Visión por Computadora II - CEAI - FIUBA

Profesores:

- Cavalieri Juan Ignacio - juanignaciocavalieri@gmail.com
- Cornet Juan Ignacio - juanignaciocornet@gmail.com
- Seyed Pakdaman - khodadad.pakdaman@gmail.com

Programa de la materia

1. **Clase 1:** Introducción a problemas en visión por computadora. Clasificación. Arquitecturas AlexNet Y VGGNet. Data Augmentation.
2. **Clase 2:** Residual Networks, Arquitecturas ResNet. Arquitecturas Inception. Transfer Learning. Presentación de TP final.
3. **Clase 3**: Localización y detección de objetos. Algoritmo de Sliding Windows. mAP. Métodos de dos etapas: R-CNN, Fast R-CNN, Faster R-CNN. Métodos de una etapa: YOLO, SSD, etc.
4. **Clase 4:** Segmentación de imágenes. Segmentación semántica: U-Net. Segmentación de instancias: Mask R-CNN.
5. **Clase 5:** Neural Style Transfer. GradCAM. Redes generativas y aplicaciones.
6. **Clase 6:** Aprendizaje no supervisado. Visual transformers.
7. **Clase 7:** Ejemplo de aplicación a cargo de Seyed Pakdaman.
8. **Clase 8:** Presentación de trabajos integradores.

Dinámica de las clases

- Segmento teórico
- Descanso
- Segmento práctico

Requisitos para la aprobación

- Trabajo práctico integrador
  - Se puede hacer en grupos de 2 o 3 personas.
  - Se presenta en la última clase.

Herramientas

- Lenguaje: Python
- Librerías comunes: PyTorch, Numpy, Pandas, Matplotlib
- Entornos de trabajo: Colab, Kaggle, SageMaker Studio Lab, etc.
- Github (Link)
- Slack: Canal #vpc2

Bibliografía

- Computer Vision: Algorithms and Applications, 2nd ed. : R. Szeliski. Link
- Deep Learning: I. Goodfellow, J. Bengio, A. Courville. Link
- Dive into Deep Learning: A. Zhang et al. Link
- Papers

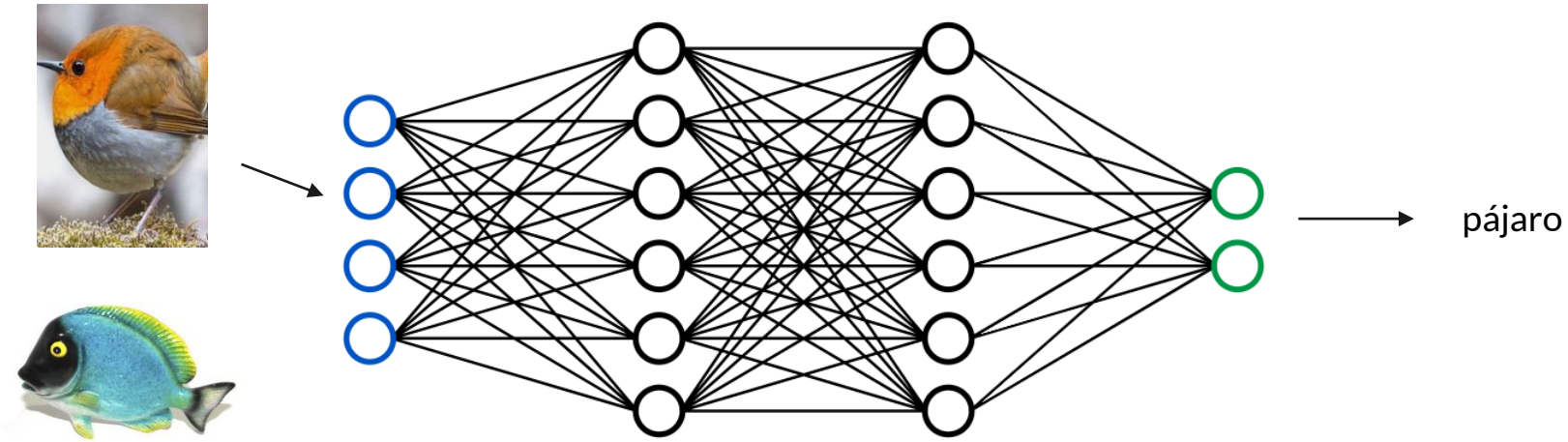
Para profundizar, no es necesaria para las clases.

Primera clase

- Tareas básicas en visión por computadora resueltas mediante redes neuronales:
  - Clasificación de imágenes
  - Detección de objetos
  - Segmentación
  - Otras: Transferencia de estilo, generación de imágenes, etc.
- Clasificación:
  - AlexNet
  - VGGNet
- Data Augmentation
- Implementación en Pytorch.

Clasificación (Image Classification)

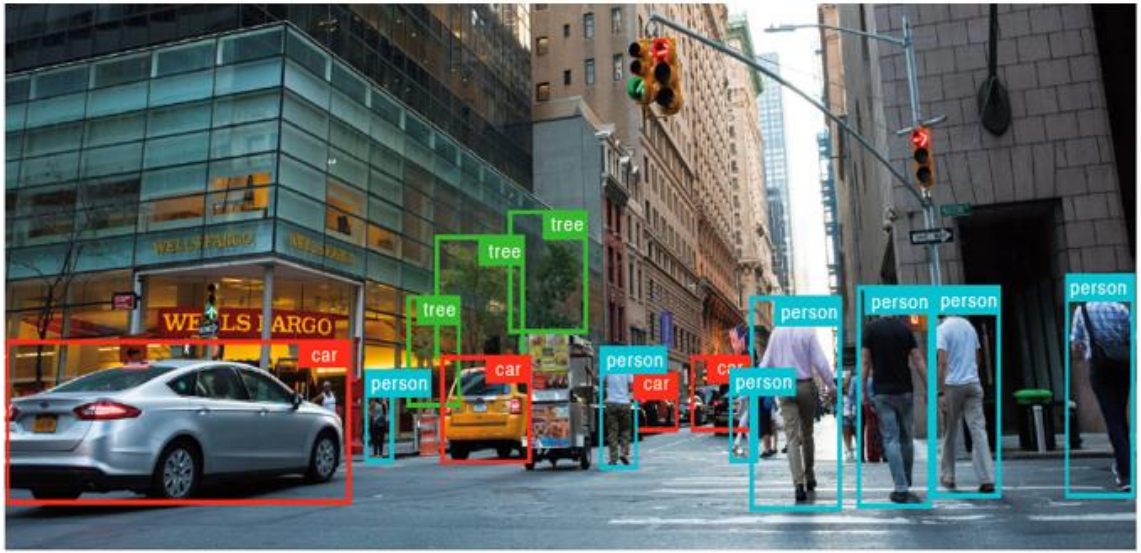
En este tipo de problemas se trata de asociar una imagen dada a una o varias clases de un conjunto predefinido de las mismas. La salida del modelo será entonces, la clase predicha.



Datasets clásicos: MNIST, Fashion-M NIST, CIFAR-10, ImageNet, etc.

Detección de objetos (Object detection)

Existen métodos de dos etapas (R-CNN, Fast R-CNN, Faster R-CNN), que tienen mejor precisión, y métodos de una etapa (YOLO, SSD, RetinaNet) que tienen mejor velocidad de inferencia.



Datasets: COCO, Pascal VOC, Imagenet, Open Images, etc

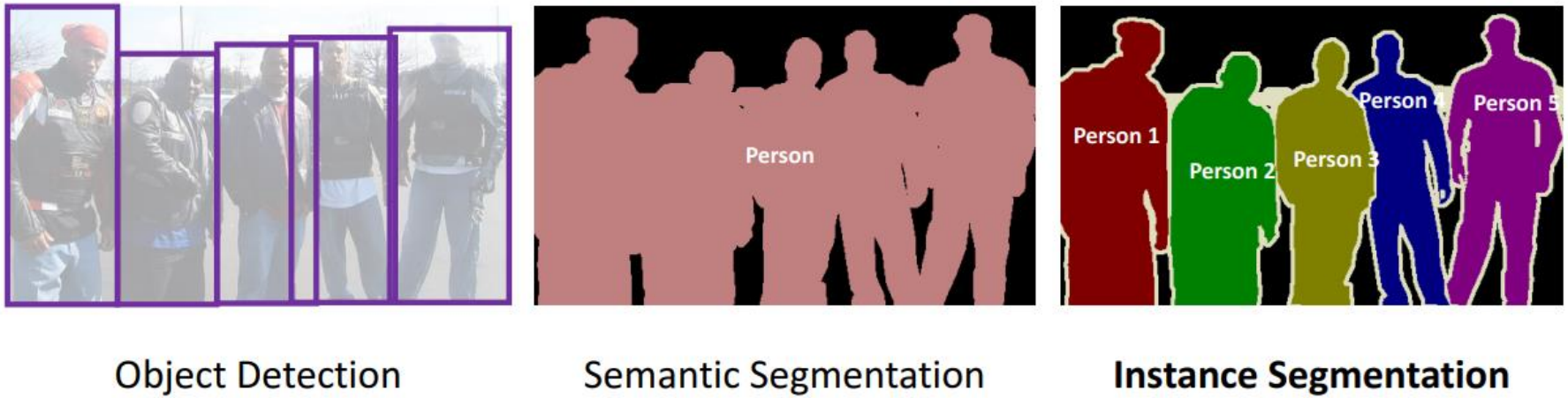
Segmentación Semántica (Semantic Segmentation)

La segmentación semántica consiste en asignarle una clase a cada uno de los pixeles de una imagen. Esto es útil para poder separar diversas zonas de la misma (grupos de pixeles) en regiones de interés que comparten las mismas características, es decir, pertenecen a la misma clase.

Datasets: COCO, Pascal VOC, Open Images, Cityscapes, etc

Segmentación de Instancias (Instance Segmentation)

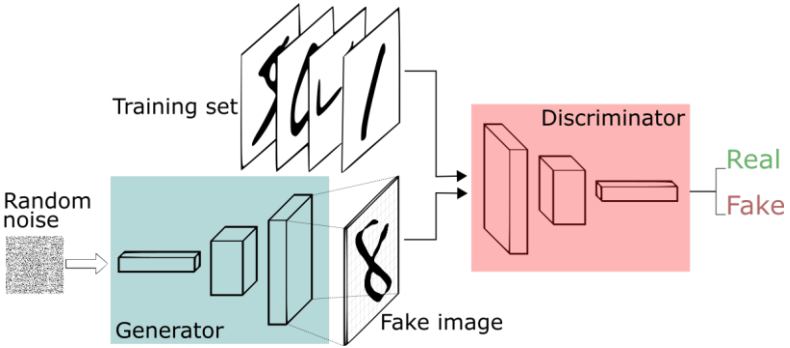
Se la puede considerar como una combinación entre Segmentación Semántica y Detección de Objetos ya que, además de segmentar la imagen por clases se realiza una diferenciación entre los distintos objetos que pertenecen a una misma clase.



Datasets: COCO, Pascal VOC, Open Images, Cityscapes, etc

Otras...

El Deep Learning aplicado a las imágenes ha permitido desarrollar algoritmos que generan imágenes realistas, hacen transferencia de estilo o traducen texto a imágenes, entre otras.



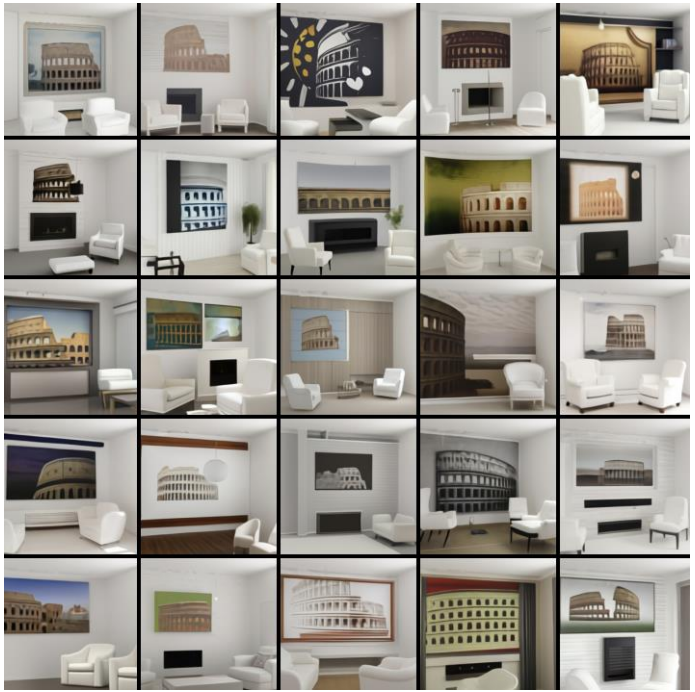
Otras...

DALL.E 2: Puede crear imágenes realistas a partir de una descripción en lenguaje natural.

En el ejemplo de la imagen la descripción fue:

“a living room with two white armchairs and a painting of the collosseum. the painting is mounted above a modern fireplace”

https://openai.com/dall-e-2/



Otras...

Stable Diffusion es similar a DALL.E 2 pero desarrollada por un grupo independiente y es de código abierto!

En el ejemplo de la imagen la descripción fue:

“a photograph of an astronaut riding a horse”

https://stability.ai/blog/stable-diffusion-public-release

https://github.com/CompVis/stable-diffusion

https://colab.research.google.com/github/huggingface/notebooks/blob/main/diffusers/stable_diffusion.ipynb

Otras...

SAM (Segment Anything Model) es un modelo de segmentación capaz de generalizar sobre objetos y clases sobre las que no fue entrenado. Fue desarrollado por Meta.

Website: https://segment-anything.com/ y blog: link

Otras...

Sora es un modelo desarrollado por OpenAI que puede transformar descripciones de texto en videos super realistas.

Prompt:
A movie trailer featuring the adventures of the 30-year-old space man wearing a red wool knitted motorcycle helmet, blue sky, salt desert, cinematic style, shot on 35 mm film, vivid colors.

Website: https://openai.com/index/sora/ y blog: link

Algunos desafíos:

Supongamos que tenemos imágenes **RGB** de **1280 x 720 píxeles**, entonces la cantidad de atributos de entrada a la red será:

1280 x 720 x 3 = **2.764.800 entradas!**

Luego, si consideramos una primera capa de nuestra red con **1000** neuronas conectadas con la entrada, mediante una capa densa, la cantidad de parámetros en esa capa será:

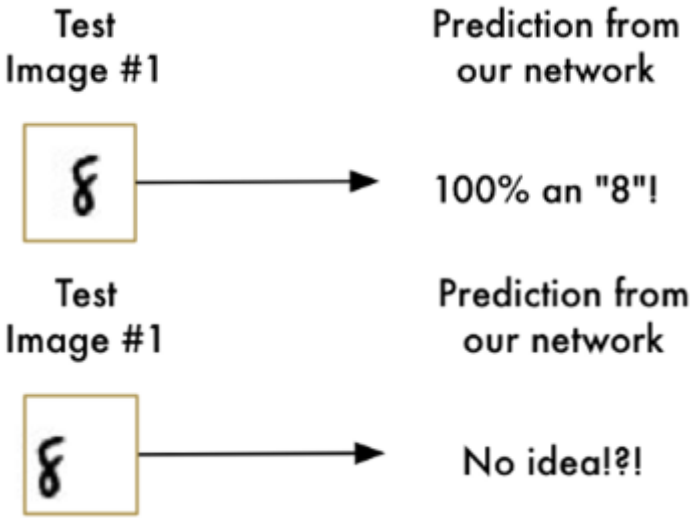
cantidad de entradas x cantidad de neuronas = cantidad de parámetros
2.764.800 x 1000 = **~2.764 millones de parámetros!!**

Problemas:

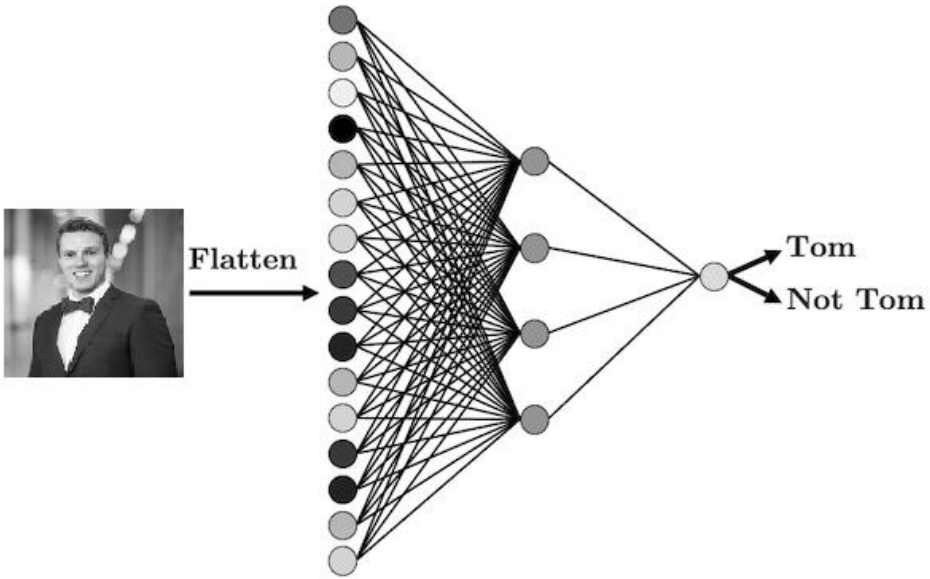
- Enorme necesidad de memoria y poder de cálculo
- La enorme cantidad de parámetros hace que sea más fácil sobreentrenar el modelo.

Algunos desafíos:

Invariancia a la traslación



Características localizadas



Algunos desafíos:

Viewpoint variation

Scale variation

Deformation

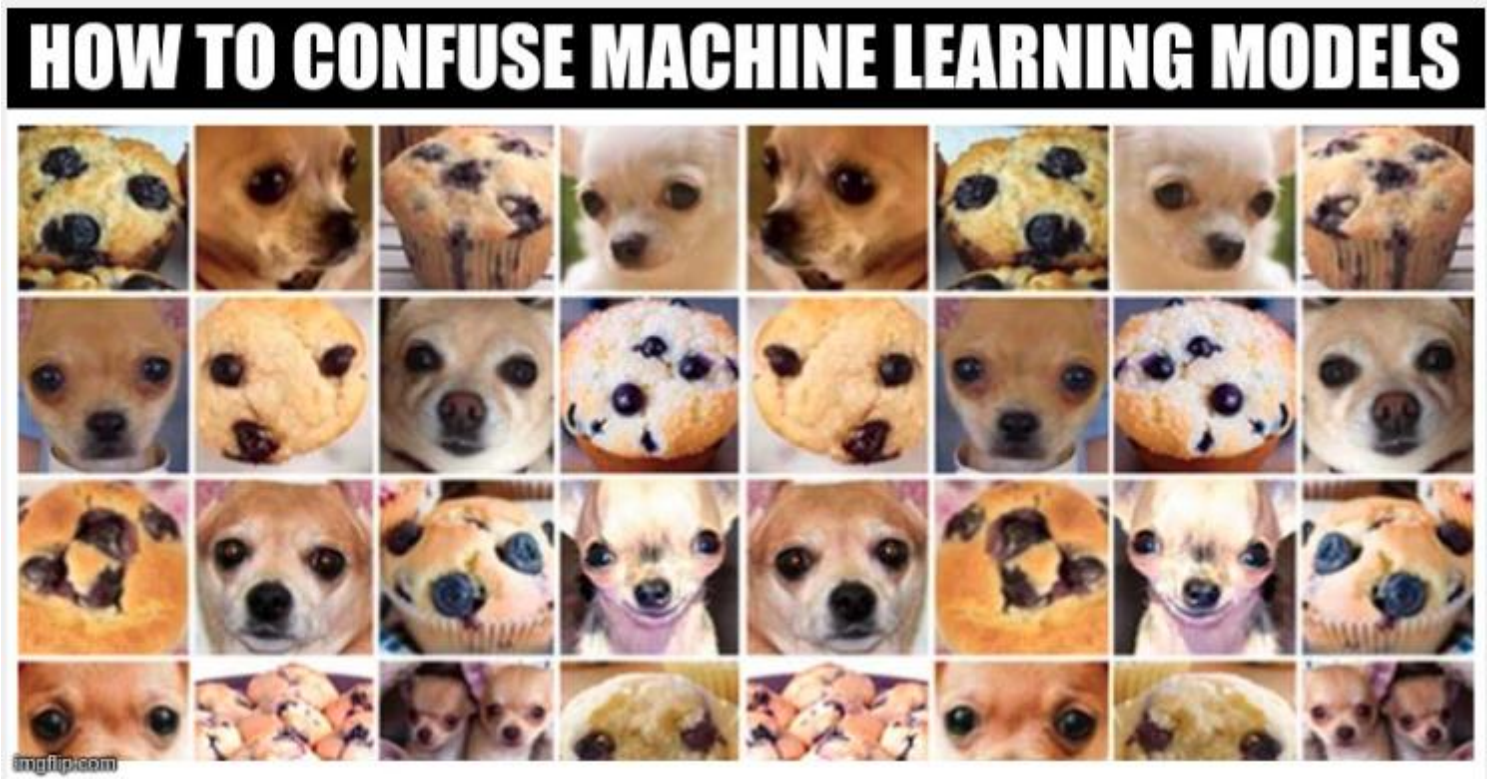
Occlusion

Illumination conditions

Background clutter

Intra-class variation

Algunos desafíos:



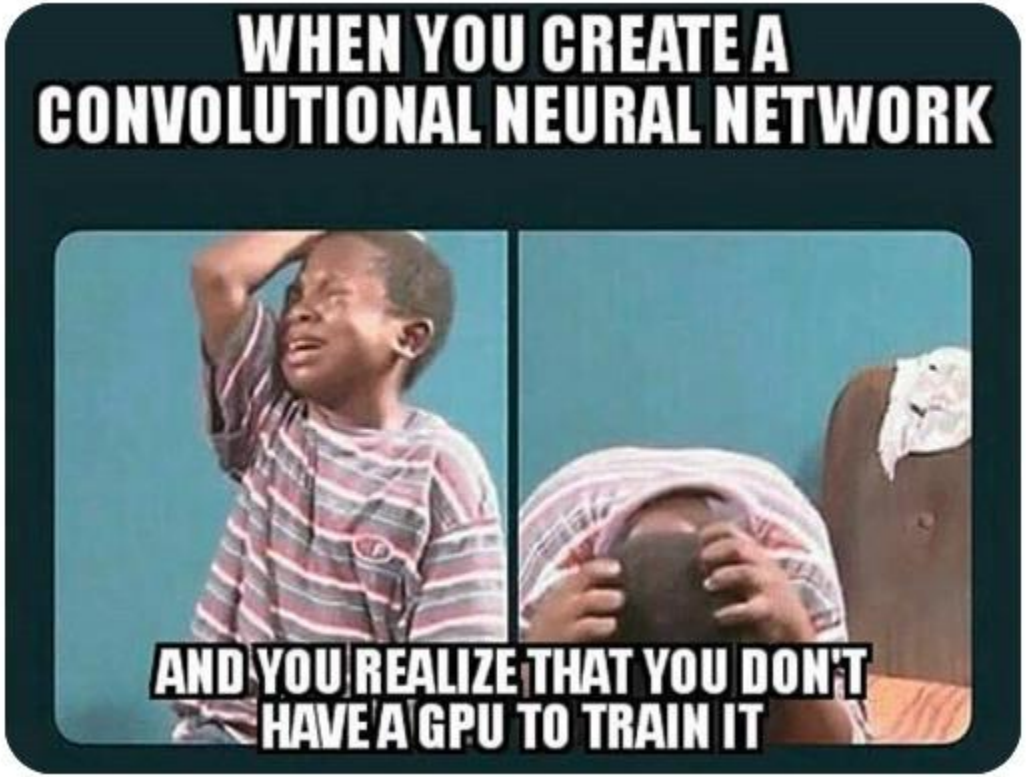
Solución: Redes Convolucionales

Las características de los tipos de capas que conforman una red neuronal convolucional nos permiten extraer características (o features) de los datos de entrada, de forma más localizada (conexiones esparsas), con más robustez frente a las variaciones en los datos y reduciendo el número de parámetros necesarios.

- Capas Convolucionales (CONV)
- Capas de Pooling (POOL)

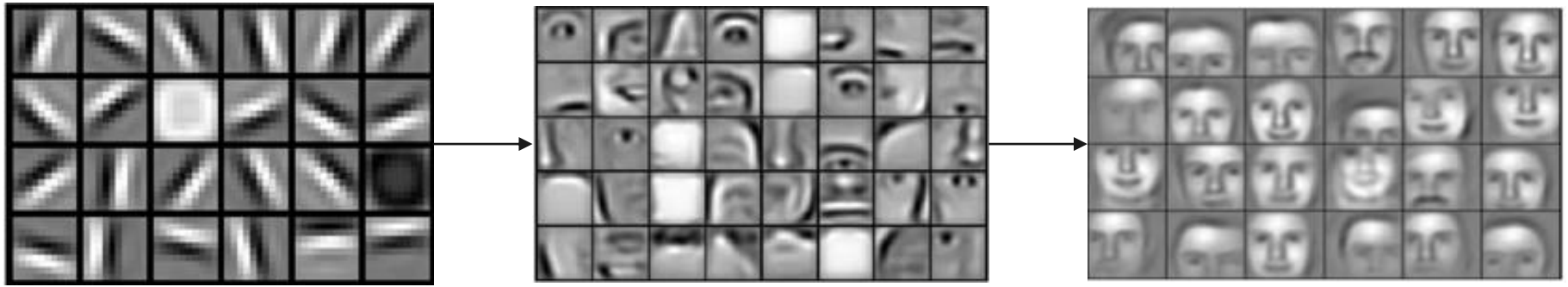
Paper: Link

Redes Convolucionales



Aprender de los datos: Capa Convolucional

- Capa detectora de bordes
- Capa detectora de partes de caras
- Capa detectora de caras enteras



Capa convolucional
Resumen dimensiones

Dado un volumen de entrada $n \times n \times n_C$ y una capa de n_f filtros de $f \times f \times n_C$, las dimensiones del volumen de salida de la capa convolucional serán:

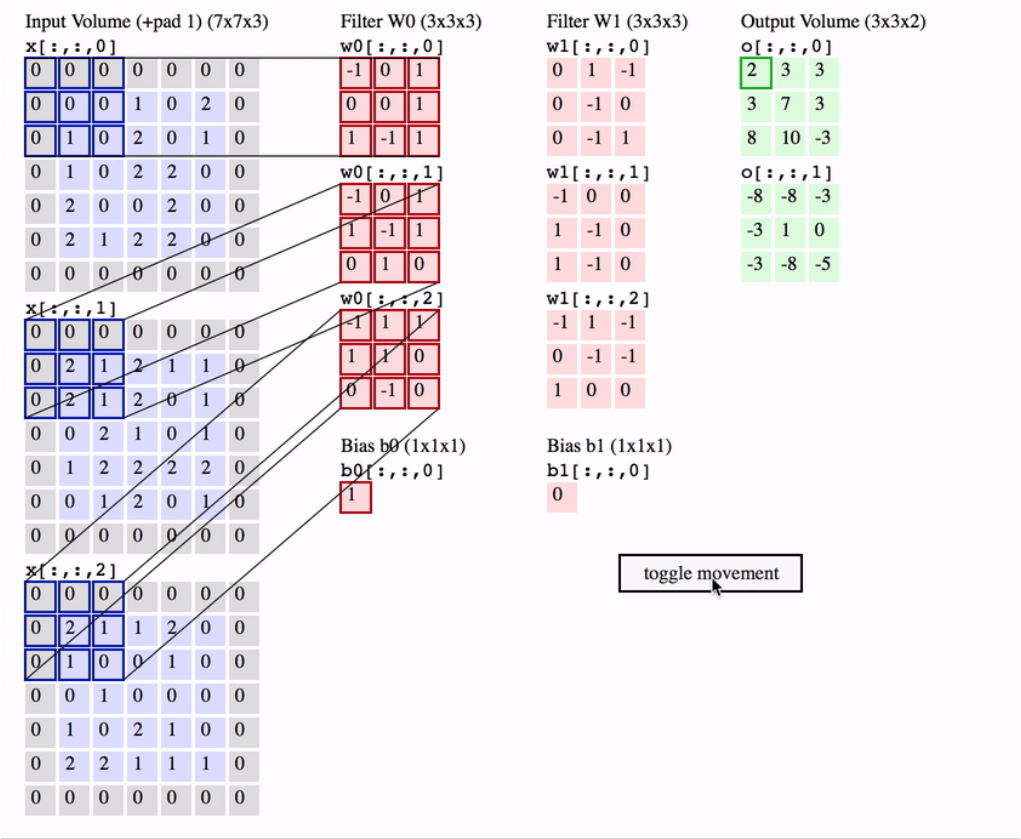
$$\left\lfloor \left(\left(\frac{n - f + 2p}{s} \right) + 1 \right) \right\rfloor \times \left\lfloor \left(\left(\frac{n - f + 2p}{s} \right) + 1 \right) \right\rfloor \times n_f$$

Luego, la cantidad de parámetros entrenables dentro de dicha capa convolucional puede calcularse como:

$$f \times f \times n_C \times n_f + n_f$$

La suma del ultimo n_f se debe a los parámetros de bias.

Tener en cuenta que el valor de n_C de la capa siguiente será igual a la cantidad de filtros convolucionales de la capa actual.



Capa Pooling
Resumen dimensiones

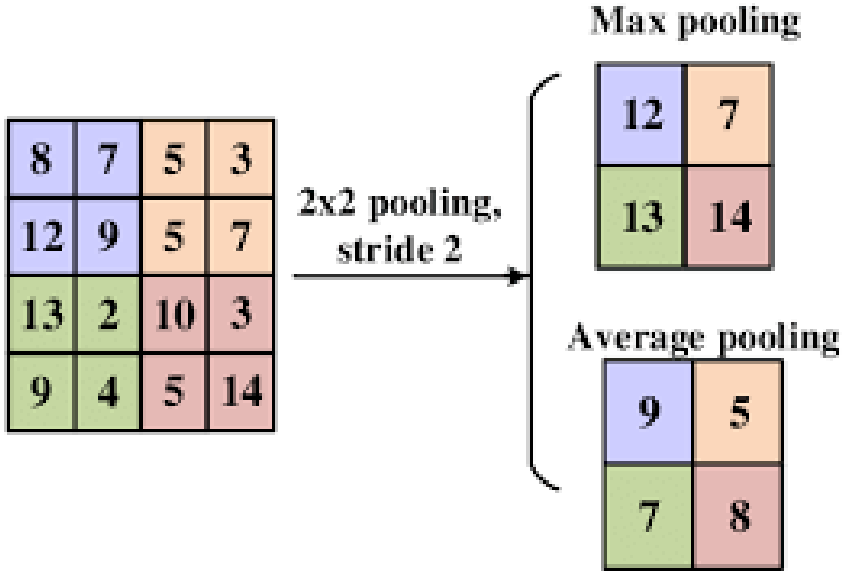
Dado un volumen de entrada de

$$n_H \times n_W \times n_C$$

y una capa de pooling con tamaño de filtro f y stride s. Las dimensiones de la salida serán

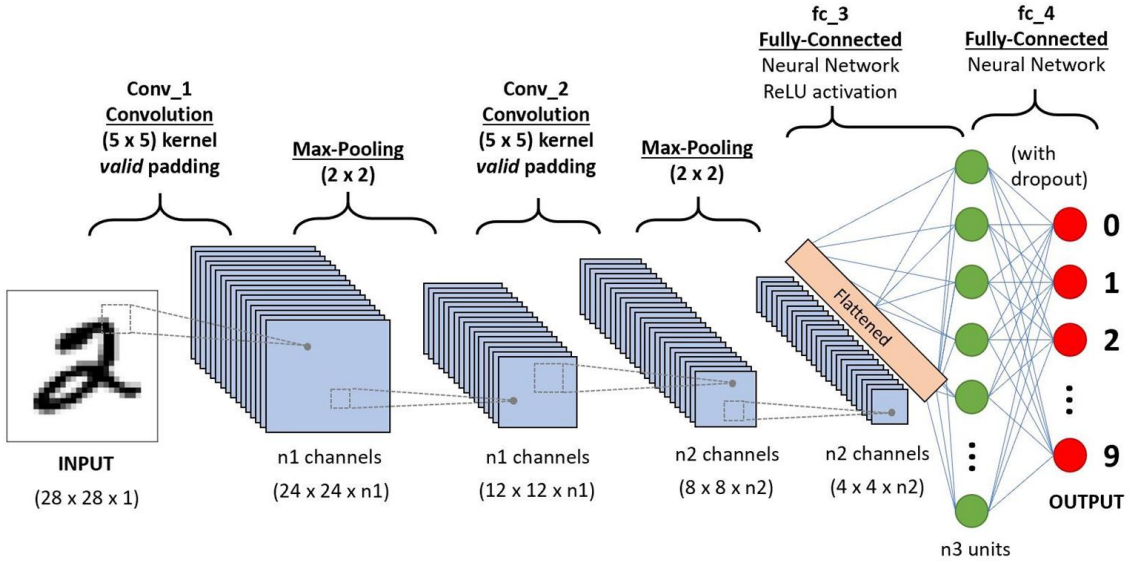
$$\left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W - f}{s} + 1 \right\rfloor \times n_C$$

Las capas de pooling no aportan parámetros entrenables ni modifican la cantidad de canales del volumen de entrada.



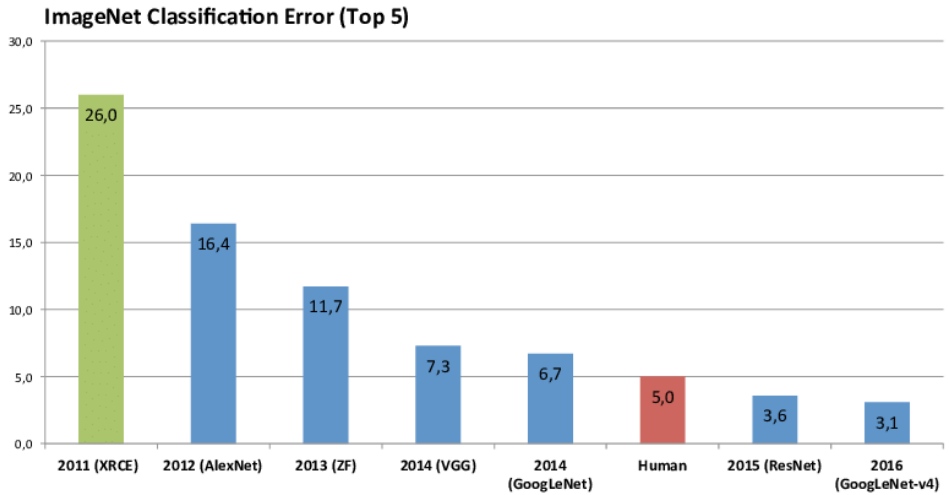
Clasificación de imágenes

El esquema de las redes para clasificación y, en general, para procesamiento de imágenes, consta de una primera etapa con capas convolucionales y de pooling para la extracción de características, seguida de capas densas para la clasificación final.



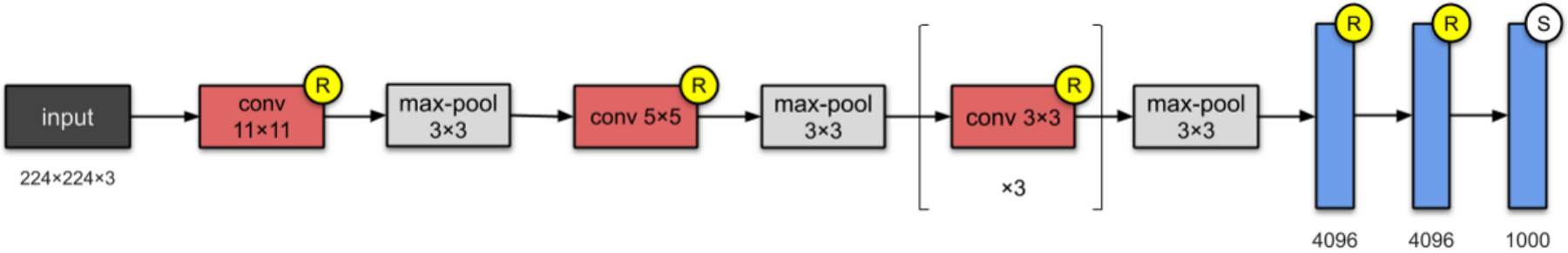
Arquitecturas clásicas: AlexNet

En 2012 causó un gran impacto por obtener un puntaje significativamente mayor que el segundo puesto en ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), basando su modelo en capas convolucionales y de pooling. A partir de ese momento todos los ganadores comenzaron a ser redes convolucionales profundas.



Krizhevsky, et al., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Link

AlexNet



Illustrated: 10 CNN Architectures. Link

Características de AlexNet

Entre las novedades introducidas en esta arquitectura encontramos:

- Activaciones ReLU (Rectified Linear Units)
- Uso de múltiples GPUs para entrenar el modelo
- Dropout
- Local Response Normalization (no tan usado hoy en día)
- Capas Pool con ventanas superpuestas
- Data Augmentation
- 60 M de parámetros

Arquitecturas clásicas: VGGNet

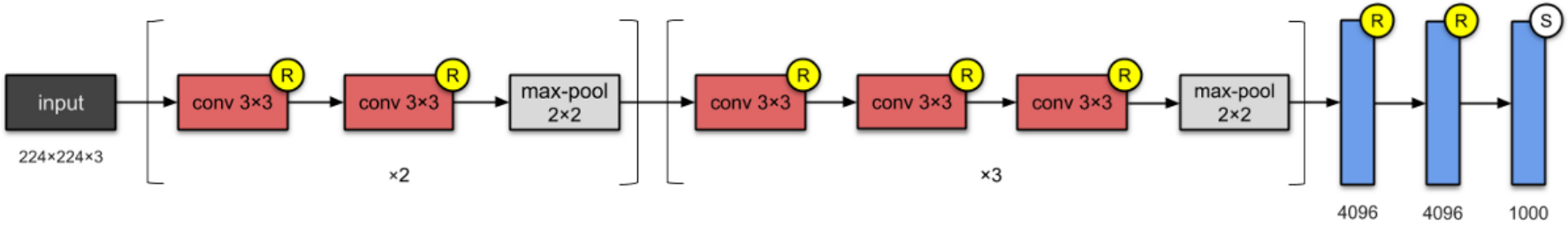
Las redes VGG-16 y VGG-19 fueron presentadas en 2014 obteniendo el primer y segundo puesto en las tareas de localización y clasificación de la competencia ILSVRC, respectivamente. Estas redes, si bien están basadas en muchos de los principios introducidos por AlexNet, cuentan con algunas características destacables:

- Utiliza tamaños de filtro de 3x3 a lo largo de toda la red.
- Agrega más capas convolucionales.
- Para reducir la dimensionalidad solo se emplearon capas de Max-Pooling. Todas las convolucionales son con stride igual a 1.
- Entrenamiento con multi escalado de imágenes (Multi-Scale Training).
- 138M y 144M de parámetros.

Simonyan, et al., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. Link

VGGNet

VGG-16



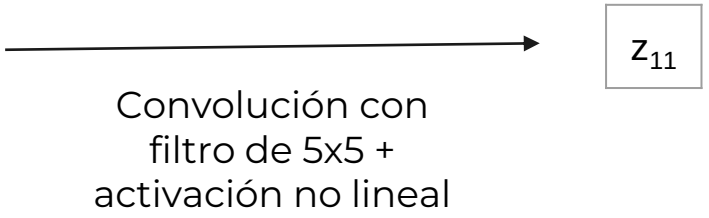
Para el caso de VGG-19, los bloques de 3 capas convolucionales tenían 4 capas, seguidas de un max-pooling.

Illustrated: 10 CNN Architectures. Link

VGGNet: Tamaño de filtros

¿Cuántas formas tengo de procesar lo que se encuentra en una región de 5x5 pixeles?

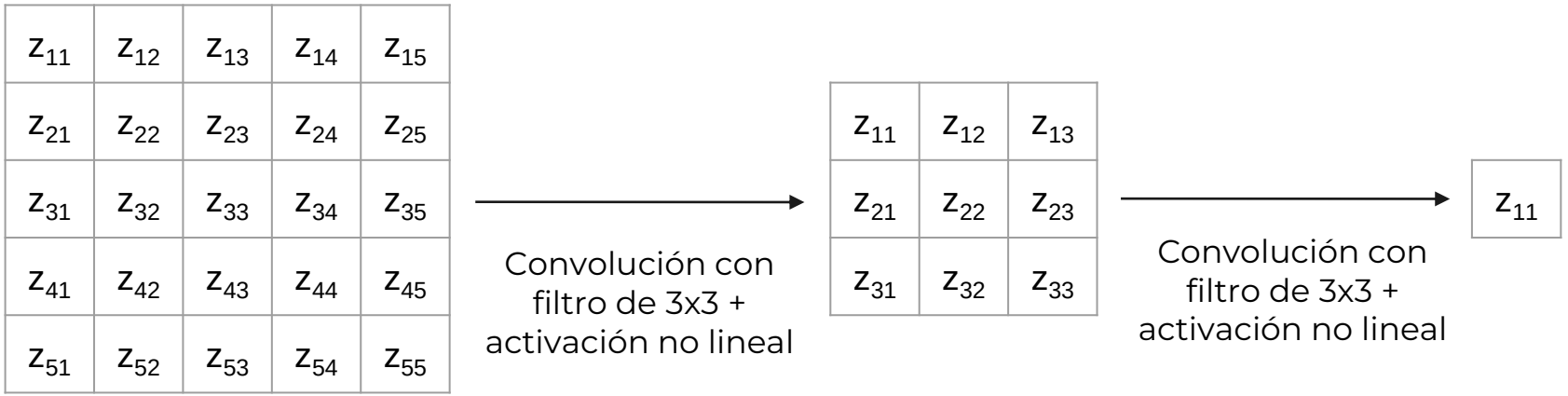
z_{11}	z_{12}	z_{13}	z_{14}	z_{15}
z_{21}	z_{22}	z_{23}	z_{24}	z_{25}
z_{31}	z_{32}	z_{33}	z_{34}	z_{35}
z_{41}	z_{42}	z_{43}	z_{44}	z_{45}
z_{51}	z_{52}	z_{53}	z_{54}	z_{55}



- 1 convolución con tamaño de filtro 5x5
- Presenta 1 función con no-linealidad
- Requiere 25 parámetros

VGGNet: Tamaño de filtros

¿Cuántas formas tengo de procesar lo que se encuentra en una región de 5x5 pixeles?



- 2 convoluciones con tamaño de filtro 3x3 y stride 1
- Presenta 2 funciones con no-linealidad
- Requiere 18 parámetros

VGGNet: Tamaño de filtros

A diferencia de redes como AlexNet, VGG utiliza capas con tamaños de filtro más pequeños. Dado que utilizar más cantidad de capas con filtros más pequeños permite equiparar el **campo receptivo** (Receptive Field) de filtros mayores, se presentan las siguientes ventajas:

- Mayor posibilidad de utilizar no-linealidades, lo cual facilita la detección certera de las features.
- Menor cantidad de parámetros necesarios.
- Convergencia mas rapida.
- Menor probabilidad de sobreentrenamiento.

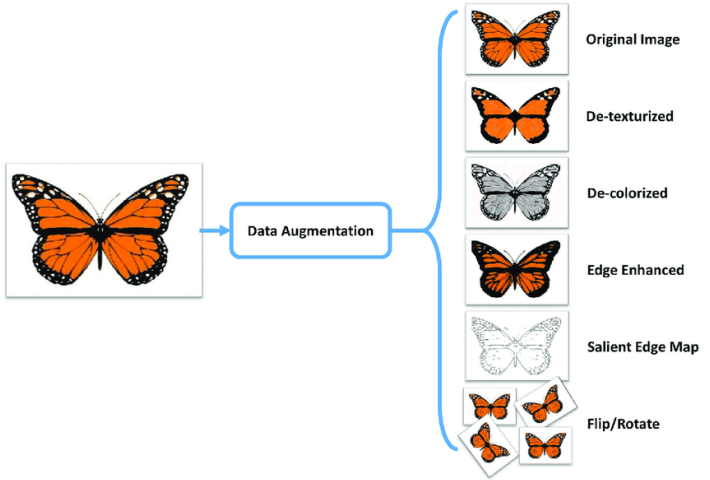
Sin embargo, utilizar redes con cada vez más capas no siempre es lo mejor y trae problemas asociados...

Data Augmentation

¿Qué es?

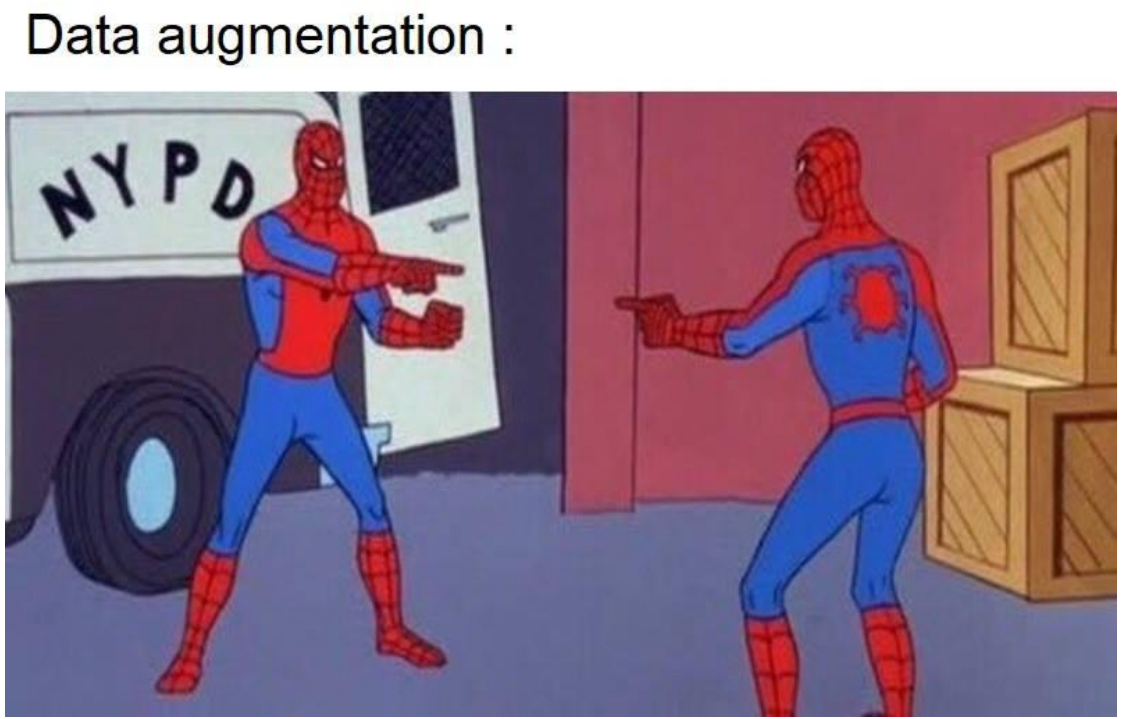
Consiste en generar “nuevos” ejemplos de datos de entrenamiento, para darle más variabilidad a nuestro conjunto de datos. Los nuevos datos pueden generarse a partir de los ya existentes, mediante distintos tipos de transformaciones, o pueden sintetizarse para crear ejemplos completamente nuevos.

No es una herramienta exclusiva de computer vision, se puede aplicar en otros dominios!



Shorten and Khoshgoftaar, 2019. A survey on Image Data Augmentation for Deep Learning link

Data Augmentation



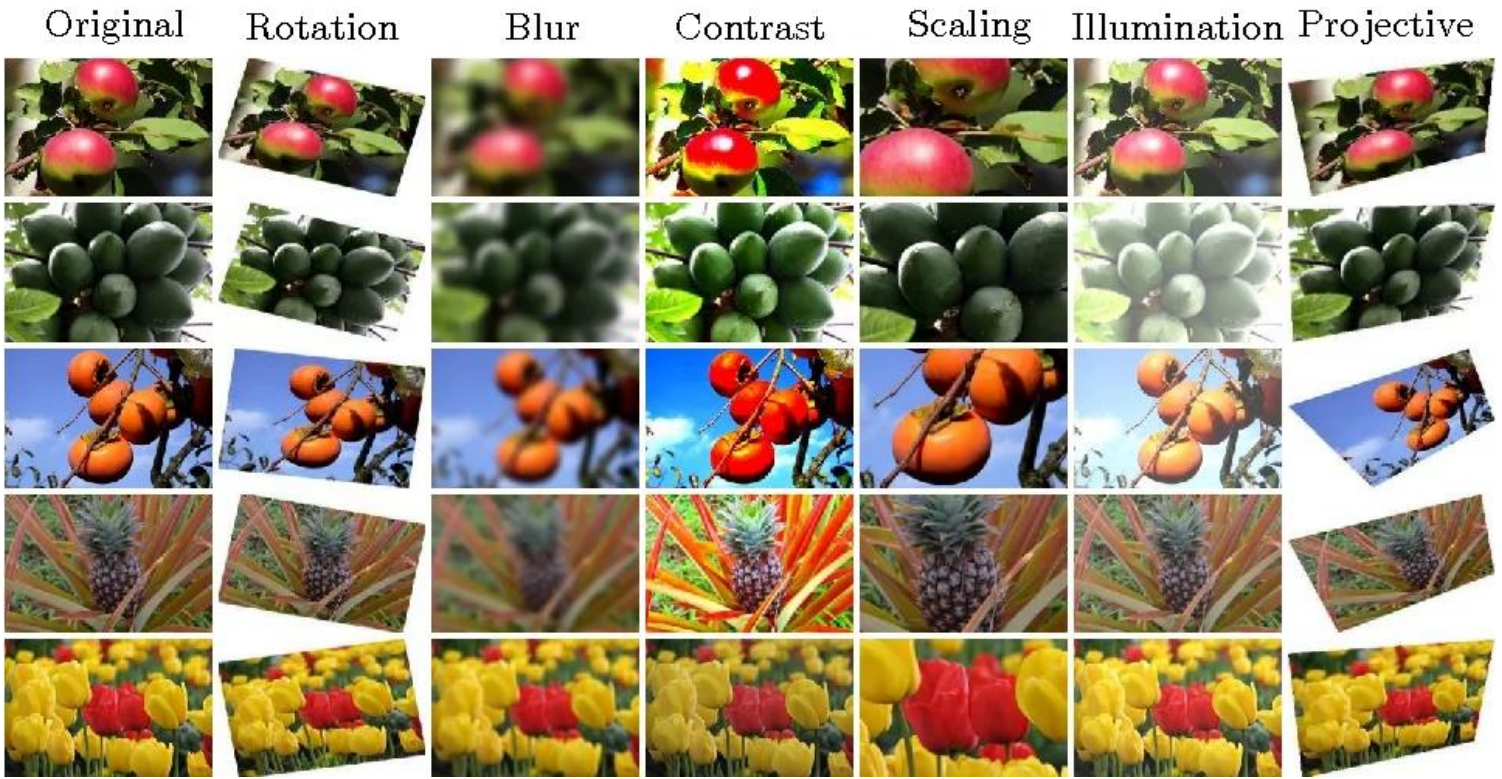
Data Augmentation

¿Para que se utiliza?

1. **Incrementar el tamaño del conjunto de datos:** Indispensable cuando el conjunto de datos es pequeño y cuando es costoso obtener o etiquetar nuevos datos.
2. **Evitar el sobreentrenamiento:** Busca mejorar la capacidad de los modelos de generalizar sobre los datos.
3. **Permite balancear las clases del conjunto de datos:** Compensa las variaciones en la distribución de los datos.
4. **Mejorar las métricas del modelo:** Enrobustece al modelo frente a variaciones en la distribución de los datos de validación o testeo.

Shorten and Khoshgoftaar, 2019. A survey on Image Data Augmentation for Deep Learning link

Transformaciones básicas

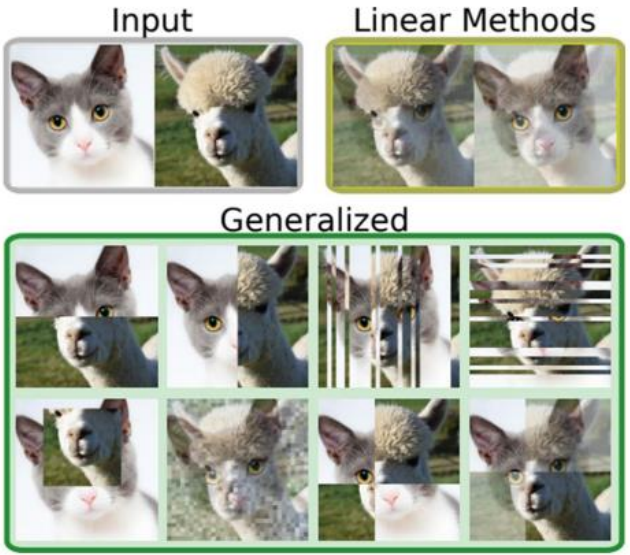


Transformaciones de color

Más Transformaciones

Mezcla de imágenes

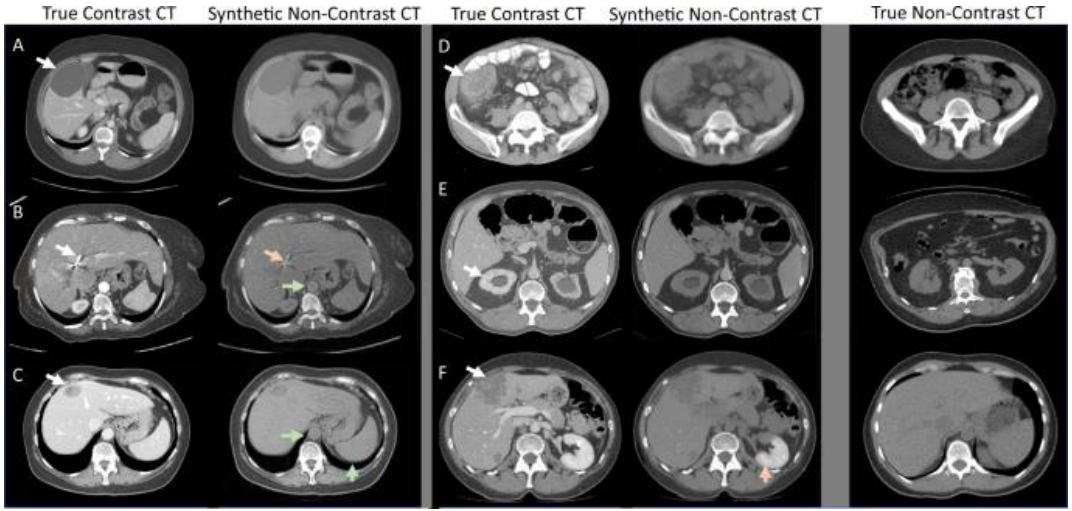
La mezcla de imágenes resulta ser una práctica contraintuitiva desde el punto de vista humano pero hay estudios que demuestran su funcionamiento en la clasificación de imágenes.



Hiroshi Inoue, 2018. Data Augmentation by Pairing Samples for Images Classification link

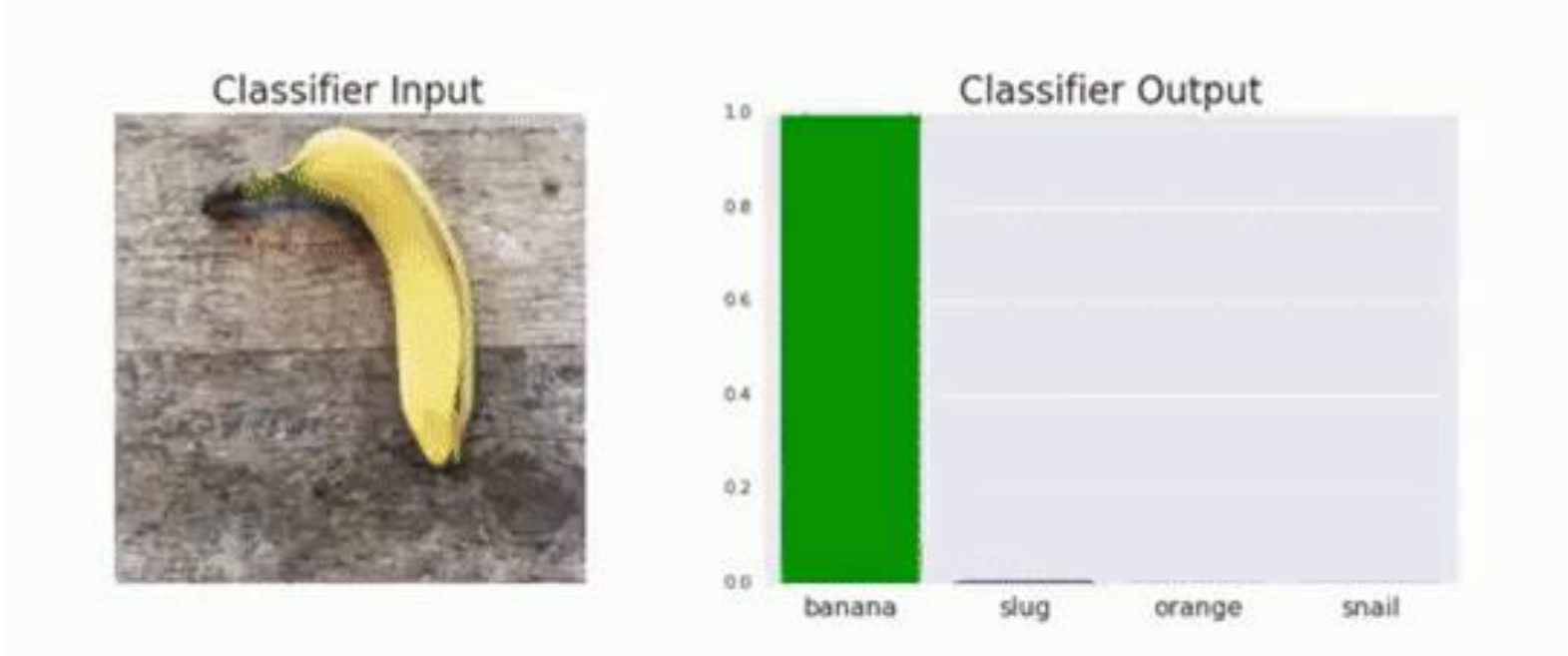
Transformaciones basadas en Deep Learning

Una posible técnica es utilizar redes neuronales para generar imágenes que formen parte del conjunto de entrenamiento. Utilizando Cycle-GAN se pueden generar imágenes de tomografías realistas.



Sandfort et al., 2019. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks link

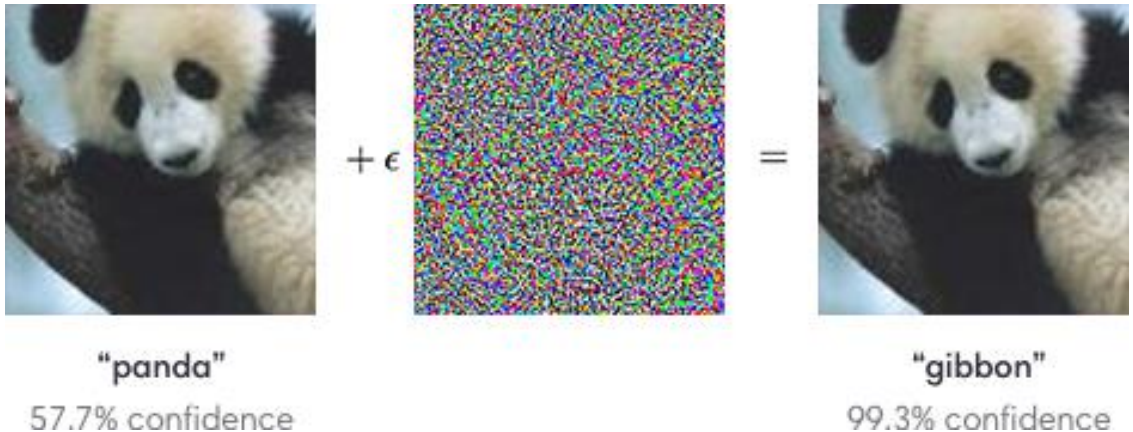
Ataques a las redes neuronales



En muchos casos, es factible engañar a este tipo de redes neuronales, obteniendo predicciones erróneas sobre imágenes que, a simple vista, tienen otro contenido.

Ataques adversarios

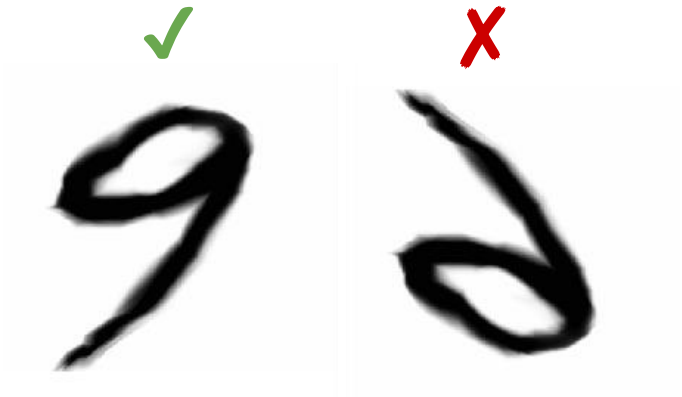
El entrenamiento adversario puede ayudar a encontrar posibles técnicas de aumentación que ayuden a mejorar los puntos débiles de las redes convolucionales.
Son muy importantes cuando las aplicaciones de estos modelos están relacionadas con identificación de personas, por ejemplo.



Malos ejemplos de Data Augmentation

Se debe tener cuidado con las transformaciones elegidas porque pueden modificar el valor de las etiquetas según sea el conjunto de datos que estamos utilizando.

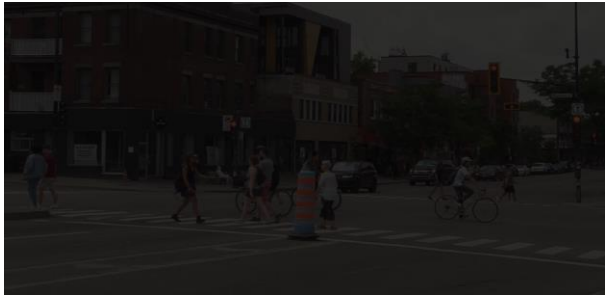
Etiqueta: 9



Etiqueta: Perro

Malos ejemplos de Data Augmentation

Oscurecer mucho la imagen puede hacer que el objeto de interés sea imposible de detectar.



Algunos objetos están muy caracterizados por un color particular.

Otras precauciones a tener en cuenta...

- Si nuestro dataset original contiene biases, podemos llegar a **arrastrar dichos biases** con las modificaciones implementadas.
- Las mejores transformaciones serán aquellas que **modifiquen características que no queremos que nuestro modelo aprenda**, pero hay que tener cuidado de no romper las etiquetas.
- En general, utilizar Data Augmentation hace que el **entrenamiento sea más lento** dado que nuestro dataset es más grande. Por lo tanto, siempre se recomienda **aumentar las iteraciones de entrenamiento** cuando se utilizan este tipo de técnicas.
- Además, algunos métodos de Data Augmentation presentan un **tiempo de cómputo considerable**, enlenteciendo aún más dicho entrenamiento.
- Utilizar métodos de Data Augmentation basados en Deep Learning **requiere un trabajo extra de generación y evaluación de dichos modelos**.

Consideraciones de diseño

Online y Offline Data Augmentation

La implementación de Data Augmentation, sobre los datos de entrenamiento, suele hacerse de dos formas: **online** u **offline** augmentation.

En **offline** augmentation, el dataset de entrenamiento se incrementa generando nuevos archivos, ya sea transformados o sintéticos, que se almacenan junto con los originales. Aporta mayor velocidad durante el entrenamiento.

En **online** augmentation, las transformaciones se van aplicando de forma aleatoria sobre los datos de cada batch durante el proceso de entrenamiento. Este método resulta conveniente cuando la cantidad de datos es demasiado grande, o el espacio de almacenamiento de los mismos es limitado.

Estas implementaciones no son excluyentes!

Consideraciones de diseño

Test Time Augmentation (TTA)

Es una técnica utilizada sobre el **conjunto de testeo** para medir y mejorar la performance del modelo. Consiste en realizar transformaciones sobre dichas imágenes y promediar las predicciones realizadas sobre las diferentes versiones de cada una para formar la predicción final.

No se suelen utilizar transformaciones muy complejas: rotación, recorte, cambio de escala, cambio de colores.

Se puede interpretar como un ensemble de modelos pero del lado de los datos.

La principal desventaja es que agrega más tiempo de procesamiento a cada inferencia.

https://stepup.ai/test_time_data_augmentation/

Ejemplo práctico

Entrenamiento de red neuronal aplicando técnicas de Data Augmentation.

Link a Colab

Comparación de librerías para aplicar Data Augmentation.

Link a Colab

Tensorboard

¿Qué es?

Tensorboard es una herramienta de visualización de experimentos y modelos de deep learning.

¿Qué se puede hacer?

Permite hacer un seguimiento de métricas como loss y accuracy, ver un grafo de la red neuronal, registrar resultados de los modelos, ver desviaciones en la distribución de los pesos, entre otras.

Ventajas y desventajas

- Fácil de usar de forma personal o en equipos pequeños.
- No permite versionar los datos y modelos.
- No escala bien cuando hay millones de datos.

Ejercicio

Entrenamiento sobre dataset de manos - efectos del Data Augmentation.

Link a Colab

Encuesta de clase

link