80%) y un conjunto de validación (20%).

Realizado este proceso, el dataset está listo para ser utilizado en el modelo. Estas preparaciones son necesarias para garantizar que las imágenes se ajusten correctamente al modelo y pueden ser procesadas de manera eficiente durante el entrenamiento y validación.

V. DEFINICIÓN DE METRICAS EN CÓDIGO

Es necesario definir las métricas personalizadas para evaluar el desempeño de un modelo de aprendizaje automático. Estas métricas, como recall, precisión y puntuación F1, son ampliamente utilizadas en problemas de clasificación. Sirven para medir la capacidad del modelo de clasificar correctamente las muestras en distintas clases. Además, se incluye la función para calcular la matriz de confusión. Estas métricas resultan esenciales para comprender y evaluar la eficacia del modelo en la tarea de clasificación.

VI. MODELO: REDES NEURONALES ARTIFICIALES

Se define una red neuronal convolucional (CNN) para el aprendizaje automático. La red CNN se compone de múltiples capas convolucionales que extraen características de las imágenes de entrada y reducen su dimensionalidad. Luego, las capas completamente conectadas procesan las características extraídas y realizan la clasificación final. El modelo utiliza funciones de activación como ReLU y softmax para introducir no linealidad y calcular las probabilidades de clasificación. Además, se emplean técnicas como dropout y batch normalization para mejorar el rendimiento y la generalización del modelo. El modelo se compila utilizando la función de pérdida de entropía cruzada categórica y el optimizador Nadam. Por último, las métricas de evaluación utilizadas incluyen precisión, recall, puntuación F1 y matriz de confusión para evaluar el desempeño del modelo en la clasificación.

VII. ENTRENAMIENTO DEL MODELO

El modelo se entrena en un conjunto de datos de entrenamiento durante un número determinado de épocas. Durante el entrenamiento, se utilizan los generadores de datos train y valida para proporcionar los lotes de datos de entrenamiento y validación respectivamente, se establece el número de pasos por época de la mitad del tamaño de los generadores y por último se define la utilización de 30 épocas, lo suficiente para alcanzar una accuracy de 0.7

| E | Pérdida | Acierto | F1 | Precisión | Recall |
|----|---------|---------|--------|-----------|--------|
| 1 | 1.4213 | 0.4752 | 2.3965 | 4.8390 | 1.6303 |
| 2 | 1.1793 | 0.5776 | 1.6657 | 1.9273 | 1.4769 |
| 3 | 0.9621 | 0.6713 | 1.4288 | 1.5380 | 1.3396 |
| 4 | 0.8786 | 0.6970 | 1.6925 | 1.9054 | 1.5295 |
| 5 | 1.3697 | 0.4987 | 1.6161 | 1.8875 | 1.4219 |
| 6 | 0.9213 | 0.7070 | 1.3220 | 1.3910 | 1.2653 |
| 7 | 0.7974 | 0.7431 | 1.3963 | 1.4544 | 1.3465 |
| 8 | 0.8596 | 0.7268 | 1.2631 | 1.2914 | 1.2398 |
| 9 | 0.7515 | 0.7469 | 1.3592 | 1.4267 | 1.3024 |
| 10 | 0.7790 | 0.7441 | 1.3103 | 1.3380 | 1.2877 |
| 11 | 0.8374 | 0.7412 | 1.2258 | 1.2351 | 1.2193 |
| 12 | 0.8784 | 0.7487 | 1.2122 | 1.2169 | 1.2104 |
| 13 | 0.8567 | 0.7406 | 1.2534 | 1.2918 | 1.2214 |
| 14 | 0.7422 | 0.7692 | 1.2237 | 1.2414 | 1.2098 |
| 15 | 0.9340 | 0.7662 | 1.2516 | 1.2818 | 1.2267 |
| 16 | 0.8972 | 0.7881 | 1.1240 | 1.1128 | 1.1373 |
| 17 | 1.0167 | 0.7682 | 1.1654 | 1.1529 | 1.1804 |
| 18 | 0.8427 | 0.7931 | 1.1701 | 1.1589 | 1.1835 |
| 19 | 0.9551 | 0.7719 | 1.1631 | 1.1509 | 1.1779 |
| 20 | 0.7316 | 0.7676 | 1.3170 | 1.3496 | 1.2900 |

Figura 8: resultados de entrenamiento del modelo

VIII. VALIDACIÓN DEL ENTRENAMIENTO

Antes de mostrar las métricas de validación de desempeño del modelo, es necesario mostrar las pruebas realizadas por el modelo, que permiten ver los resultados obtenidos, para esto se define un código que permita tomar una cantidad de imágenes aleatorias del dataset, luego utilizar el modelo de CNN para predecir la clase de la muestra tomada aleatoriamente, lo que da un resultado visible de la siguiente forma:

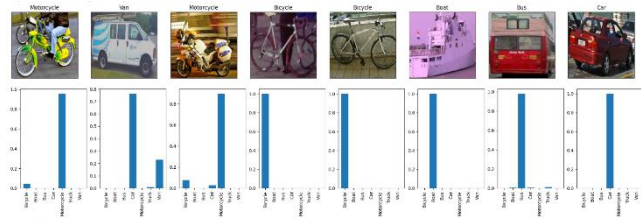


Figura 9: validación del entrenamiento

En esta distribución de imágenes se muestra como fue utilizado el modelo para predecir la clase, en la parte superior de la figura se muestra la imagen con su clase real, y en la

parte inferior se muestra una gráfica con la totalidad de clases y la predicción del modelo.

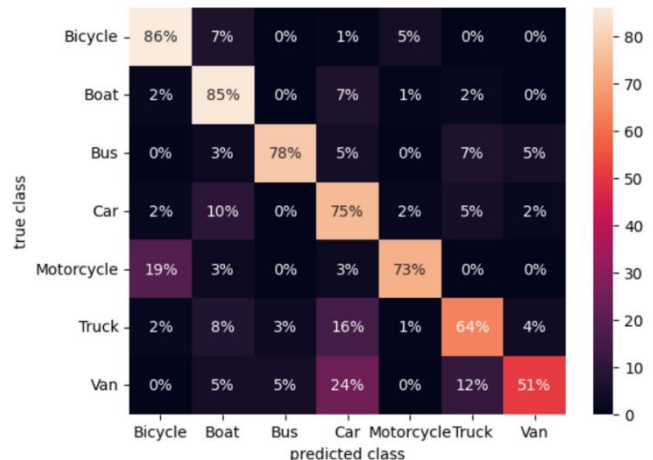
En general el modelo predice satisfactoriamente las imágenes que ingresan, con el porcentaje asignado por el modelo a cada clase, en algunas muestras se evidencia una predicción correcta de casi 100%, en otras vemos que este valor disminuye por las características parecidas, pero lo hace sin equivocarse la mayoría de las veces.

IX. VALIDACIÓN DE METRICAS

Ahora, se utilizan secciones de código que permiten mostrar gráficamente los resultados del modelo después de entrenar y validar, para esto, se utilizó la matriz de confusión y la curva ROC

A. MATRIZ DE CONFUSIÓN

Para mostrar la matriz de confusión se utiliza la biblioteca 'seaborn' para crear el mapa de calor. Esta representación visual permite una mejor comprensión de cómo el modelo está clasificando las muestras en cada clase y puede ayudar a



identificar áreas de mejora en el rendimiento del modelo:

Figura 10: matriz de confusión del modelo

B. CURVA ROC

En el caso del problema multiclase del proyecto, se utiliza una técnica de binarización de etiquetas para calcular las curvas ROC y el área bajo la curva (AUC) para cada clase. Se utiliza 'scikit-learn' para realizar esta tarea. La puntuación de ROC AUC es una medida de la capacidad del modelo para distinguir correctamente entre las clases. Además, oscila entre 0 y 1, siendo 1 un rendimiento perfecto y 0.5 un rendimiento aleatorio o equivalente a adivinar al azar. Una puntuación más alta indica un mejor rendimiento. Para esta figura, el ROC AUC score es: 0.8464515246198154. Este resultado indica que el modelo tiene una alta capacidad para distinguir entre las clases y realizar predicciones precisas.

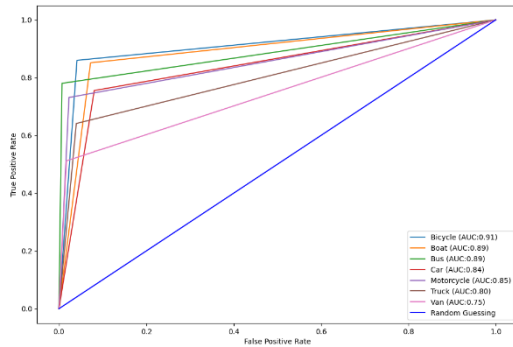


Figura 11: curva ROC del modelo

X. ANALISIS DE RESULTADOS

El modelo de aprendizaje automático desarrollado para clasificar imágenes de vehículos ha logrado una precisión del 76% en la tarea de identificación. Este resultado demuestra una capacidad prometedora para reconocer y categorizar imágenes relacionadas con vehículos teniendo en cuenta que dentro del conjunto de imágenes existen algunas con poca calidad o con otros objetos que pueden dificultar la tarea de clasificación. Sin embargo, es importante tener en cuenta que existe un margen de mejora para alcanzar una precisión aun mayor. Se pueden explorar diversas estrategias para optimizar el modelo, como la recopilación de un conjunto de datos más amplio y diverso, el ajuste de los hiperparámetros del algoritmo o la implementación de técnicas avanzadas de preprocesamiento de imágenes. Estos enfoques pueden ayudar a abordar los desafíos y las limitaciones actuales, permitiendo así mejorar la precisión y la robustez general del modelo.

Tras analizar la matriz de confusión, se han identificado algunos desafíos, principalmente en la categoría de 'Motorcycle'. Se observó que el 19% de las muestras de esta clase fueron erróneamente clasificadas como 'Bicycle'. Además, se encontraron dificultades en la categoría de "Truck", donde aproximadamente el 16% de las muestras fueron clasificadas incorrectamente como "Car". Por último, la clase más problemática para nuestro modelo resultó ser la de "Van", ya que el 24% de las imágenes se clasificaron incorrectamente como "Car".

Estos errores de clasificación destacan la necesidad de abordar y mejorar la capacidad del modelo para distinguir con precisión entre estas categorías específicas de vehículos. Sin embargo, los errores son comprensibles debido a la calidad de algunas de las imágenes y a que la clasificación de algunos vehículos en estas clases puede resultar ambiguo solo con la información que se puede extraer de la imagen. En este sentido se hace importante realizar un análisis más detallado de las características distintivas y los patrones visuales presentes en las imágenes de motocicletas, camiones y furgonetas para lograr una mayor precisión en su clasificación. Asimismo, es recomendable ampliar y diversificar el conjunto de datos de entrenamiento, especialmente en estas categorías problemáticas, con el fin de

proporcionar al modelo una mayor exposición a casos difíciles y garantizar una mejor generalización en su capacidad de clasificación.

XI. COMPARACIÓN CON LA ENTREGA 1

Al igual que en los artículos revisados, este estudio también se basó en el uso de un modelo de redes neuronales convolucionales. A través de este enfoque, se exploró la capacidad del modelo para capturar características espaciales relevantes en los datos de entrada y realizar tareas específicas. Siguiendo las consideraciones necesarias y evaluando las arquitecturas más adecuadas, nuestro modelo fue entrenado y evaluado utilizando conjuntos de datos representativos.

A diferencia de muchos de los artículos que revisamos, en lugar de utilizar modelos preexistentes con técnicas como transfer learning, decidimos construir nuestro propio modelo desde cero. Esta elección nos otorgó un mayor control sobre el desarrollo del modelo, permitiéndonos adaptarlo de manera específica a nuestra tarea y conjunto de datos. Sin embargo, esta decisión también implicó una mayor complejidad y requerimientos de tiempo en el proceso de entrenamiento. Pasamos por un proceso de entrenamiento y ajuste cuidadoso para obtener un modelo con resultados comparables a los obtenidos en los modelos de referencia. Esta elección nos dio una mejor comprensión interna del modelo y la capacidad de personalizarlo para satisfacer nuestras necesidades.

Después de realizar la evaluación de nuestro modelo, encontramos que su precisión se situó en el rango de los modelos tratados en los artículos revisados para la entrega 1 (entre 75 y 99%), evidenciando una precisión alrededor del 76%. Estos resultados demuestran la viabilidad y efectividad de nuestro modelo de redes neuronales convolucionales en la tarea específica abordada. Si bien existen oportunidades para mejorar aún más la precisión, estos resultados iniciales proporcionan una base sólida para futuros trabajos y refinamientos en nuestra investigación.

XII. REFERENCIAS

- [1] Simplilearn, "What is a confusion matrix in machine learning?," Simplilearn.com, <https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning> (accessed May 29, 2023).
- [2] K. Markham, "Simple guide to confusion matrix terminology," Data School, <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/> (accessed May 29, 2023).
- [3] Z. H. Hoo, J. Candlish, and D. Teare, "What is an ROC curve?," *Emergency Medicine Journal*, vol. 34, no. 6, pp. 357–359, 2017. doi:10.1136/emmermed-2017-206735

XIII. LINK DEL REPOSITORIO

<https://github.com/estebansalas18/clasificador-imagenes>