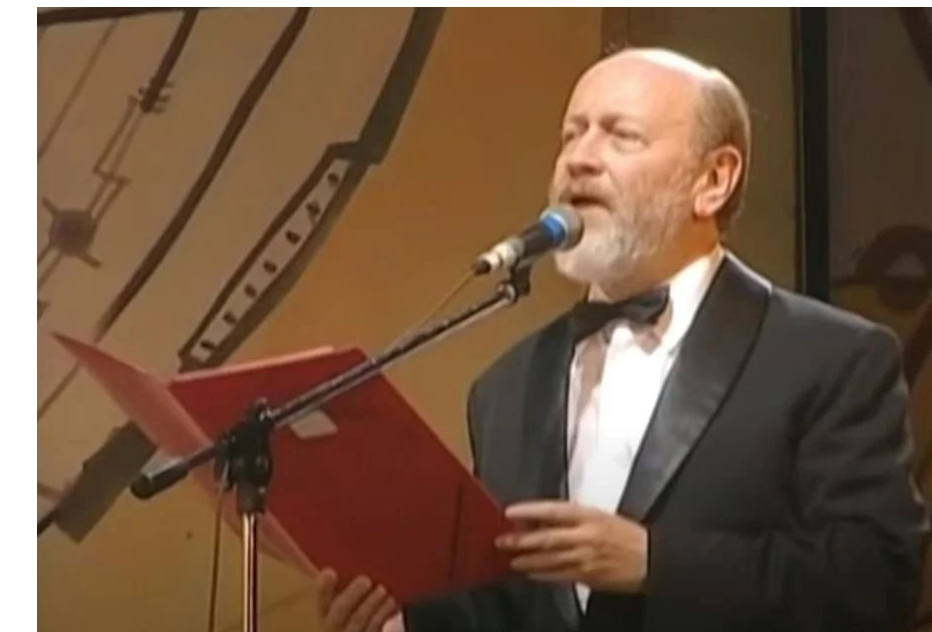

Nociones de Aprendizaje Automático para Aplicaciones Nucleares

Fuera de programa...



Extras...

- Lo que vimos en el curso les da una buena base para empezar a ver si alguna herramienta de aprendizaje automático les puede servir para atacar el problema con el que estén trabajando.
- Queríamos agregar un par de cosas prácticas adicionales que les pueden ser útiles cuando se pongan a trabajar.

Fine-tuning

- Vimos que las redes neuronales pueden aprender patrones complejos en los datos, lo que las hace extremadamente poderosas para diversas tareas.
- Sin embargo, cuando los datos de la tarea objetivo tienen una distribución ligeramente diferente a la del conjunto de datos original, podemos tener una merma en el rendimiento de la red.
- En estos casos, podemos usar nuestro modelo entrenado (o conseguir uno) y aplicar lo que se conoce como *fine-tuning* para ajustar el modelo a las características específicas del nuevo dominio.
- Lo que se suele hacer es tomar la red entrenada y “congelar” algunas capas (esto es, no se van a actualizar los pesos) y hacer un pequeño entrenamiento con los nuevos datos usando un *learning rate* bajo (*).
- Cuando el cambio en los datos y objetivos es más significativo, pero sigue existiendo una relación con la tarea original, se puede emplear *transfer learning*.
- Esto es particularmente útil en redes convolucionales, donde las primeras capas capturan características generales que pueden ser útiles en múltiples dominios.

(*) Otra opción puede ser entrenar toda la red, pero siempre con un *LR* pequeño. A modo de referencia el valor por default de Adam es 0.001

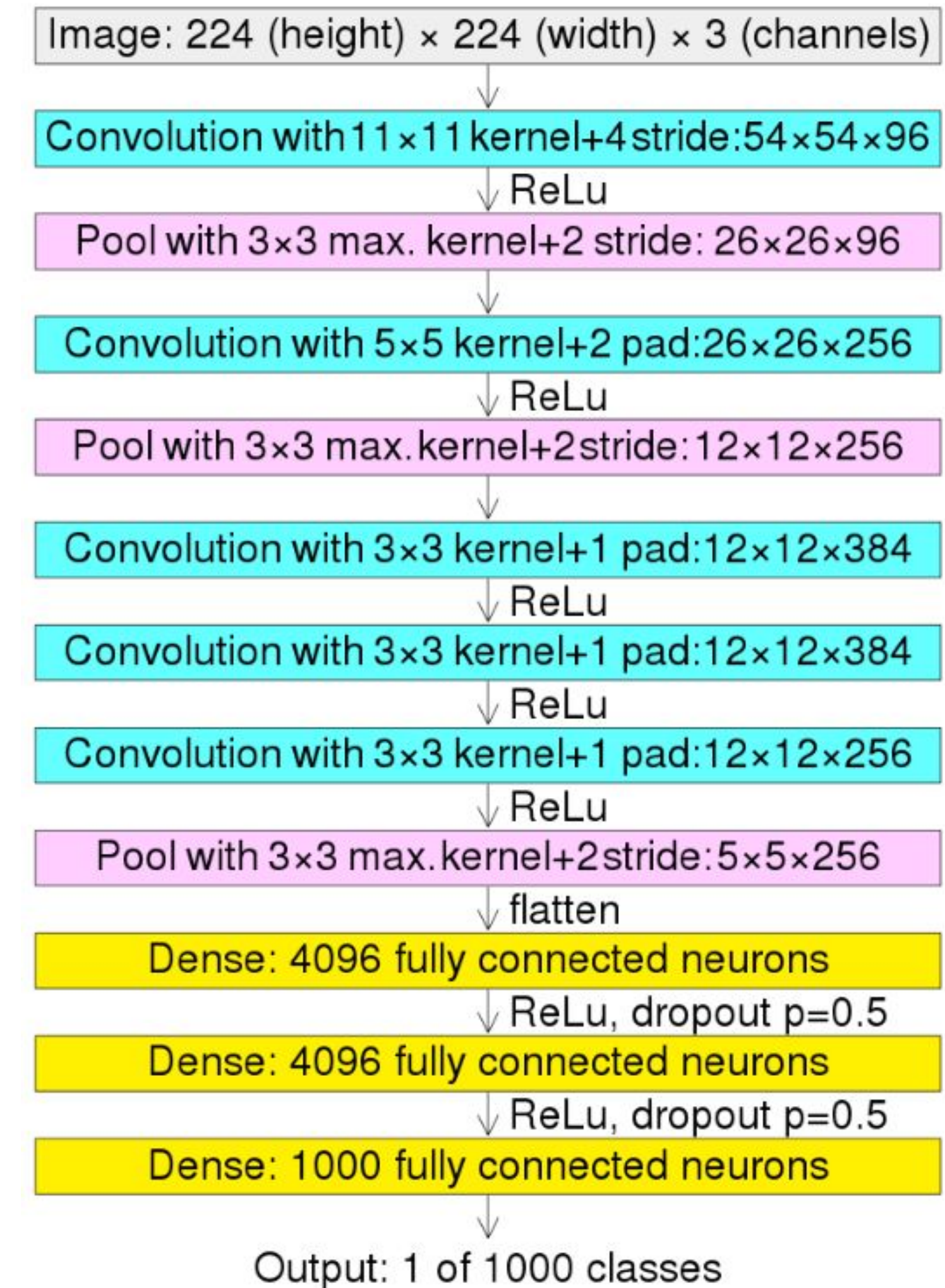
Transfer Learning

- Cuando hablamos de redes convolucionales dijimos cosas como:
 - “Las primeras capas de la red detectan patrones simples”.
 - “Los filtros se combinan y las capas más profundas detectan patrones más complejos”.
- En el notebook pueden hacer ejemplos y ver que funcionan, pero medio que sigue siendo una cuestión de fe.
- Vamos a mostrar primero un trabajo muy conocido que justamente ataca este problema, ¿cómo interpretar lo que está pasando dentro de una *CNN*.

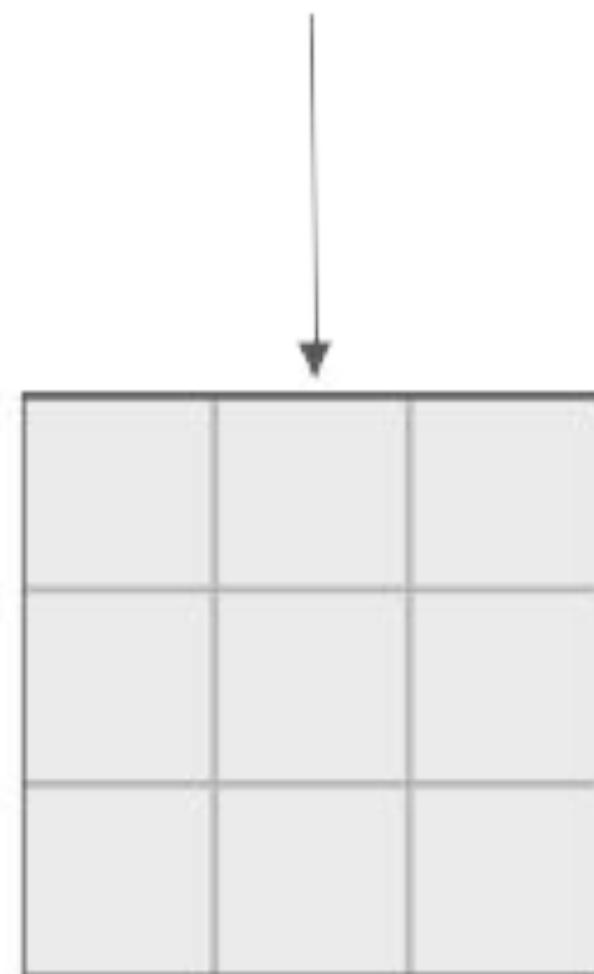
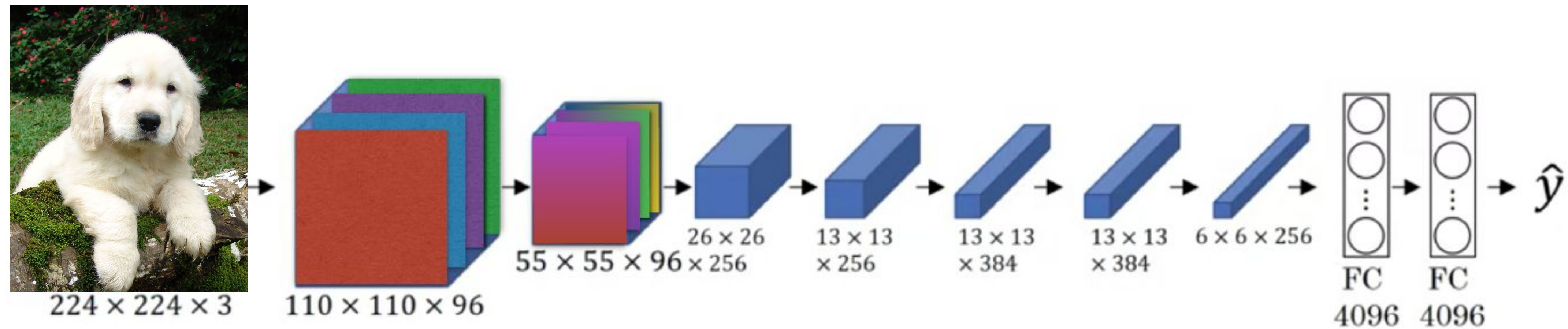
AlexNet

- Entrenada con más de un millón de imágenes.
- Clasifica 1000 categorías de objetos (como teclado, taza de café, lápiz y muchos animales).
- Compitió en ImageNet Large Scale Visual Recognition Challenge en 2012 y alcanzo un error de 15.3%, más de 10.8 puntos menos que el segundo.
- Una de las primeras en usar GPUs para la competencia.
- Superada en 2015 por una red de Microsoft que tenía más de 100 capas.
- En 2013, Visualizing and Understanding Convolutional Networks (<https://arxiv.org/abs/1311.2901>).

Fuente: Wikipedia



AlexNet



Toman los 9 parches de
pixeles que más activan
esta neurona (filtro)

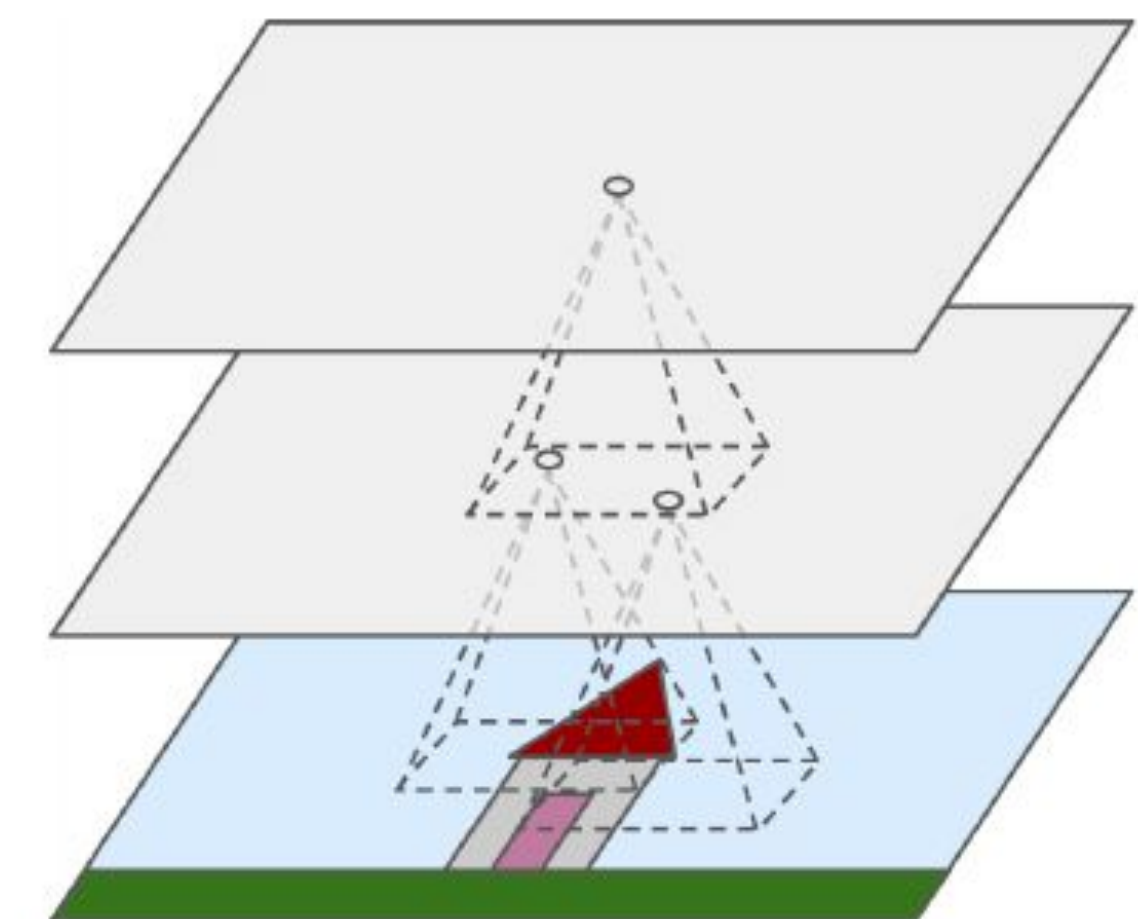


Imagen del Gerón

AlexNet

Neurona
(filtro 1)



AlexNet

Neurona
(filtro 1)

Neurona
(filtro 2)

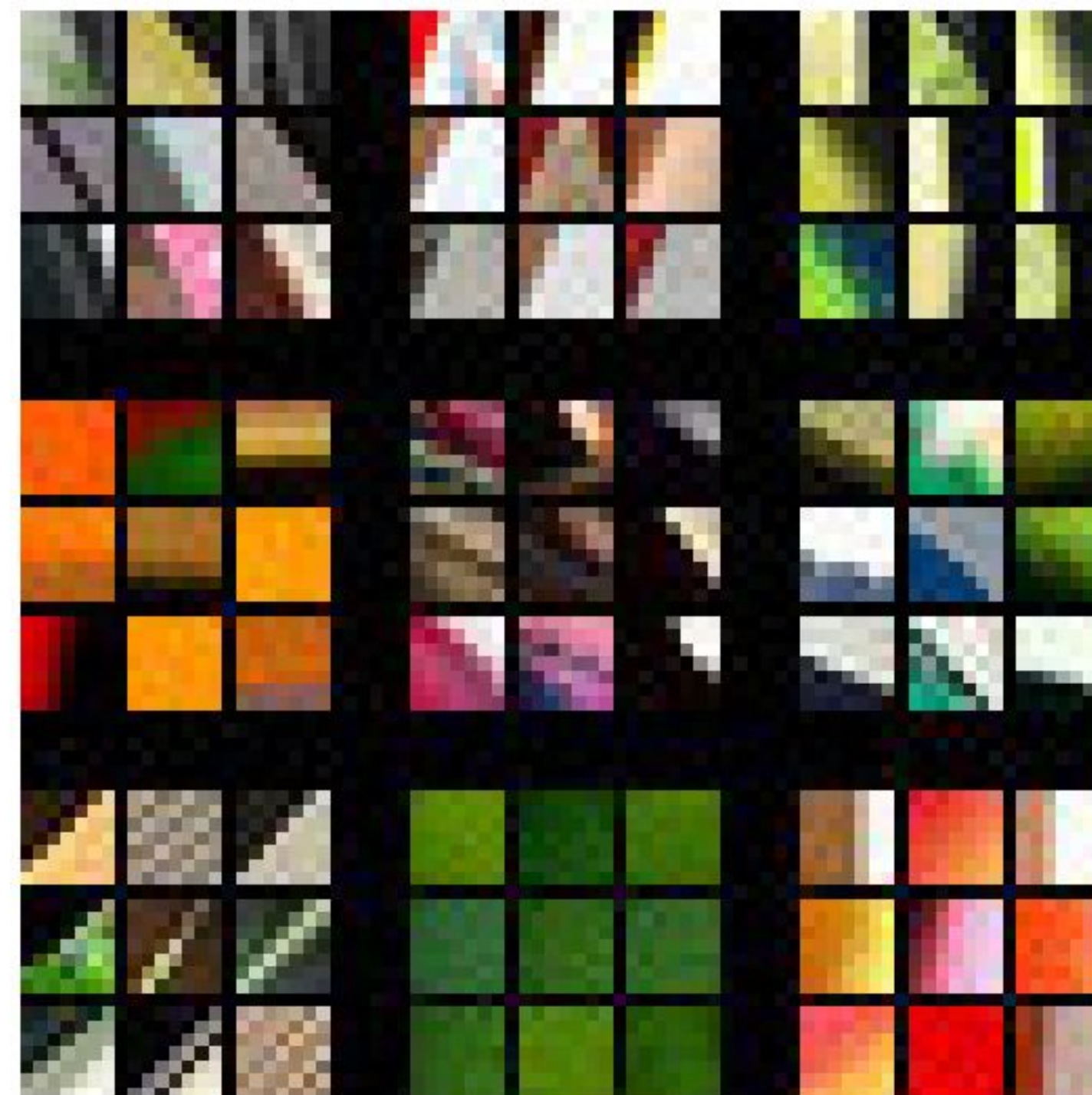


AlexNet

Neurona
(filtro 1)

Neurona
(filtro 2)

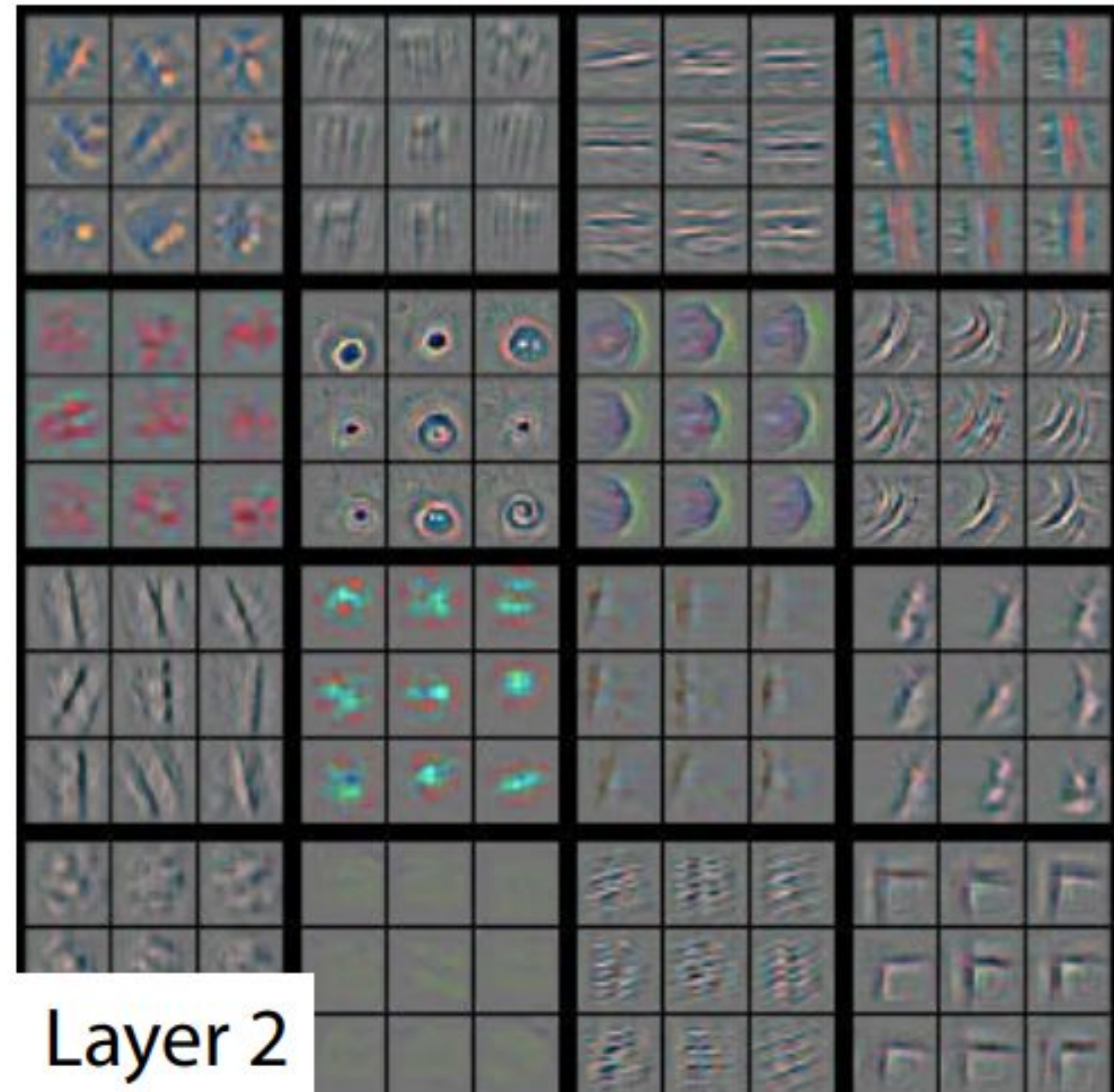
Neurona
(filtro 3)



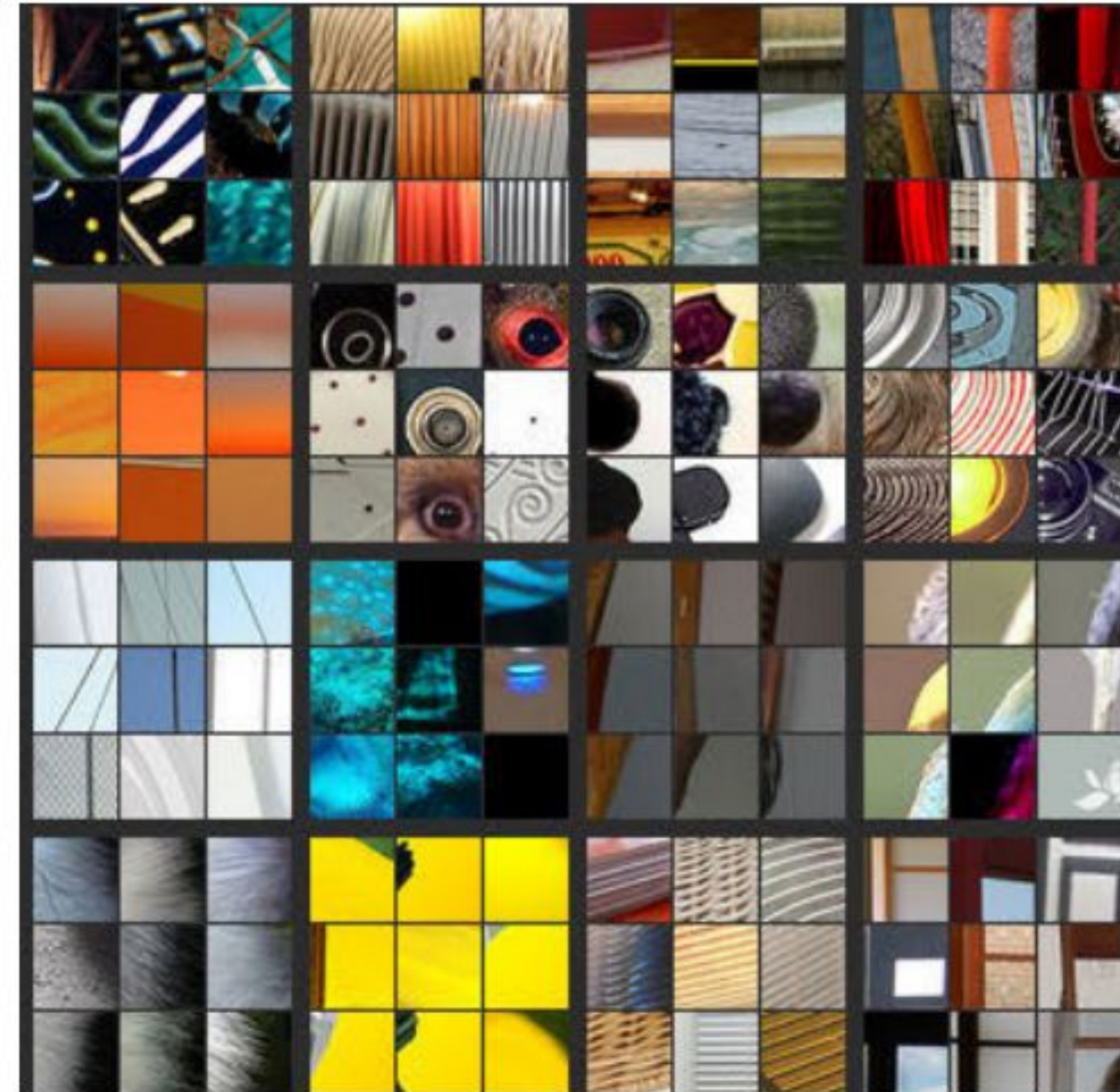
...

Neurona
(filtro 9)

AlexNet

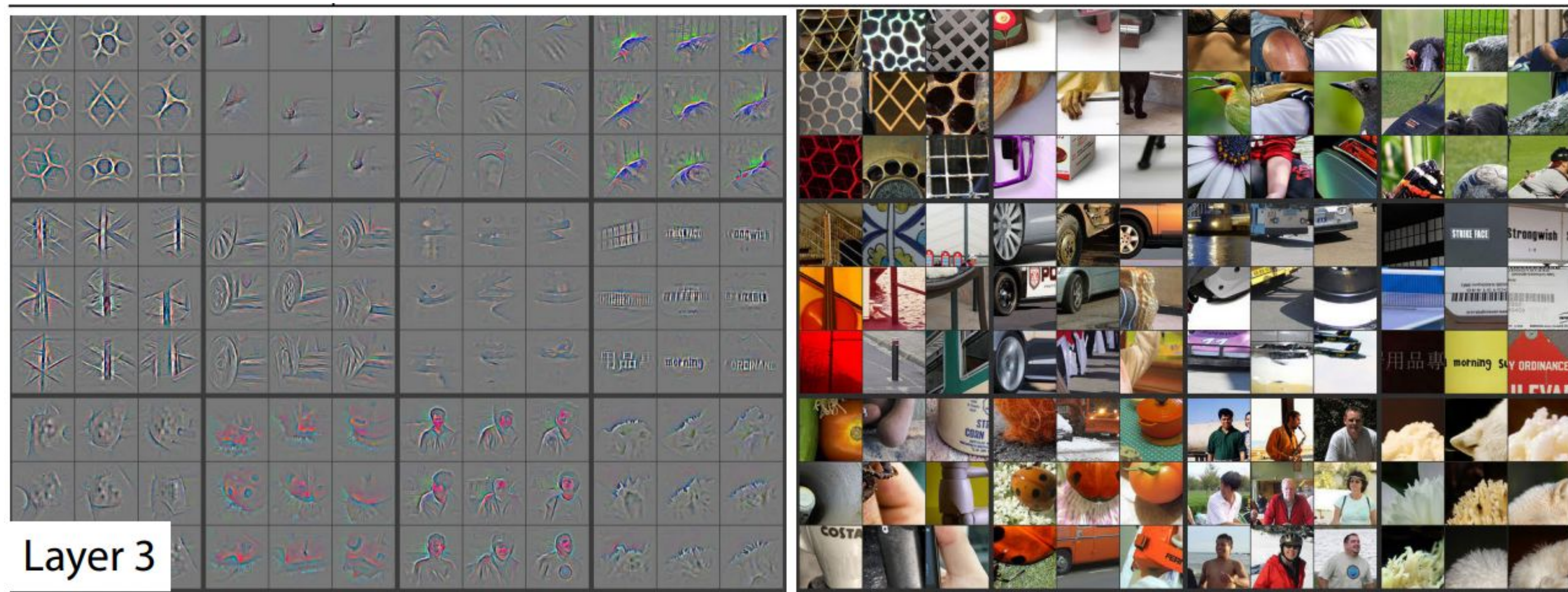


Patrones que maximizan la activación



Ejemplos de imágenes que maximizan la activación

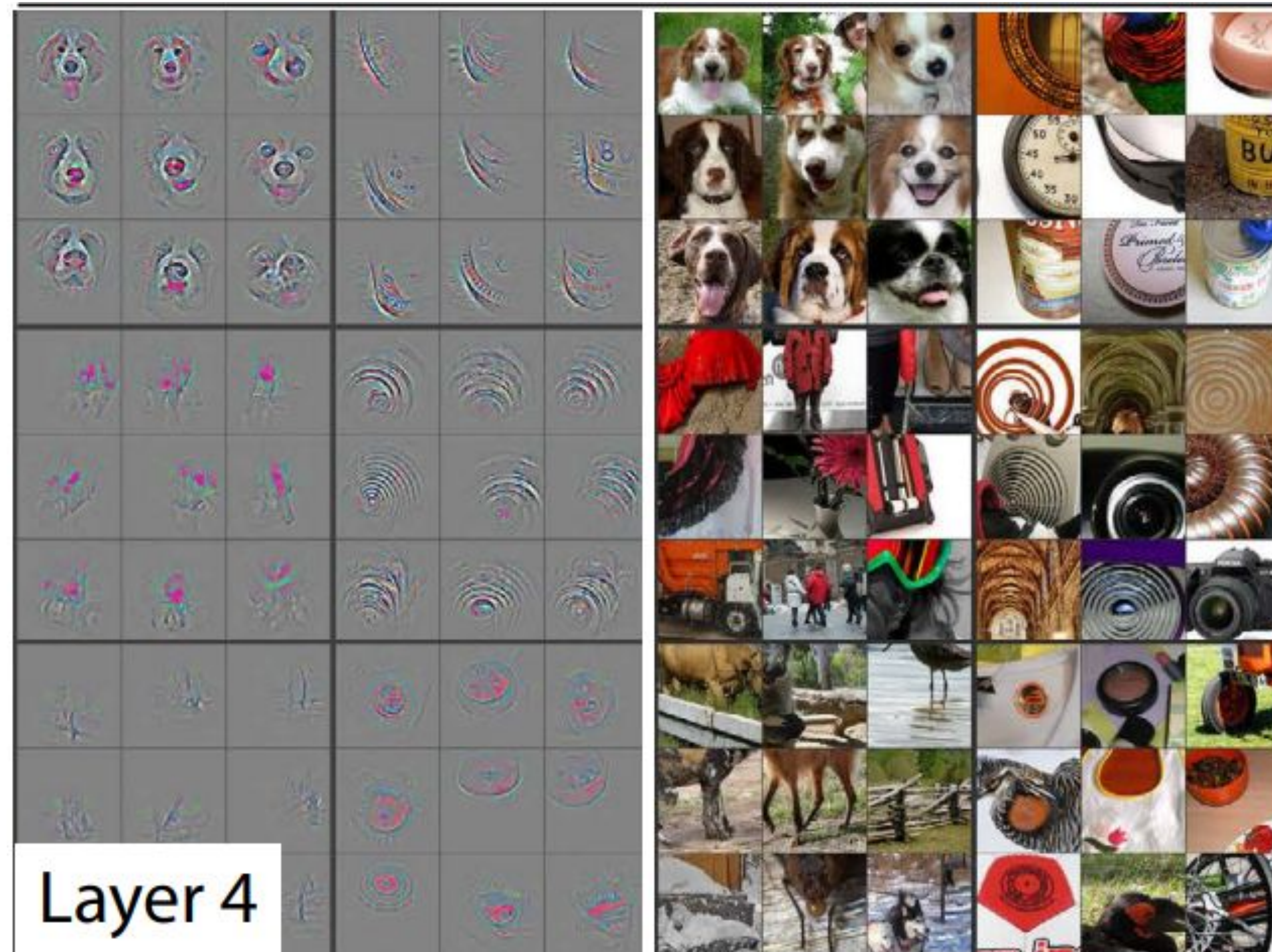
AlexNet



Patrones que maximizan la activación

Ejemplos de imágenes que maximizan la activación

AlexNet

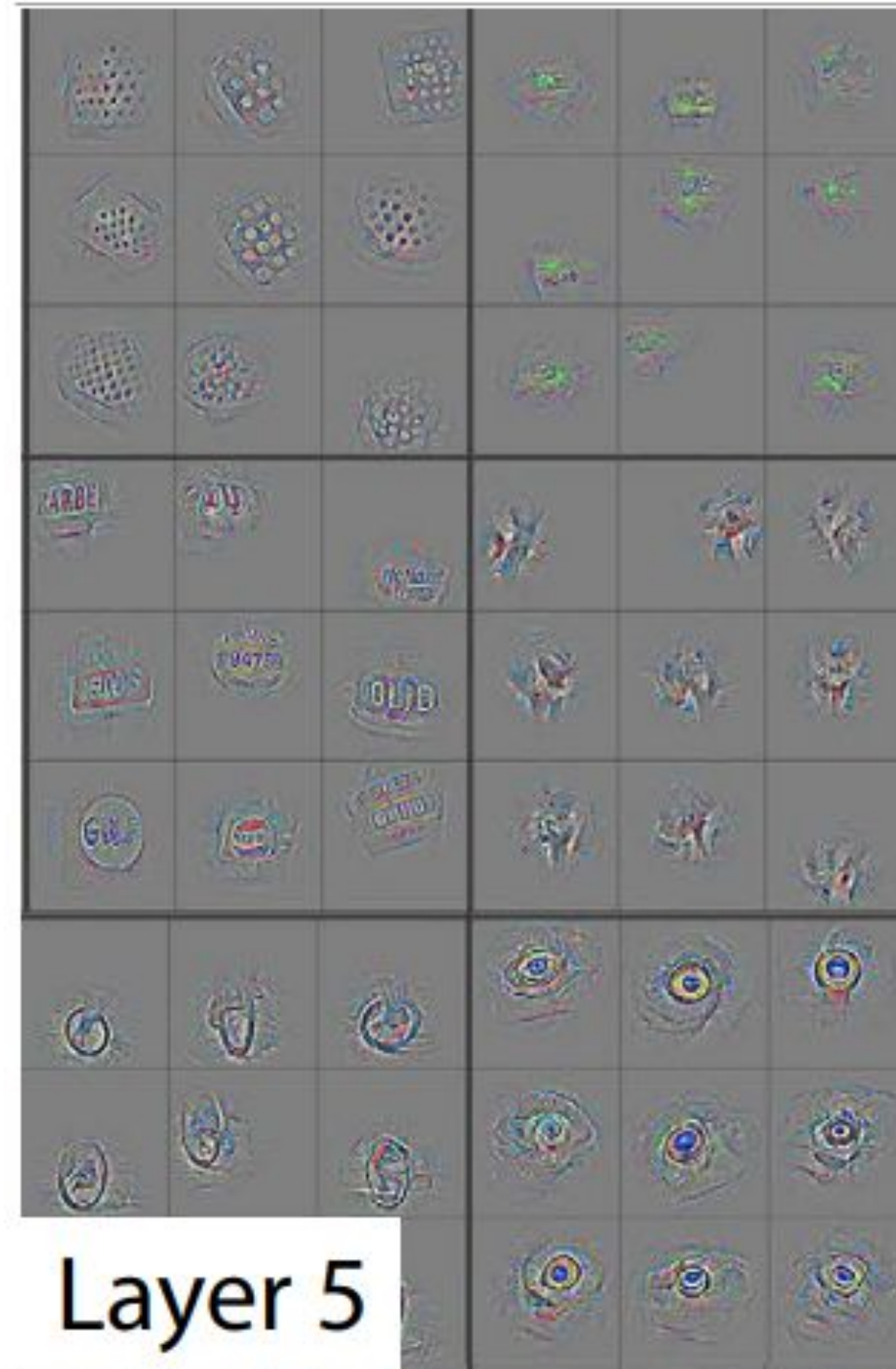


Layer 4

Patrones que maximizan
la activación

Ejemplos de imágenes que
maximizan la activación

AlexNet

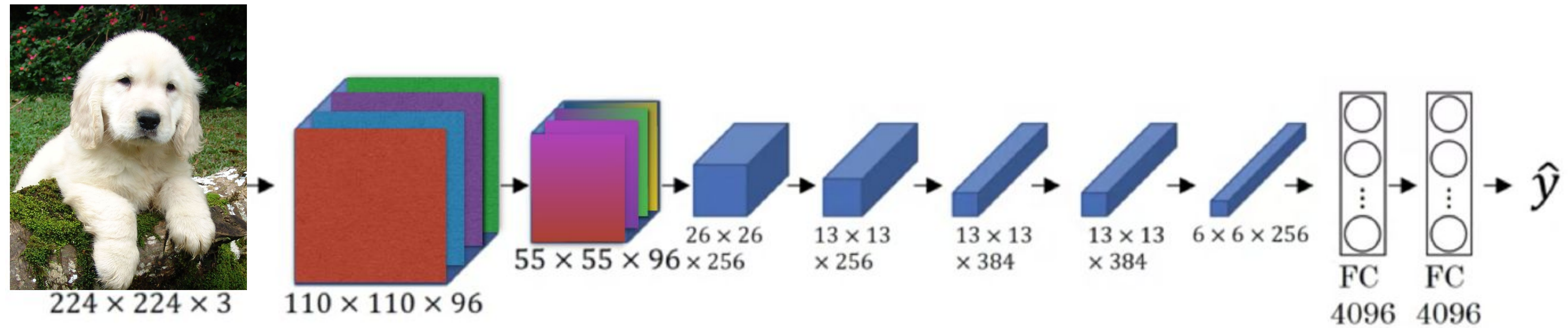


Patrones que maximizan la activación

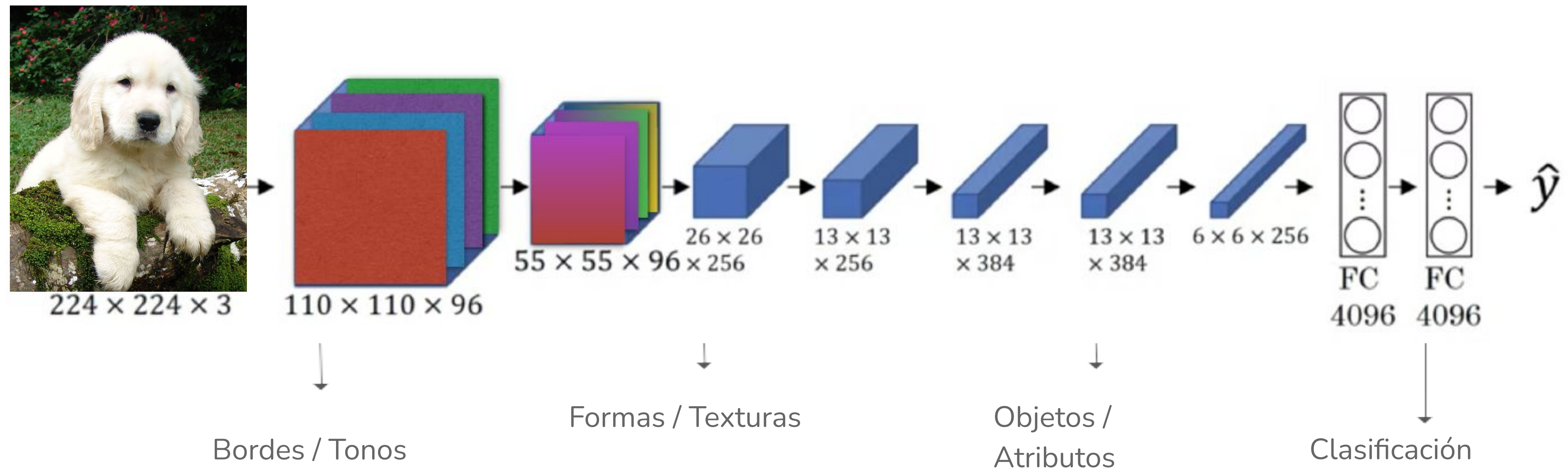


Ejemplos de imágenes que maximizan la activación

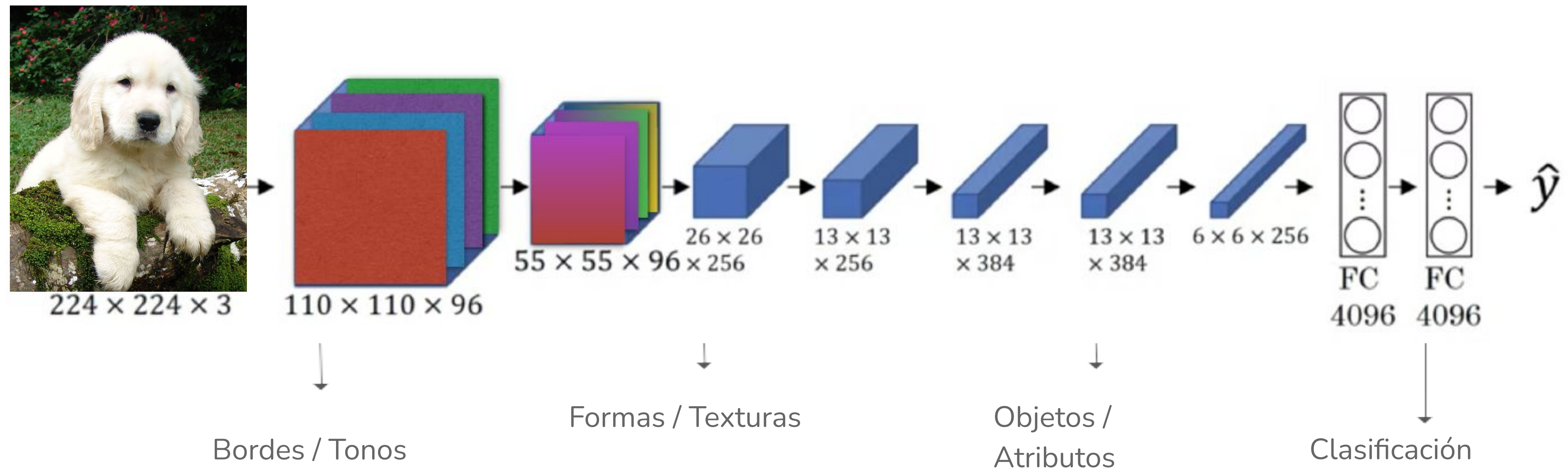
AlexNet



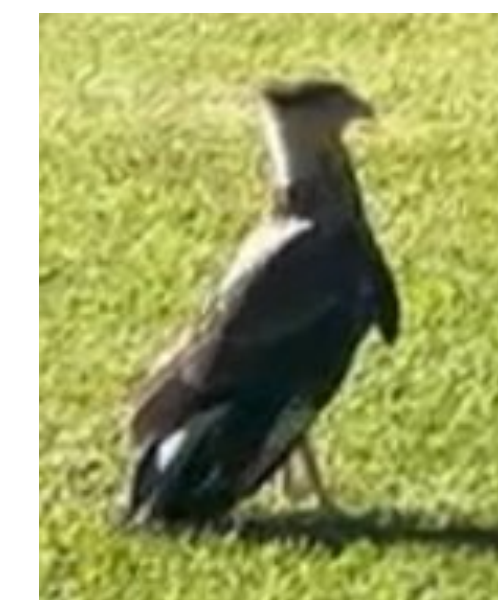
AlexNet



AlexNet



¿Qué partes les parece que pueden servir si queremos, por ejemplo, armar un clasificador que diferencia nuestra fauna autóctona de otros tipos de Caranchos?



*Carancho
Constituentis
(Fito)*



*Carancho
Chetus*

Transfer learning

- Las primeras capas de la red funcionan como detectores de características simples que son generales a todas las imágenes.
- Las más profundas ya son de objetos y atributos particulares de las imágenes con las que fue entrenada.
- Veamos como podemos aprovechar todo esto para hacer un clasificador de nuestros caranchos.
- Primero: nos armamos un dataset de imágenes para nuestro problema en particular.
- Segundo: nos conseguimos una AlexNet entrenada.

Transfer learning

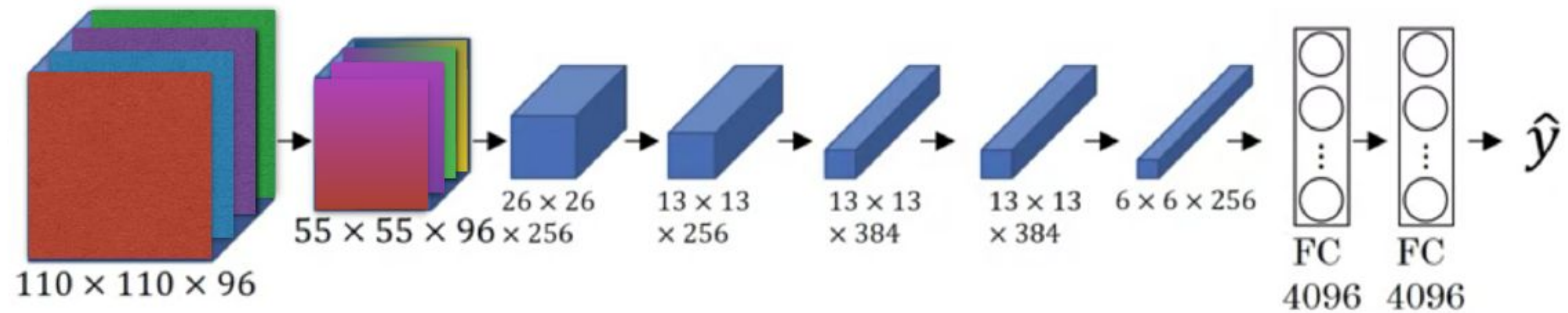


Label = 1

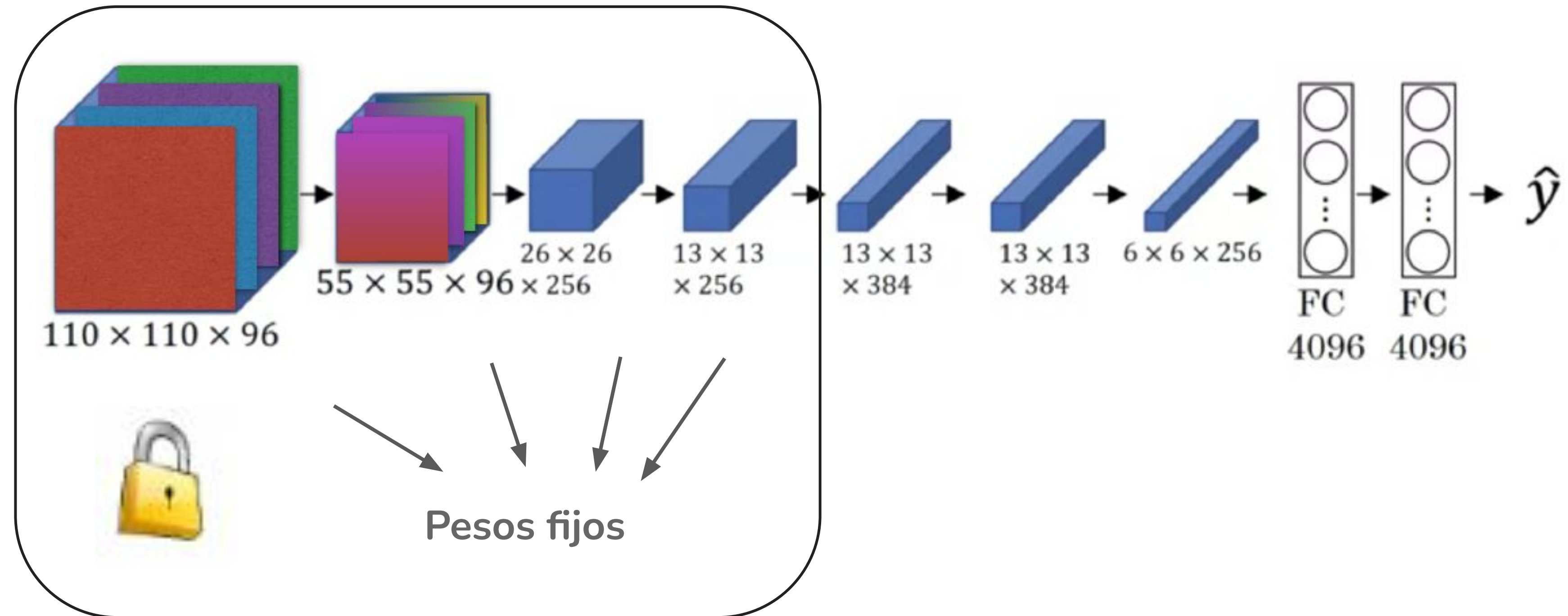


Label = 0

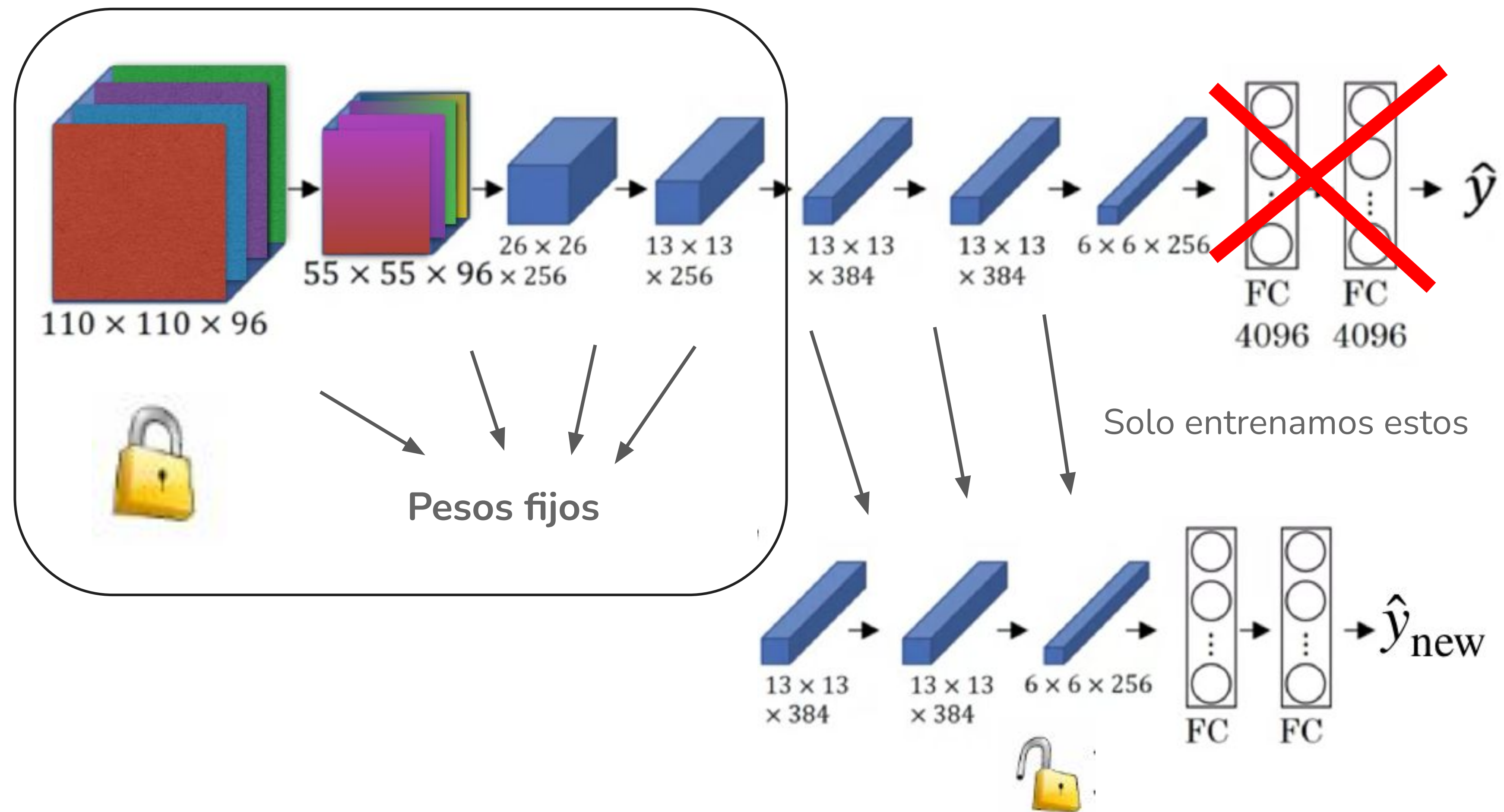
Transfer learning



Transfer learning



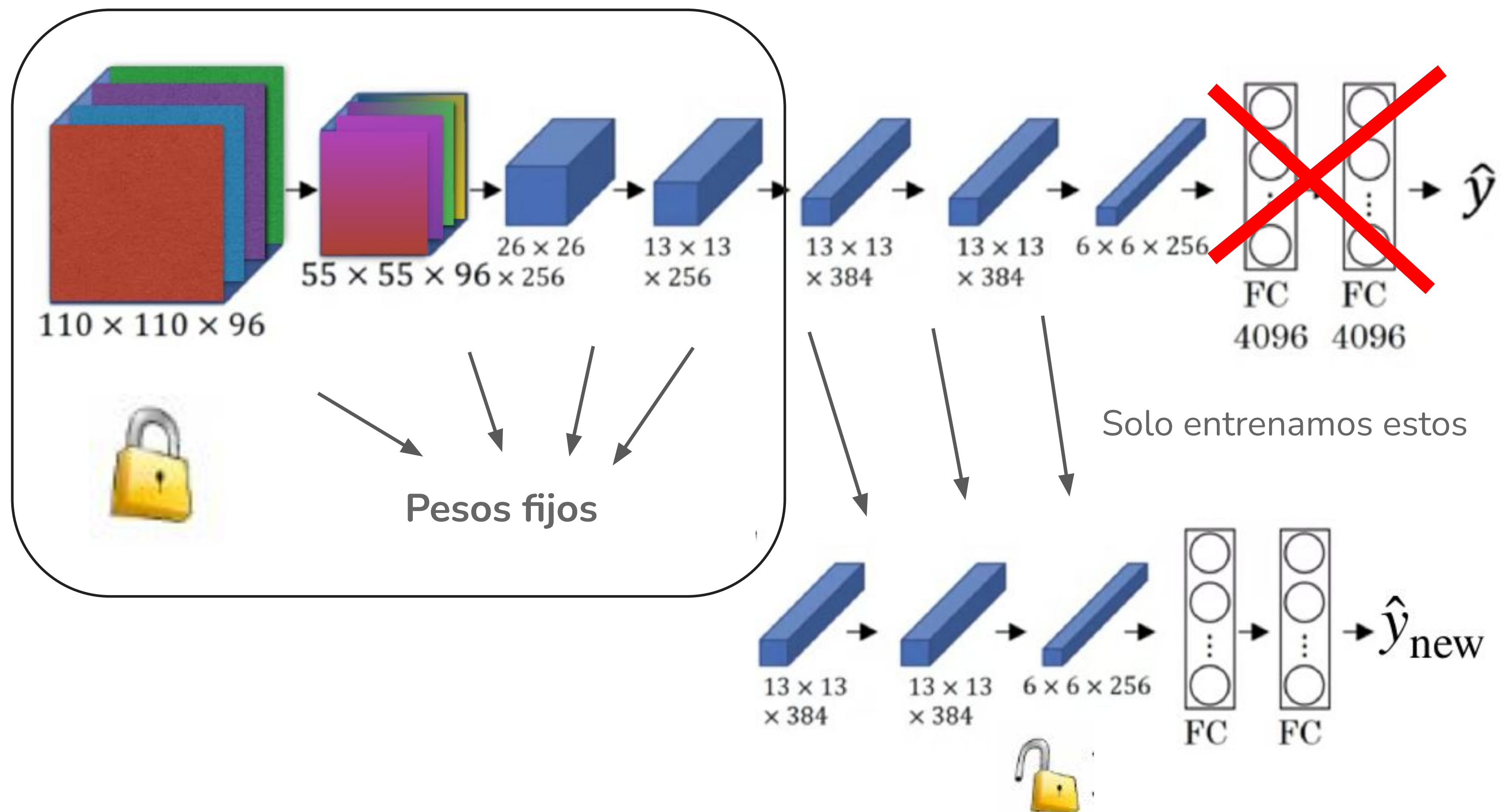
Transfer learning



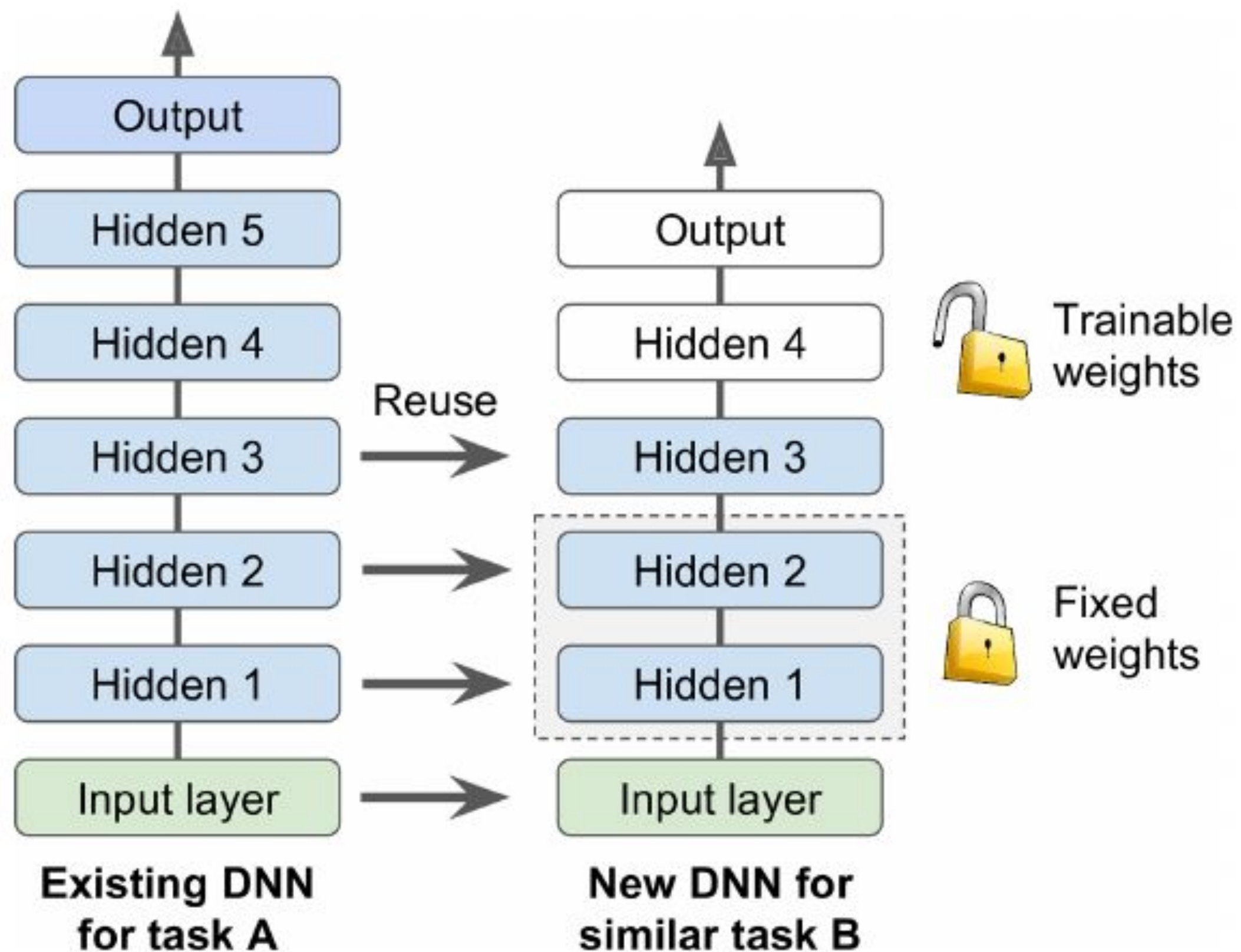
Transfer learning



...



Transfer learning



- La utilización de modelos pre entrenados para tareas más específicas es una gran forma de ahorrar datos y recursos.
- No solo se puede cambiar la capa de salida, se puede hacer todo un modelo nuevo reutilizando lo que nos sirva.
- Visión por computadora multipropósito: ResNet-50, VGG-16, VGG-19, Xception, Inception (v3, v4), MobileNet
- Object Detection: COCO, YOLO
- Procesamiento del lenguaje natural
- Multipropósito: ULMFiT, Transformer, BERT, GPT-2,3,...
- Word-Embeddings: Word2Vec, ELMo, Flair, ...

Imagen del Gerón

Transfer learning en Keras

- Usar capas (y pesos) de otro modelo descartando la capa final y agregando otra/

```
model_A = keras.models.load_model("my_model_A.h5")
model_B_on_A = keras.models.Sequential(model_A.layers[:-1])
model_B_on_A.add(keras.layers.Dense(1, activation="sigmoid"))
```

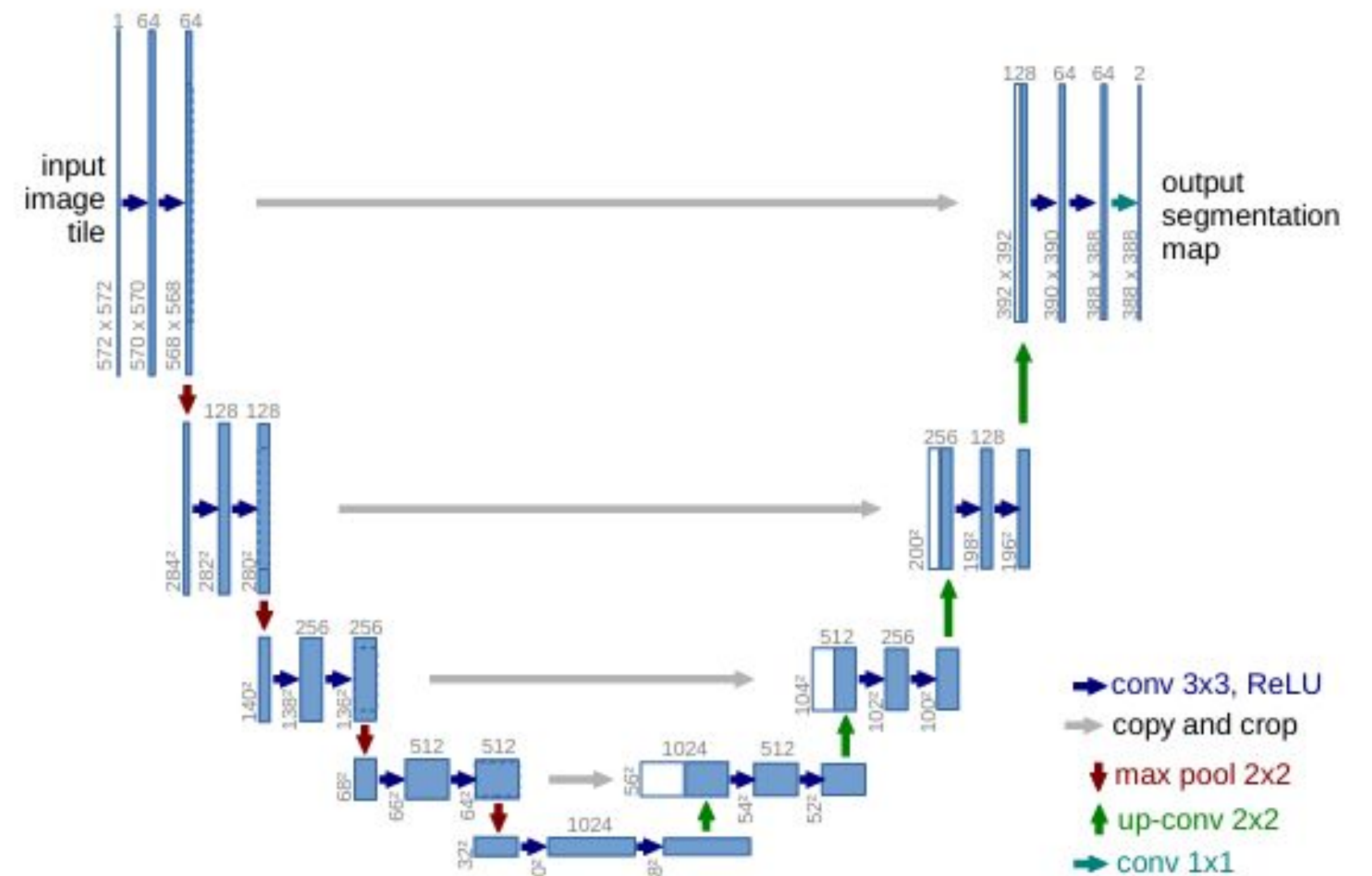
- Fijar los pesos de algunas de las capas (acá, todas, menos la última) antes de compilar/

```
for layer in model_B_on_A.layers[:-1]:
    layer.trainable = False
model_B_on_A.compile(loss="binary_crossentropy", optimizer="sgd",
                    metrics=["accuracy"])
```

- Entrenar, descongelar algunas capas, entrenar, etc.
- Hay cientos de opciones ya cerradas que hacen todo esto aún más transparente, por ejemplo https://github.com/qubvel/segmentation_models.

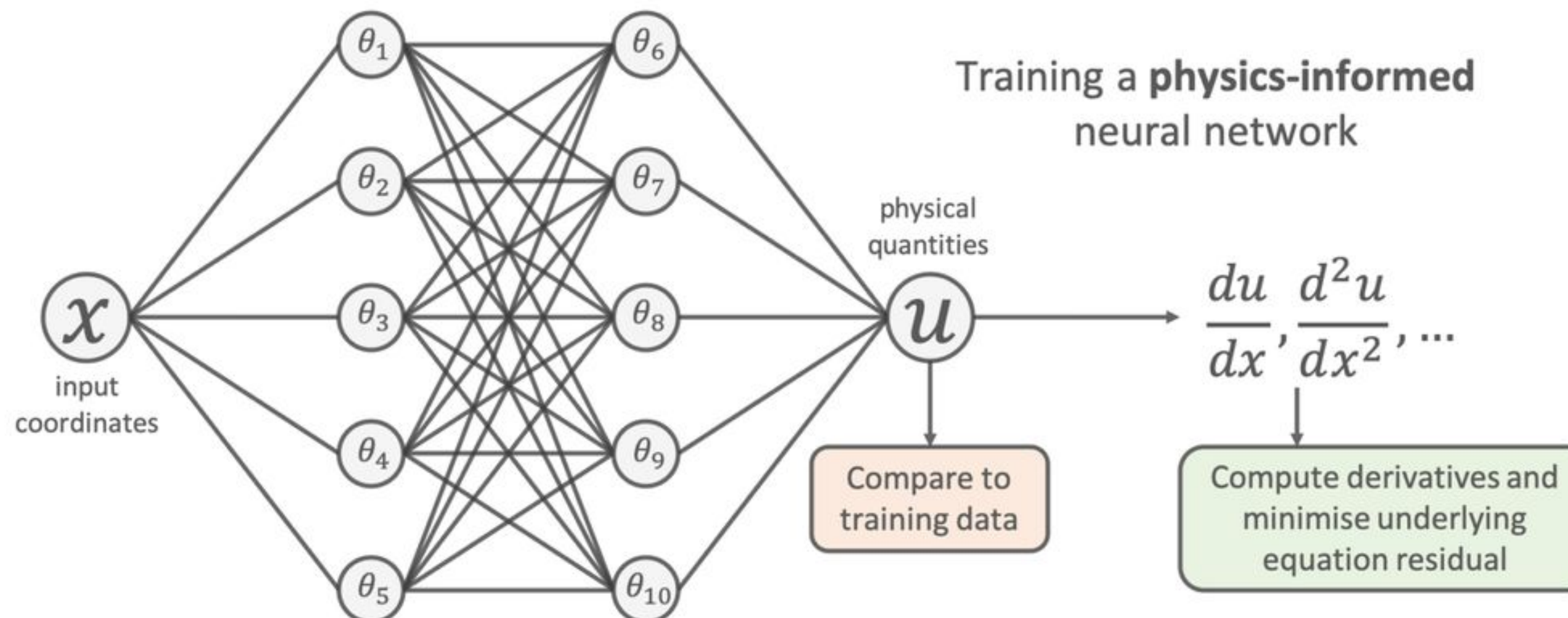
U-Net

- Arquitectura para segmentación semántica.
- *Totalmente convolucional:*
 - Permite cualquier tamaño de entrada.
 - La salida es del mismo tamaño que la entrada.
- *Encoder/Decoder.*
 - Se pasa la información extraída por un cuello de botella, forzando a retener la información más importante para la generación.
- *Skip connections entre capas.*
 - Se transfieren características de alta resolución del encoder al decoder, ayudando a recuperar detalles espaciales perdidos durante el downsampling.
- Puede ser entrenado con pocas imágenes.
- U-Net: Convolutional Networks for Biomedical Image Segmentation
(<https://arxiv.org/abs/1505.04597>)



Physics-Informed neural networks

- Combina principios físicos y técnicas de aprendizaje automático para crear modelos más precisos, robustos y explicables. Se utiliza cuando hay sistemas físicos bien entendidos pero complejos, donde las leyes de la física ayudan a guiar el aprendizaje.
- Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. <https://arxiv.org/abs/1711.10561>.
- Physics Informed Machine Learning: High Level Overview of AI and ML in Science and Engineering ([link](#)).
- Physics-Informed Neural Networks (PINNs) - The Synergy of Data & Physics in Deep Learning ([link](#)).



Oficina Internacional de Energia Atómica (IAEA)

- **AI for Atoms** ([link](#)).
 - Sitio principal, plataforma de intercambio de conocimientos de la OIEA para la colaboración en aplicaciones de IA en el ámbito nuclear. Eventos, documentos guías y workshops (pocos...). Revisar el [Github](#) que tiene datos y código que les puede llegar a servir.
- **Artificial Intelligence for Accelerating Nuclear Applications, Science and Technology** ([link](#))
 - Detalla el estado del arte, los desafíos e identifica oportunidades para acelerar las aplicaciones nucleares, la ciencia y la tecnología usando inteligencia artificial.
- **Deployment of Artificial Intelligence Applications for the Nuclear Power Industry Considerations and Guidance** ([link](#)).
 - Consideraciones y guías para los estados miembros para el desarrollo de tecnologías de IA en el ámbito nuclear (versión *draft* de febrero de 2024).

Agencia para la Energía Nuclear (NEA)

- Task Force on Artificial Intelligence and Machine Learning for Scientific Computing in Nuclear Engineering ([link](#)).
 - Diseño de un benchmark para evaluar el desempeño de la IA/ML en la simulación multifísica de sistemas de reactores nucleares (Todavía hay tiempo de participar).
 - El objetivo es mejorar la capacidad predictiva de la IA/ML mediante una verificación y validación rigurosa, centrada en precisión, robustez y transparencia, para fomentar la aceptación de estas tecnologías por parte de los reguladores y las partes interesadas en el sector nuclear.

	2022		2023		2024		2025		2026	
	Q1-Q2	Q3-Q4	Q1-Q2	Q3-Q4	Q1-Q2	Q3-Q4	Q1-Q2	Q3-Q4	Q1-Q2	Q3-Q4
Phase 1 Draft Specifications										
Phase 1 Final Specifications										
Phase 1 Execution and Report										
Phase 2 Draft Specifications										
Phase 2 Final Specifications										
Phase 2 Execution and Report										

The end

