



VISIÓN ARTIFICIAL | CURSO 2024-2025

Trabajo práctico

Clasificador Cat vs Dog

Martín Orenes Peñas
Martín Sotelo Aguirre
sotelo.153910
orenes.153979

28 de diciembre de 2024

Índice general

1	Introducción	3
2	Preprocesamiento	4
3	Extracción de Características	5
3.1	Histograma de Gradientes Orientados (HOG)	5
3.1.1	¿Por qué hemos usado HOG?:	5
3.2	Detección de Puntos Singulares (Harris, SIFT y ORB)	5
3.3	Combinación de HOG y Puntos Singulares	6
3.4	Normalización y Reducción de Dimensionalidad	6
3.5	Impacto en la Clasificación	6
4	Interpretación de los Resultados	7
4.1	¿Por qué Random Forest es el mejor clasificador?	7
4.2	¿Por qué SVM tiene un rendimiento más bajo con HOG?	7
4.3	Análisis de las imágenes mal clasificadas	8
4.4	Resultados	9
5	Conclusión del Trabajo: Clasificador Cat vs Dog	10

1. Introducción

La clasificación de imágenes es una tarea clave en visión por computador, y uno de los problemas más conocidos para aprender esta técnica es la distinción entre imágenes de gatos y perros. Aunque hoy en día los modelos basados en deep learning logran los mejores resultados en este tipo de tareas, antes de su aparición se utilizaban métodos más tradicionales, que todavía son relevantes en ciertos contextos. En este trabajo, exploramos estas técnicas clásicas para abordar el problema de clasificación de gatos y perros.

El enfoque tradicional de clasificación de imágenes consta de dos fases principales: la extracción de características y el entrenamiento del modelo. En la primera fase, transformamos las imágenes originales en vectores de características que capturen información importante sobre su contenido. En la segunda, usamos estos vectores para entrenar modelos clásicos, como árboles de decisión o máquinas de soporte vectorial, que son capaces de asignar cada imagen a una clase.

En este proyecto, implementamos un pipeline completo que incluye desde el preprocesamiento de las imágenes hasta la clasificación. Utilizamos técnicas clásicas de extracción de características como HOG (Histograma de Gradientes Orientados) y algoritmos de detección de puntos clave como Harris, SIFT y ORB. También combinamos estas características con clasificadores tradicionales, como Random Forest, SVM y AdaBoost (vistos en Aprendizaje Automático), para evaluar cuál de estas combinaciones produce los mejores resultados.

El objetivo principal es entender cómo afectan las diferentes técnicas de preprocesamiento, extracción de características y clasificación al rendimiento del sistema. Además, buscamos reflexionar sobre las ventajas y limitaciones de estas aproximaciones clásicas en comparación con los métodos más avanzados de deep learning. A lo largo de este trabajo, presentamos el diseño del pipeline, los experimentos realizados y los resultados obtenidos, junto con un análisis de cómo cada técnica contribuye al rendimiento final del modelo.

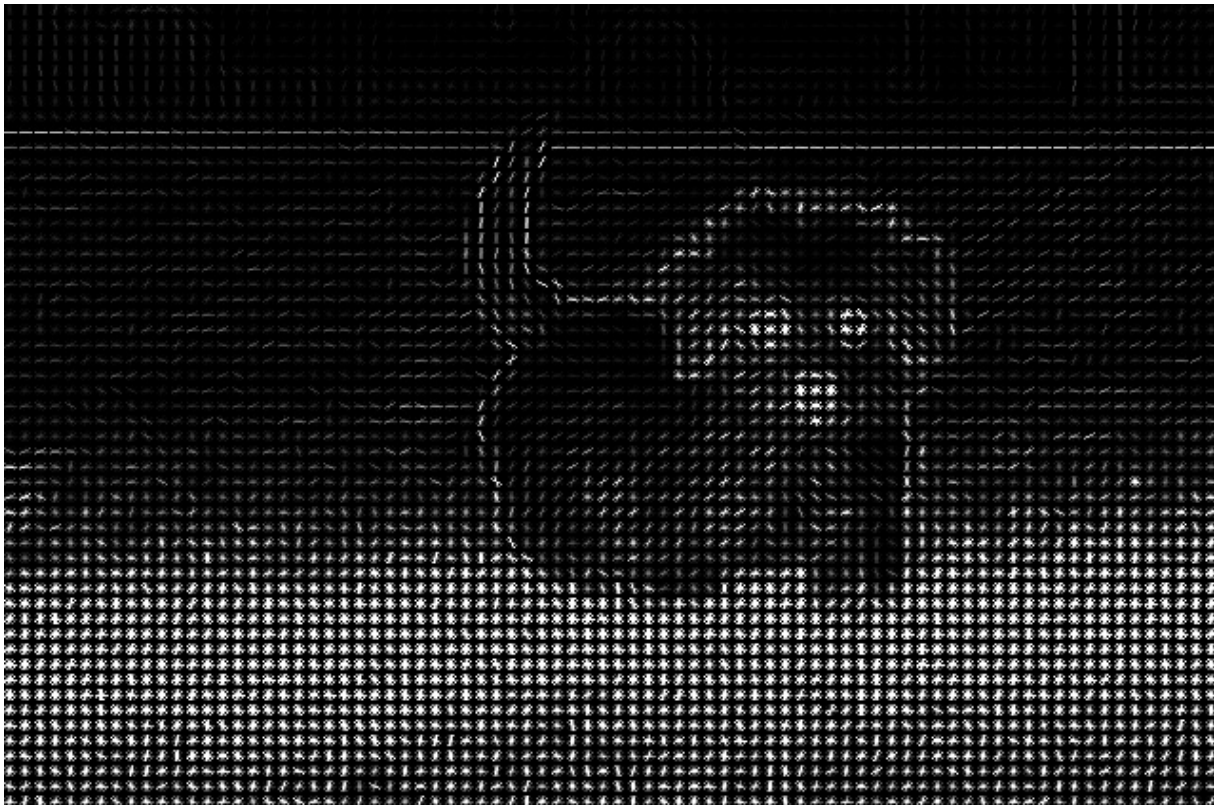


Figura 1.1: Reconocimiento de la imagen de un perro usando Histogram of Oriented Gradients (HOG)

2. Preprocesamiento

El primer paso que debemos realizar en el proyecto es el preprocesamiento de imágenes; en éste se aplicarán diversas técnicas y filtros que o bien mejorarán algunos problemas de la calidad de las imágenes o facilitarán la tarea de la extracción de características. Por ejemplo, si existieran imágenes con ruido de tipo sal y pimienta, podría ser útil aplicar un filtro de la mediana, y un filtro de suavizado gaussiano podría quitar peculiaridades de cada imagen y facilitar la generalización .

Para realizar el preprocesamiento hemos primero estudiado las imágenes y observado que en su mayoría no tienen problemas de ruido, además tampoco tienen problemas específicos en cuanto a la distribución de intensidades, si fueran todas muy oscuras o muy claras se podría aplicar una función gamma, pero al ir variando con cada imagen utilizamos una ecualización del histograma para asegurarnos de que todas tengan un rango similar. También aplicamos un filtro de suavizado para evitar pequeñas zonas de ruido particulares de cada imagen.

También aplicamos un filtro simple de realzado de bordes para facilitar la detección de esquinas. Al resaltar los bordes de cada imagen, será más sencillo identificar bordes.

Además de estas técnicas ya mencionadas, también había que redimensionar las imágenes para la extracción de características posterior (queremos vectores de los mismos tamaños) por lo que, tomando el tamaño máximo y mínimo de ambos ejes entre todas las imágenes, asignamos la media de las dos para mantener la máxima información posible para todas las imágenes.

Cabe destacar que durante el preprocesamiento sólo hemos utilizado procesos bastante generales, para no perder mucha eficiencia analizando píxel a píxel cada imagen. Utilizamos técnicas sencillas para no perder más tiempo en el preprocesamiento que en la extracción de características.

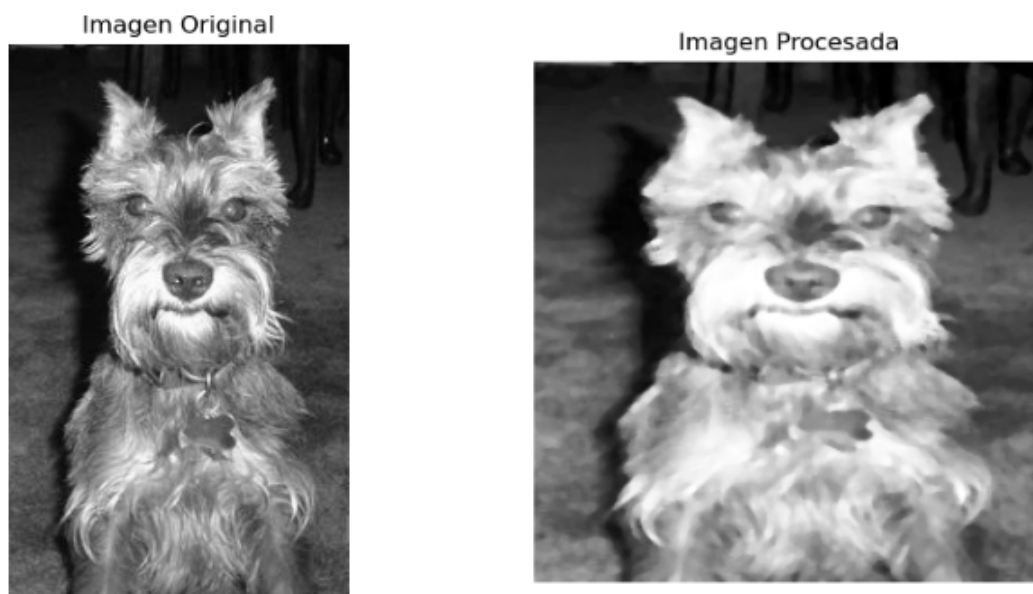


Figura 2.1: Comparación imagen original con imagen preprocesada

3. Extracción de Características

La extracción de características es un paso fundamental en nuestro pipeline de clasificación de imágenes, ya que permite representar cada imagen mediante un vector numérico de dimensiones fijas. Este vector captura información clave sobre la estructura y contenido de la imagen, lo que facilita al clasificador la tarea de distinguir entre clases (en este caso gatos y perros). Nuestro objetivo principal aquí fue obtener descriptores robustos ante variaciones en iluminación, o pequeñas deformaciones, buscando así mejorar la precisión global de la clasificación.

3.1. Histograma de Gradientes Orientados (HOG)

Optamos por HOG como uno de los primeros métodos de extracción por su efectividad en capturar la forma y los bordes predominantes en una imagen. HOG funciona calculando la magnitud y orientación del gradiente en cada píxel; luego agrupa esta información en celdas y finalmente construye un histograma que describe las direcciones de mayor variación de la intensidad en cada región.

3.1.1. ¿Por qué hemos usado HOG?:

- Captura de contornos y formas: En imágenes de gatos y perros, los contornos, la forma de la cabeza y las orejas suelen ser rasgos distintivos. HOG es particularmente bueno extrayendo este tipo de información estructural.
- No se ve muy afectado por cambios de iluminación: Al basarse en gradientes normalizados, HOG no se ve muy afectado por variaciones moderadas de brillo o contraste.
- Eficiencia computacional: HOG es relativamente rápido de calcular comparado con otros métodos como SIFT.

En la práctica, experimentamos con distintos tamaños de celda (por ejemplo, 8×8 o 16×16) y con diferentes números de bins en el histograma. Finalmente, elegimos los parámetros que ofrecieron una buena relación entre detalle de la forma y coste computacional.

3.2. Detección de Puntos Singulares (Harris, SIFT y ORB)

Además de HOG, incorporamos métodos de detección de puntos singulares. Estos métodos buscan localizar características “únicas” dentro de la imagen, por ejemplo, esquinas, intersecciones de bordes o patrones locales complejos.

- * Harris: Este método detecta esquinas al analizar cómo varían los gradientes en pequeñas vecindades de la imagen. Dado que las esquinas suelen aparecer en las siluetas y rasgos distintivos de gatos y perros (como hocicos, orejas u ojos), lo seleccionamos para capturar estas características. Para vectorizar la información obtenida, transformamos las respuestas de Harris en un histograma bidimensional que cuantifica la posición y densidad de esquinas en la imagen.
- * SIFT (Scale-Invariant Feature Transform): Además de detectar puntos clave, SIFT genera descriptores que son invariantes a escala, rotación e iluminación, lo que lo hace robusto frente a variaciones de pose y iluminación. Decidimos emplearlo por su capacidad para capturar detalles locales significativos. Promediamos los descriptores para formar un vector de características fijo, facilitando así el entrenamiento de clasificadores estándar.
- * ORB (Oriented FAST and Rotated BRIEF): ORB combina la detección rápida de esquinas mediante FAST con descriptores BRIEF, esto es un método eficiente y adecuado para aplicaciones en tiempo real. Lo utilizamos como una alternativa más ligera a SIFT para evaluar su desempeño en comparación. Los descriptores generados se promedian o rellenan con ceros si es necesario, asegurando un vector de características de tamaño fijo para los clasificadores.

3.3. Combinación de HOG y Puntos Singulares

Decidimos combinar HOG con los descriptores de puntos singulares (Harris, SIFT, ORB) para mejorar la representación de las imágenes. Mientras HOG captura la estructura global, como contornos y formas generales, los descriptores de puntos singulares aportan información local más detallada y robusta frente a cambios en escala o iluminación. Esta combinación permite que el modelo aproveche lo mejor de ambos enfoques, aunque también incrementa la dimensionalidad de los datos, lo que puede requerir técnicas adicionales de normalización y reducción de características para evitar un costo computacional excesivo.

3.4. Normalización y Reducción de Dimensionalidad

Para evitar que un tipo de descriptor domine al otro por tener magnitudes más altas, aplicamos técnicas de normalización que centran los valores en media 0 y desviación estándar 1.

3.5. Impacto en la Clasificación

Finalmente, usamos distintos clasificadores (Random Forest, SVM, Logistic Regression, etc.) para medir el aporte real de cada tipo de descriptor. De esta manera, pudimos observar:

- Cómo HOG destaca en la detección de contornos, beneficioso en situaciones con bordes bien definidos.
- Cómo la detección de puntos singulares (Harris, SIFT, ORB) refuerza la robustez ante variaciones en pose, iluminación o ruido de fondo.
- Que la combinación de descriptores mejora la representación de la imagen, aumentando generalmente la precisión del sistema.
- En conjunto, este enfoque nos ha permitido alcanzar una tasa de acierto bastante buena sin recurrir a redes neuronales profundas, demostrando así la efectividad de las técnicas de visión clásicas para la clasificación de perros y gatos.

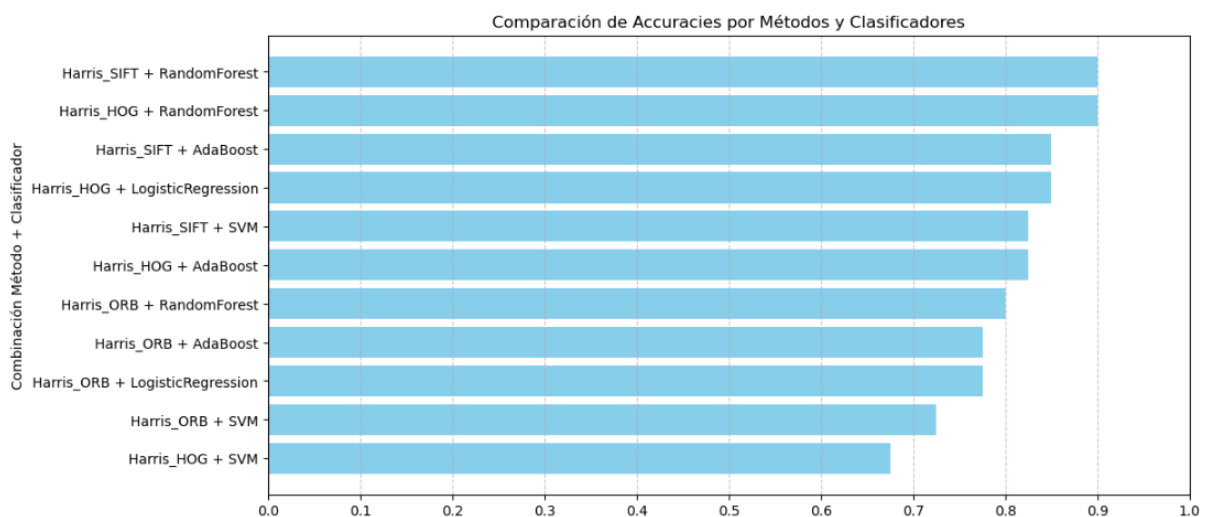


Figura 3.1: Resultados de precisión para diferentes combinaciones de métodos de extracción de características y clasificadores

4. Interpretación de los Resultados

Los resultados obtenidos muestran que las combinaciones de Harris_HOG + RandomForest y Harris_SIFT + RandomForest son las más efectivas, alcanzando un 90 % de precisión. Esto se explica porque HOG es particularmente bueno capturando contornos y formas globales en las imágenes, mientras que SIFT aporta robustez frente a variaciones como cambios en la escala o rotación de los objetos. Por su parte, Random Forest demuestra ser un clasificador que gestiona eficazmente datos con alta dimensionalidad y relaciones complejas entre las características.

Por otro lado, combinaciones como Harris_HOG + LogisticRegression y Harris_SIFT + AdaBoost logran un 85 % de precisión, lo que sugiere que AdaBoost aprovecha bien las características locales de SIFT, mientras que la Regresión Logística se ve beneficiada por la representación estructurada que genera HOG.

En contraste, las combinaciones basadas en ORB muestran un Accuracy más bajo, con un máximo de 80 % de precisión. Esto podría deberse a que los descriptores generados por ORB, aunque computacionalmente eficientes, no logran capturar tanta información como HOG o SIFT. Finalmente, SVM tiene un rendimiento variable dependiendo del descriptor utilizado, mostrando mejores resultados con SIFT debido a su capacidad para manejar características con más información.

En resumen, los métodos HOG y SIFT destacan como las mejores estrategias de extracción de características, especialmente cuando se combinan con Random Forest, que demostró ser el mejor clasificador en este problema y que es por tanto el que hemos decidido que es el más idóneo.

4.1. ¿Por qué Random Forest es el mejor clasificador?

Random Forest es el mejor clasificador porque puede manejar eficientemente datos de alta dimensionalidad como los generados por HOG y SIFT. Esto se debe a que combina múltiples árboles de decisión, cada uno entrenado con subconjuntos aleatorios de características, lo que mejora la generalización y reduce el riesgo de sobreajuste. Además, es robusto frente al ruido y modela bien tanto relaciones lineales como no lineales, aprovechando la diversidad en las características globales (HOG) y locales (SIFT) para lograr una precisión superior.

4.2. ¿Por qué SVM tiene un rendimiento más bajo con HOG?

El SVM tiene un rendimiento más bajo con HOG porque, al usar un kernel lineal, no puede captar relaciones complejas en las características que genera HOG. Además, aunque normalizamos los datos, las diferencias en la escala de las características pueden dificultar que el modelo separe correctamente las clases. Por último, el tamaño reducido del dataset y la alta cantidad de características de HOG complican la optimización del modelo, afectando su precisión.

4.3. Análisis de las imágenes mal clasificadas

Haciendo uso de la función `imprimirMalClasificados(best_y_pred, yTest, XTest)`, imprimimos las imágenes mal clasificadas por nuestro mejor clasificador (Harris_HOG + RandomForest) para analizar por qué está fallando.

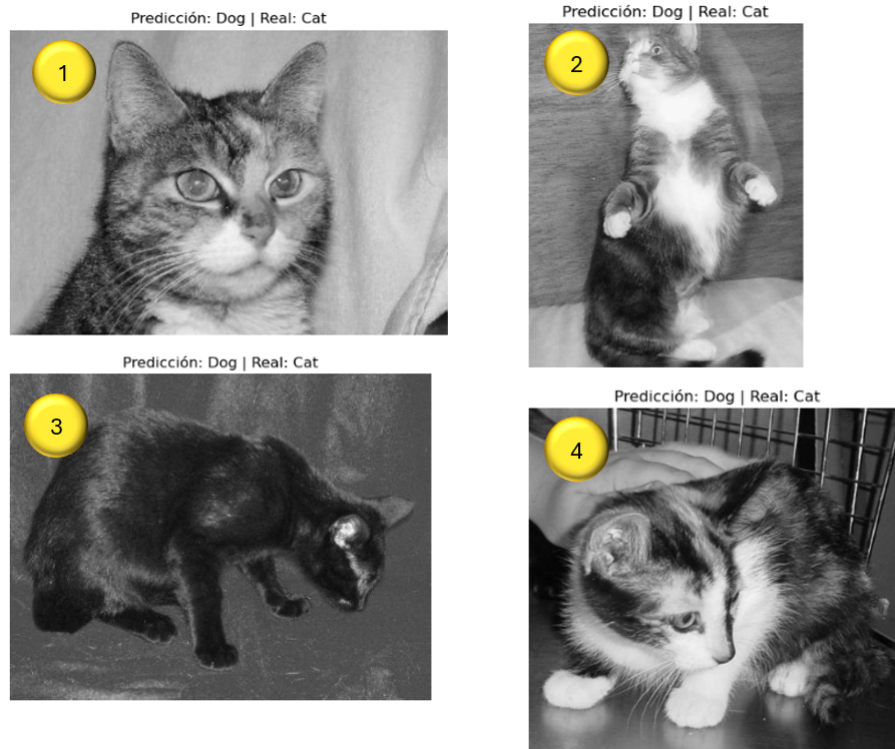


Figura 4.1: Resultados de precisión para diferentes combinaciones de métodos de extracción de características y clasificadores

Como se puede apreciar, únicamente falla al clasificar gatos (los clasifica como perros). Creemos que se debe a que el método HOG se basa principalmente en bordes y contornos globales. Y como se puede apreciar en las imágenes 3 y 4 se puede confundir el contorno del fondo con los contornos del animal. Asimismo, en la imagen 2, el gato tiene una postura poco común, lo que puede hacer que se confunda con un perro.

En lo referente a la imagen 1, el fondo de la imagen es claro y tiene bordes suaves. Esto puede llevar a que el descriptor HOG capture contornos del fondo, desviándose de los contornos relevantes del gato. Además, si en el conjunto de entrenamiento hay perros con fondos o postura similar, el modelo podría confundir esta imagen con un perro.

La matriz de confusión nos ayuda a visualizar este problema a la hora de clasificar gatos.

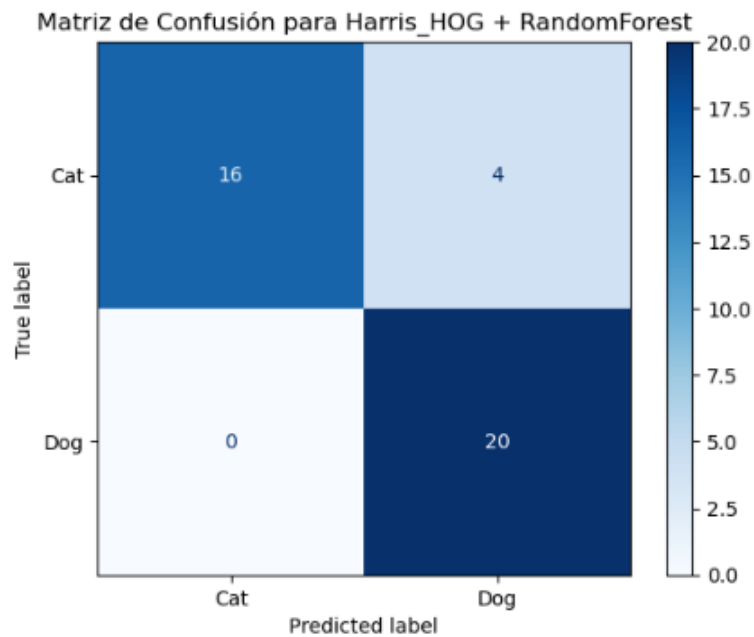


Figura 4.2: Matriz de confusión

4.4. Resultados

	Método de Extracción	Clasificador	Precisión
1	Harris_HOG	RandomForest	0.9
2	Harris_SIFT	RandomForest	0.9
3	Harris_HOG	LogisticRegression	0.85
4	Harris_SIFT	AdaBoost	0.85
5	Harris_HOG	AdaBoost	0.825
6	Harris_SIFT	SVM	0.825
7	Harris_SIFT	LogisticRegression	0.825
8	Harris_ORB	RandomForest	0.8
9	Harris_ORB	AdaBoost	0.775
10	Harris_ORB	LogisticRegression	0.775
11	Harris_ORB	SVM	0.725
12	Harris_HOG	SVM	0.675

Figura 4.3: Resultados de precisión para diferentes combinaciones de métodos de extracción de características y clasificadores

5. Conclusión del Trabajo: Clasificador Cat vs Dog

A lo largo de este proyecto, hemos trabajado en el diseño y desarrollo de un clasificador para distinguir entre imágenes de gatos y perros utilizando técnicas clásicas de visión por computador. Este enfoque nos permitió explorar todas las etapas del pipeline de procesamiento, desde el preprocesamiento hasta la clasificación, y reflexionar sobre las ventajas y limitaciones de estos métodos en comparación con enfoques más modernos. A continuación, compartimos nuestras principales conclusiones.

1. Uso de técnicas clásicas

Decidimos abordar este problema con herramientas clásicas como HOG, Harris, SIFT y ORB, dejando de lado el Deep Learning, para comprender mejor cómo funcionan estos métodos más tradicionales. Aunque somos conscientes de que las redes neuronales suelen ser más precisas, estas técnicas demostraron ser efectivas en datasets pequeños como el que utilizamos. Esto nos ayudó a valorar la flexibilidad y simplicidad de los métodos clásicos, especialmente en escenarios donde no se dispone de grandes volúmenes de datos ni de hardware especializado.

2. Etapas clave del trabajo

- **Preprocesamiento:** El preprocesamiento fue fundamental para mejorar la calidad de las imágenes y facilitar la extracción de características. Aplicamos técnicas como la ecualización del histograma y el realce de bordes, lo que permitió destacar detalles importantes como las texturas y los contornos. Este paso nos mostró la importancia de preparar las imágenes adecuadamente antes de pasar a las siguientes etapas.
- **Extracción de características:** Probamos diferentes métodos para extraer características relevantes de las imágenes. Entre ellos, HOG fue el más consistente, ya que capturó bien las estructuras y patrones relevantes. Además, descubrimos que combinar características como Harris y HOG podía mejorar los resultados en algunos casos, lo que nos dio una idea de cómo integrar descriptores para obtener una mejor representación de las imágenes.
- **Clasificación:** Experimentamos con varios clasificadores, como Random Forest, AdaBoost, SVM y Logistic Regression, y también probamos un Voting Classifier. Nos sorprendió descubrir que los métodos de ensemble, como Random Forest y Voting Classifier, ofrecieron los mejores resultados, ya que supieron manejar bien las variaciones en las características extraídas.

3. Resultados obtenidos

El mejor desempeño lo logramos con la combinación Harris-HOG y Random Forest, que alcanzó la mayor precisión en el conjunto de prueba. Sin embargo, notamos que el tamaño reducido del dataset (200 imágenes de entrenamiento) fue un factor limitante. A pesar de ello, creemos que los resultados son satisfactorios dado el enfoque clásico que utilizamos.

4. Limitaciones del proyecto

Reconocemos que nuestro trabajo tiene ciertas limitaciones:

- Tamaño del dataset:** Con un conjunto de datos mayor, los modelos podrían haber tenido un mejor desempeño, y la evaluación habría sido más representativa.
- Pipeline clásico:** Aunque útil para aprender, este enfoque requiere muchas etapas manuales y ajustes específicos, lo que lo hace menos práctico comparado con los modelos modernos.

Este proyecto nos permitió entender cómo las técnicas clásicas pueden ser aplicadas para resolver problemas simples de clasificación de imágenes. Aunque sabemos que estos métodos son superados por el Deep Learning en muchos casos, son una herramienta útil en escenarios donde los recursos son limitados o los datos escasos.