

Documentación práctica 1 INAR

1. Resumen del Modelo Original CNN

El modelo original contaba con una arquitectura CNN simple diseñada para clasificar imágenes de perros y gatos. Utilizaba capas convolucionales y densas en una configuración básica.

Tamaño de imagen: El modelo trabajaba con imágenes de (256, 256) píxeles.

Batch Size: El tamaño era de 125.

Capas Originales:

- 4 capas convolucionales, con un número de filtros creciente (32, 64, 128, y 256).
- Después de cada capa convolucional había una capa de MaxPooling2D con tamaño (2, 2) para reducir la dimensionalidad.
- Una capa de Dropout para reducir el sobreajuste.
- Una capa Flatten para aplanar la salida de las capas convolucionales antes de pasar a las capas densas.
- Una capa densa de 128 neuronas seguida de una capa de salida de 2 neuronas.

2. Cambios Realizados

En mi versión de la CNN, he realizado varios ajustes con el objetivo de mejorar su precisión. Los cambios realizados son los siguientes:

- Reducción del tamaño de la imagen:
 - Cambié el tamaño de las imágenes de entrada de (256, 256) a (128, 128). La razón de este ajuste fue reducir el peso del modelo, haciendo el entrenamiento más rápido y sin perder demasiada información visual para la clasificación.

- **Aumento del Número de Capas Convolucionales:**
 - Añadí una capa de Conv2D con 256 filtros y otra con 512 filtros. Cada una de estas capas tienen después una de MaxPooling.
 - Este aumento de la profundidad permite al modelo capturar características más complejas en etapas más profundas, para diferenciar entre imágenes con mayor precisión.
- **Ajuste de la capa densa final:**
 - En el modelo original, la capa densa final tenía 512 neuronas, la cual podía llevar al sobreajuste.
 - Reduje esta capa a 64 neuronas, disminuyendo el riesgo de sobreajuste.
- **Dropout:**
 - Añadí Dropout (0.5) después de la última capa convolucional y en la capa densa. Esto reduce el sobreajuste al evitar que el modelo dependa demasiado de ciertas neuronas.

3. Pruebas Realizadas

	A	B	C	D	E	F	G	H	I
1	Intento	Tamaño de Imagen	Capas Convolucionales	Capas de MaxPooling	Capas Dropout	Capas Flatten	Capas Densas	Épocas	Accuracy
2		1 (128, 128)	4	4	2	1	2	12	0,647
3		2 (128, 128)	5	5	2	1	2	12	0,689
4		3 (192, 192)	5	5	2	1	2	10	0,482

En el intento 1, usé el modelo base con una configuración que incluía un tamaño de imagen de (128, 128), cuatro capas convolucionales, cuatro capas de MaxPooling, dos capas de Dropout y dos capas densas en un entrenamiento de 12 épocas. Este modelo alcanzó un accuracy de 0.647.

En el intento 2, añadí una capa convolucional y una capa de MaxPooling para aumentar la profundidad del modelo. Mantuve el mismo tamaño de imagen y número de épocas. Estos cambios permitieron que el modelo capturara más detalles visuales, mejorando el accuracy a 0.689.

En el intento 3, aumenté el tamaño de imagen a (192, 192) para captar más detalles en cada imagen. Sin embargo, reduje las épocas a 10 para reducir el tiempo de entrenamiento, ya que este era muy extenso. Este cambio no fue del todo efectivo, ya que el accuracy cayó a 0.482, probablemente porque el modelo necesitaba más tiempo de entrenamiento.

En conclusión, el segundo intento consiguió el mejor balance entre profundidad y tiempo de entrenamiento.

4. Visualización de Capas Intermedias en la CNN

Para comprender mejor cómo la CNN interpreta las imágenes, he creado una visualización de las capas intermedias. Esto permite observar qué patrones visuales detecta el modelo en cada capa, mostrando cómo transforma la información desde las capas iniciales hasta las más profundas.

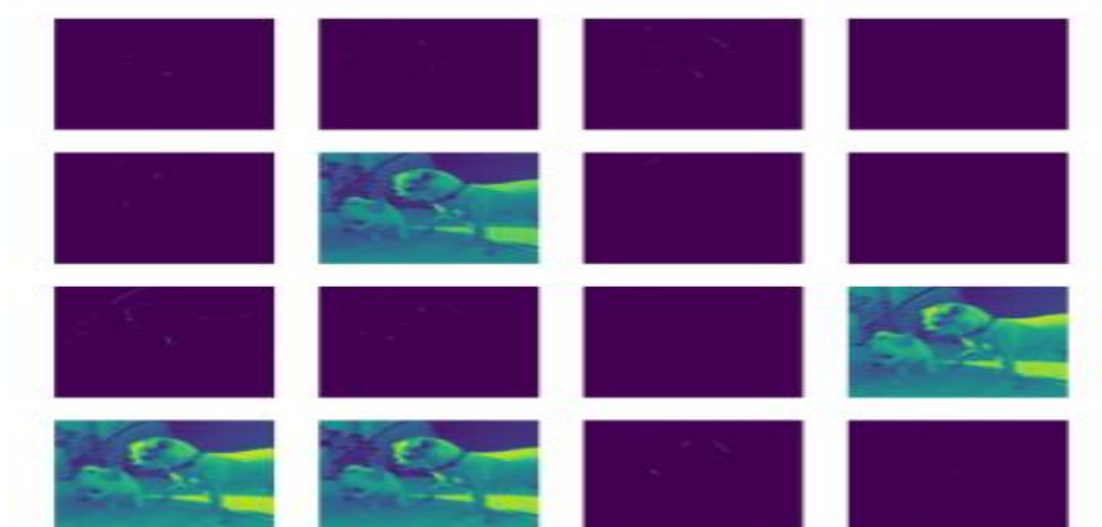
Metodología:

1. Selección de Capas: Se eligieron capas clave a diferentes profundidades de la red para observar el proceso de transformación de las características visuales.
2. Visualización: Se seleccionaron varias características detectadas por las capas convolucionales y se normalizaron para mejorar su interpretación visual.
3. Análisis: Se observó cómo las primeras capas tienden a detectar bordes y texturas generales, mientras que las capas posteriores captan patrones más específicos que pueden asociarse con características distintivas de los animales.

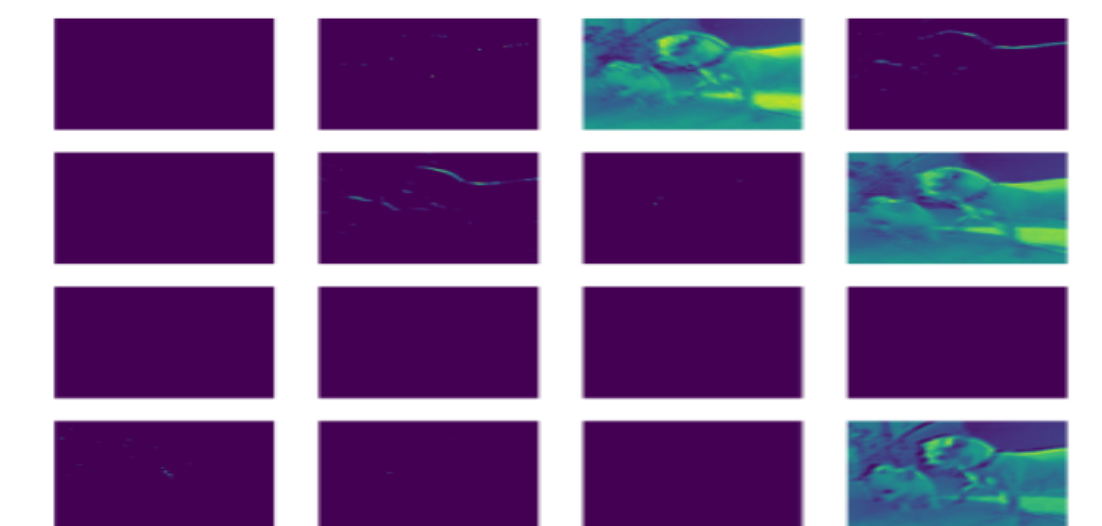
Capas Iniciales (Capas 0 y 2):

- Las primeras capas muestran activaciones relacionadas con bordes y texturas. En estas capas, el modelo se enfoca en detalles básicos, como contornos y cambios de color.
- En las activaciones, podemos ver que las zonas resaltadas corresponden a formas generales y no a detalles del animal. Esto es esencial para captar las estructuras iniciales antes de pasar a la detección de características más complejas en capas posteriores.

Activación de la capa 0



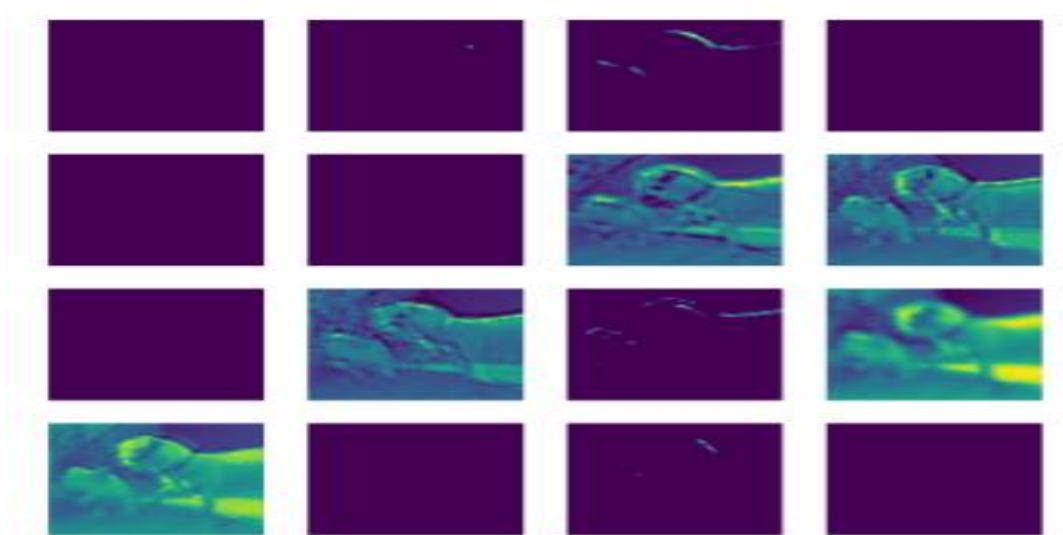
Activación de la capa 2



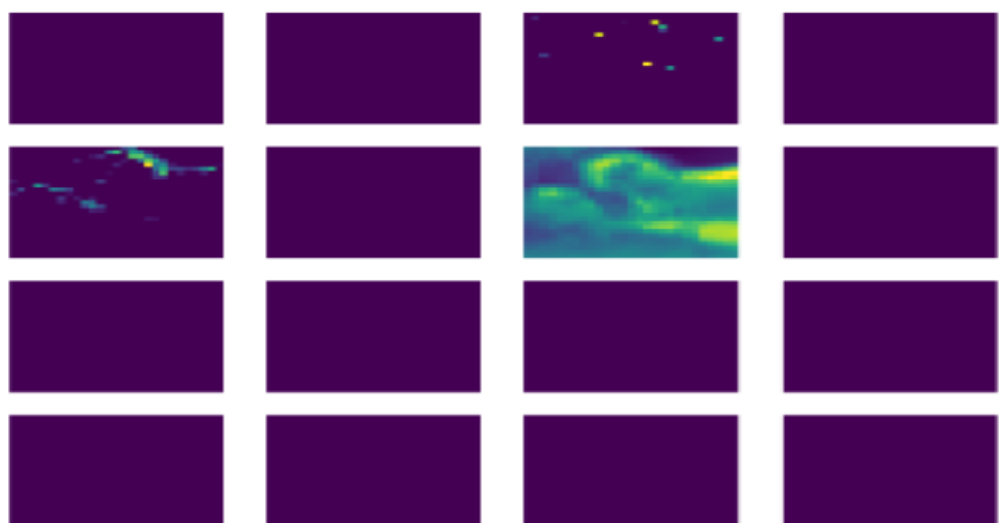
Capas Intermedias (Capas 4 y 6):

- En estas capas intermedias, las activaciones comienzan a mostrar patrones más complejos. Aquí, el modelo empieza a captar texturas y formas que podrían relacionarse con características más definidas de los animales, como el pelaje o partes del cuerpo.
- Observando las activaciones, notamos que algunas partes de la imagen comienzan a destacar más, lo que sugiere que el modelo está refinando la detección de detalles específicos. Esta fase de procesamiento ayuda a diferenciar entre perros y gatos al captar patrones de textura o formas únicas que caracterizan a cada clase.

Activación de la capa 4



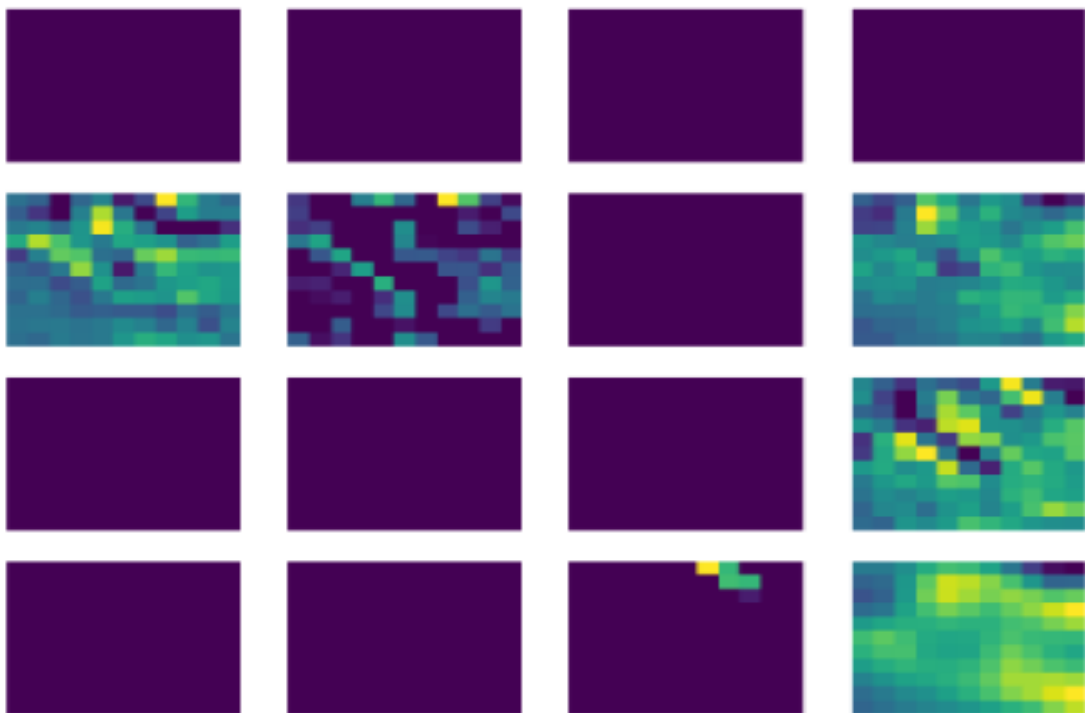
Activación de la capa 6



Capas Profundas (Capas 8 y superiores):

- En las capas finales, el modelo muestra activaciones concentradas en áreas que probablemente corresponden a detalles muy específicos o características distintivas. Es en estas capas donde se podría esperar que el modelo identifique patrones complejos, como ojos, orejas, o ciertos detalles del pelaje.
- En las imágenes de activación observadas, las zonas con activación parecen enfocarse en áreas más pequeñas de la imagen. Esto permite al modelo tomar decisiones de clasificación basadas en las características de más alto nivel, que son las más útiles para distinguir entre clases (perro y gato).

Activación de la capa 8



5. Exploración de errores y mejoras

Para revisar los errores del modelo, he creado un fichero llamado predictions.csv, que contiene las predicciones generadas para cada imagen de prueba, el valor 1 corresponde a perros y 0 a gatos. Para este análisis, me enfoqué he enfocado en las imágenes que van del 1000 hasta 1009.

El código que genera el fichero de las predicciones es el siguiente:

```
# Guardar las predicciones en el diccionario
predictions_dict[int(file_no_extension)] = label

# Convertir las predicciones en un DataFrame
predictions_df = pd.DataFrame(predictions_dict.items(), columns=["id", "label"])

# Ordenar el DataFrame por el ID de la imagen
predictions_df = predictions_df.sort_values(by='id', ascending=True)

# Guardar las predicciones en un archivo CSV
predictions_df.to_csv('predictions.csv', index=False)

print("Predicciones guardadas en predictions.csv")
```

Después de revisar los resultados, el modelo falló en clasificar estas imágenes:

- Imagen 1009: Interpretada como gato incorrectamente.
- Imagen 1004: Interpretada como gato incorrectamente.
- Imagen 1003: Interpretada como perro incorrectamente.
- Imagen 1002: Interpretada como gato incorrectamente.

Análisis de los errores:

1. Imagen 1009: Aquí aparecen dos perros, uno de ellos en primer plano. La imagen tiene una iluminación alta y varias sombras, lo que podría haber confundido al modelo, ya que podría relacionar sombras y zonas oscuras con características de un gato.
2. Imagen 1004: Es un perro grande acostado. La pose relajada y el ángulo en el que fue captada la imagen, podrían haber confundido al modelo a la hora de detectar los rasgos de un perro.
3. Imagen 1003: Esta es una foto de un gato al aire libre, en el que hay césped de fondo. Aunque es claramente un gato, la textura del pelaje y el entorno podrían haber hecho que el modelo no reconociera las características de gato.
4. Imagen 1002: Un perro pequeño jugando en el suelo con un juguete en la boca. Cuando los perros están en poses poco comunes o interactuando con objetos, el modelo parece tener problemas para identificar correctamente sus rasgos faciales.

Posibles causas de los errores:

Estos fallos indican que el modelo tiene problemas en situaciones como:

- Imágenes con varios elementos o sujetos.
- Perros en poses poco comunes o interactuando con objetos.
- Condiciones de iluminación que generan sombras y contraste fuertes.

Posibles mejoras del modelo:

Para mejorar en estos casos, podría ser útil aplicar Data Augmentation en el entrenamiento, agregando más variedad en poses, rotaciones, y ajustes de iluminación en las imágenes. Además, incluir más fotos en el dataset con características similares a las de estas imágenes podría ayudar al modelo a reconocer mejor estos escenarios.

1. Resumen del Modelo Original VGG16

El modelo original es una red pre entrenada VGG16 diseñada para clasificación en 1000 categorías, que cuenta con capas convolucionales y densas.

- Tamaño de imagen: Este modelo utiliza imágenes de tamaño (224, 224) píxeles.
- Batch Size: El tamaño por lotes es de 125.
- Capas:
 - Capas convolucionales: 5 bloques de convolución, cada uno con varias capas que suman hasta 13 capas convolucionales, aumentando progresivamente los filtros
 - Capas de MaxPooling: Cada bloque de convolución termina con una capa de MaxPooling para reducir las dimensiones.
 - Capas densas: Después de aplanar las características, cuenta con dos capas densas de 4096 neuronas y una capa de salida de 1000 neuronas.

2. Cambios Realizados

He adaptado el modelo VGG16 para optimizarlo en la tarea de clasificación de perros y gatos utilizando una VGG19. Los ajustes realizados son los siguientes:

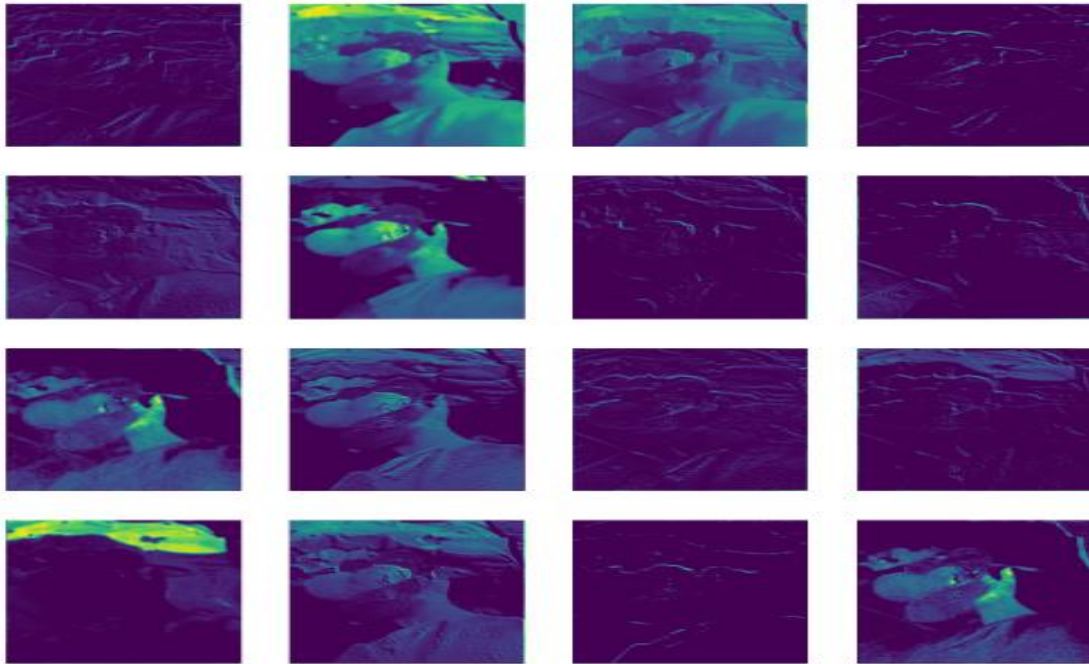
- Cambio de Modelo: En lugar del modelo VGG16, he utilizado un modelo VGG19 pre entrenado en ImageNet, el cual es más profundo, ya que cuenta con 19 capas convolucionales en lugar de las 16 de la VGG16.
- Congelación de Capas Iniciales: En esta versión, congelé todas las capas de la VGG19 excepto la última capa de salida, permitiendo que el modelo conserve las características de las capas pre entrenadas en ImageNet. Esto reduce el tiempo de entrenamiento y mejora la estabilidad del modelo.
- Capas Adicionales: Añadí capas densas adicionales después de la capa de aplanado para mejorar la capacidad del modelo de captar patrones más detallados en las imágenes de perros y gatos. Añadí las siguientes capas:
 - Una capa densa de 256 neuronas con activación ReLU y Dropout (0.5) para reducir el sobreajuste.
 - Una capa adicional de 128 neuronas seguida de otro Dropout (0.5), lo que ayuda a mejorar la generalización del modelo en el conjunto de validación.

3. Visualización de Capas Intermedias en el Modelo VGG19

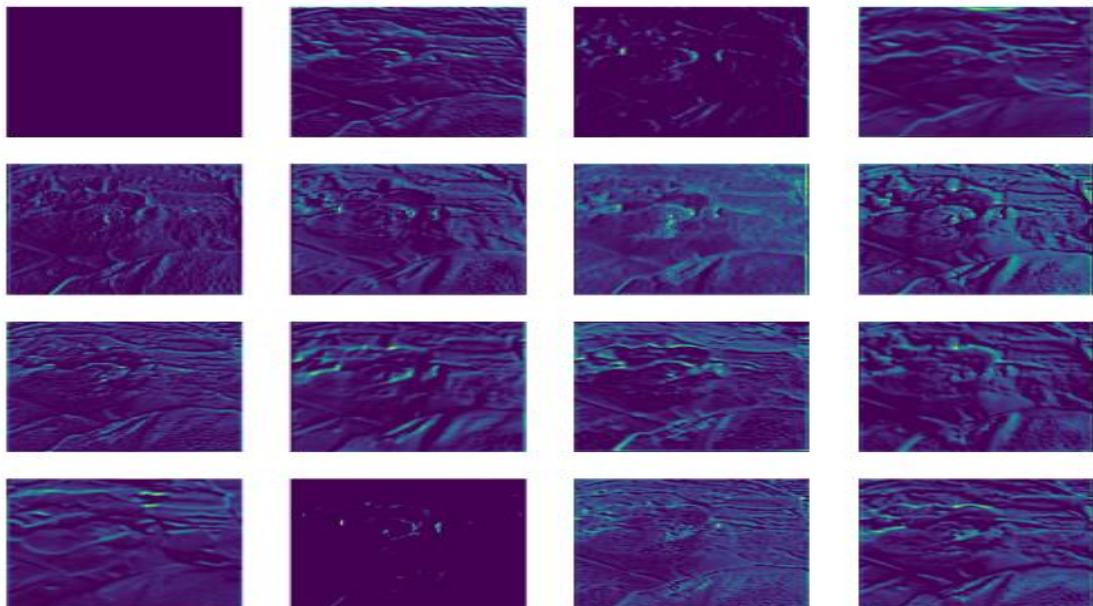
Capas Iniciales (Capas 1 y 4):

En estas primeras capas, el modelo detecta bordes y contornos básicos, enfocándose en formas y variaciones de color generales. Las activaciones resaltan áreas de alto contraste, ayudando al modelo a captar la estructura general de la imagen.

Activación de la capa 1



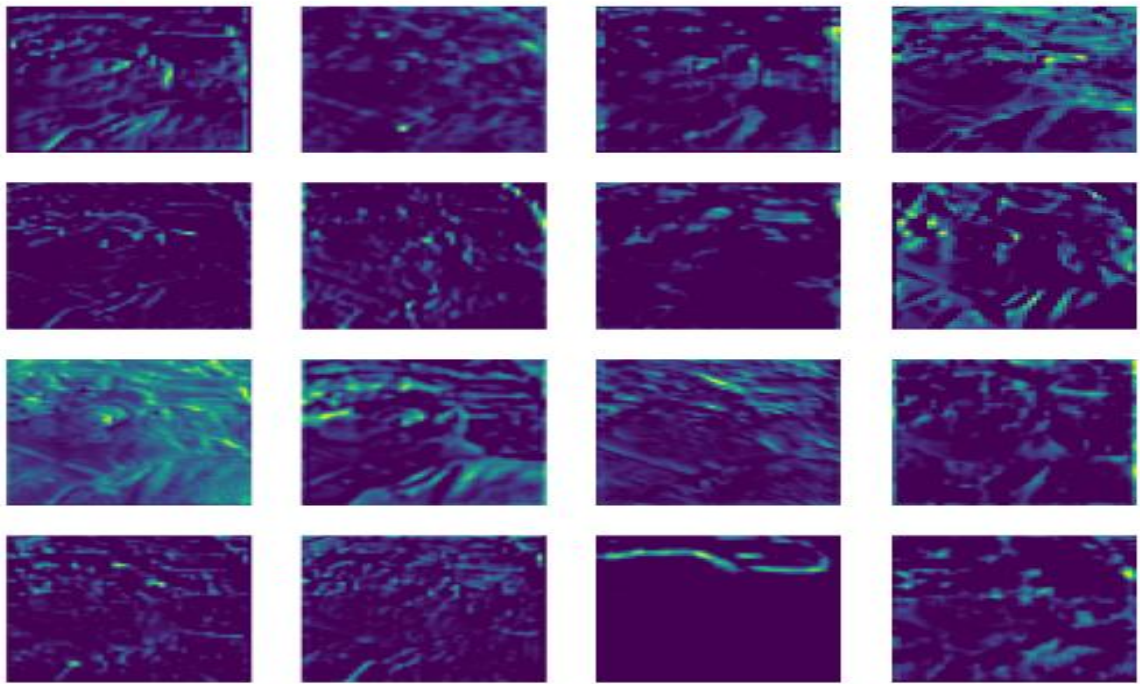
Activación de la capa 4



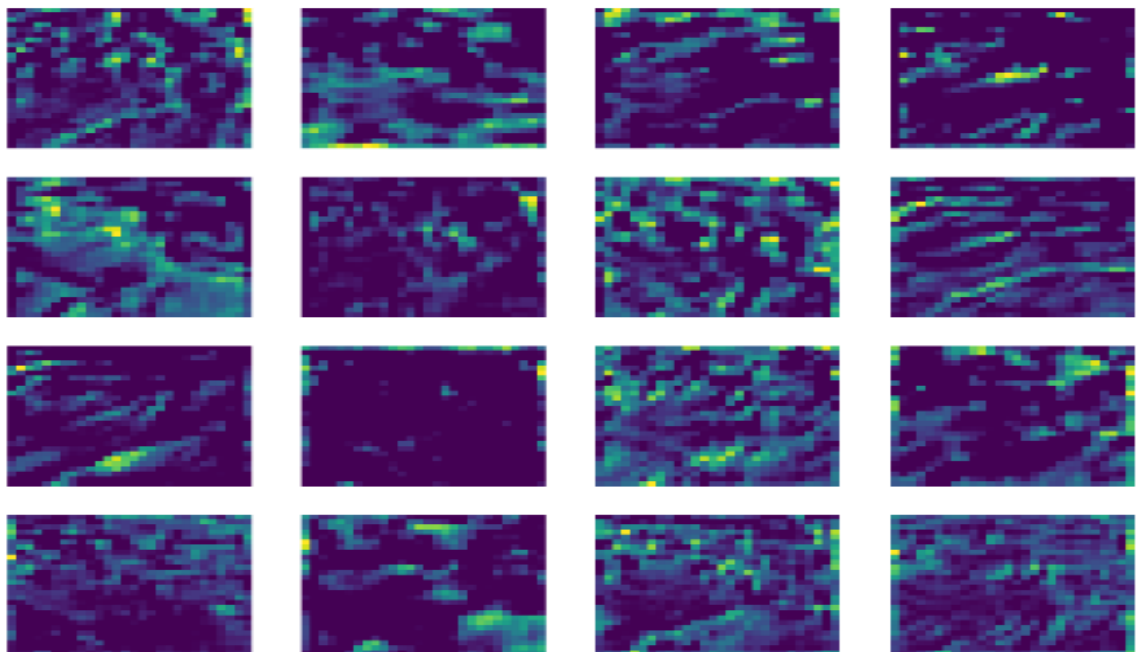
Capas Intermedias (Capas 7 y 12):

En las capas intermedias, el modelo empieza a captar texturas y formas específicas que ayudan a diferenciar entre perros y gatos. Las activaciones destacan detalles intermedios, como patrones de pelaje y áreas de textura que aportan mayor precisión a la clasificación.

Activación de la capa 7



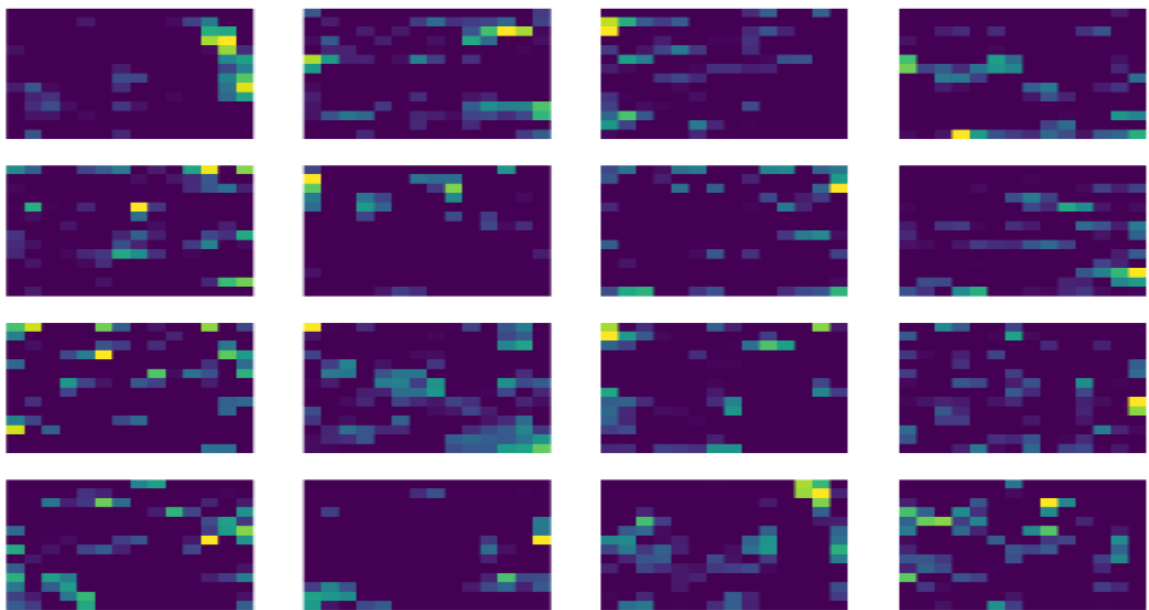
Activación de la capa 12



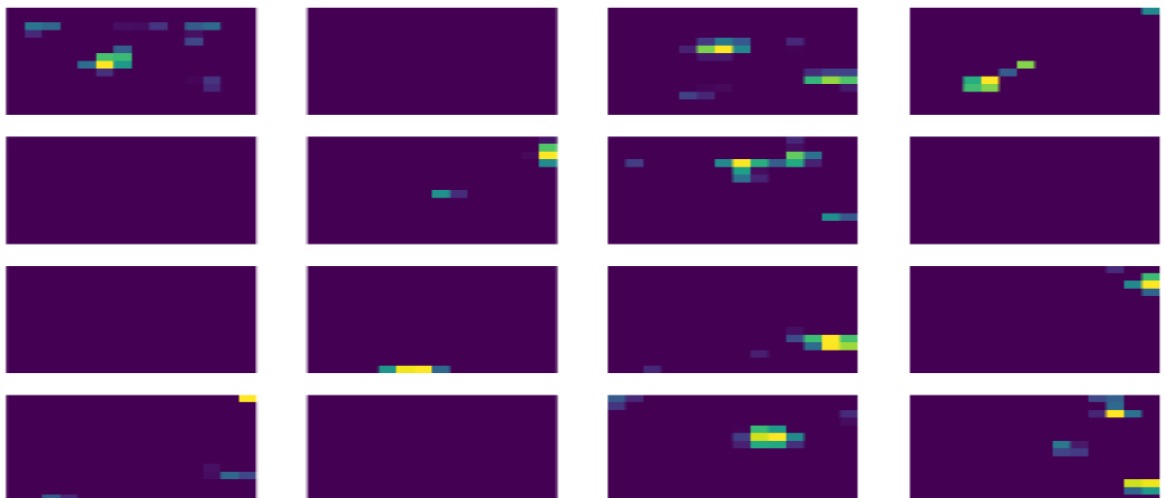
Capas Profundas (Capas 16 y 20):

En las capas profundas, el modelo se enfoca en características muy específicas y detalladas, esenciales para la clasificación final. Las activaciones se concentran en áreas pequeñas, identificando patrones exclusivos que permiten distinguir claramente entre ambas clases de animales.

Activación de la capa 16



Activación de la capa 20



4. Exploración de errores y mejoras

En este apartado, seguiré el mismo procedimiento que en la CNN para revisar la precisión del modelo VGG19 en un conjunto de 10 imágenes de prueba. Identificaré los errores de clasificación y analizaré las posibles causas de cada uno. Por último, una vez analizadas las imágenes, propondré una serie de mejoras para mi modelo VGG19.

Análisis de Errores:

1. Imagen 1004: Aquí tenemos a un perro tumbado en el suelo y captado desde un ángulo superior. La postura y el ambiente pueden dar una impresión más ambigua para el modelo, ya que no se resaltan características específicas de un perro. Como el perro ocupa un gran espacio en la imagen sin un enfoque claro en su cara, el modelo podría haber tenido problemas para capturar las características típicas de un perro.
2. Imagen 1005: La posición del gato de perfil y la iluminación pueden dificultar que el modelo distinga correctamente los rasgos de un gato, más aún, si su rostro no es visible.

3. Posibles Causas del Error:

- En ambas imágenes los rostros de los animales no son del todo visibles, lo cual parece ser una limitación para el modelo.
- En la imagen 1005, el gato está acostado en un entorno que carece de contraste claro, dificultando al modelo el identificar los límites y detalles del animal.

Posibles Mejoras para el Modelo:

- Data Augmentation: Agregar más imágenes de gatos y perros en poses y ángulos raros durante el entrenamiento podría ayudar al modelo a adaptarse a este tipo de situaciones.