11089 - Sistemas Inteligentes - UNLu

Examen Parcial #1- 29/05/2021

Deep Face Recognition

Cavasin Nicolás - Legajo #143501

Resumen:

Uno de los más importanes contribuyentes al surgimiento de las redes neuronales convolucionales -CNN de ahora en más- fué la disponibilidad de datasets con grandes cantidades de datos que permitan entrenar modelos exhaustivamente. Sin embargo, para el reconocimiento facial no existen datasets *públicos* de gran escala pues pertenecen a compañías como Facebook o Google cuyo último modelo (año 2015) fué entrenado con 200 millones de imágenes y 8 millones de identidades únicas, valores que son 3 órdenes de magnitud mayor que los abiertos al público. Por estos motivos es que los autores del documento se proponen dos objetivos:

1. Construir un dataset y abrirlo.
2. Utilizarlo para entrenar a diferentes arquitecturas de CNNs en el reconocimiento facial contrastando sus rendimientos.

Para la creación del dataset se aplicó una estrategia *multi-step* de 5 etapas con el fin de lograr la mejor consolidación posible. Estas son: selección de candidatos, recolección de imágenes adicionales, purificación, remoción de duplicados y filtrado manual.

1 - Selección de candidatos:

Extrajeron el ranking de popularidad de la Internet Movie Database (IMDb) y lo intersectaron con el Freebase Knowledge Graph obteniendo una lista de 5000 candidatos (2500 masculinos y 2500 femeninos). Descargaron 200 imágenes para cada uno de los 5000 nombres en la lista y las presentaron a *anotadores* -4 grupos de 50 personas- para que seleccionen las identidades cuyas imágenes asociadas eran lo suficientemente puras (90% o más). Resultado parcial: 3250 identidades. Por último eliminaron las que aparecen en los datasets *Labelled Faces in the Wild (LFW)*[1] y *YouTubeFaces (YTF)*[2] para evitar repetición. Resultado final: 2622 identidades.

2 - Recolección de imágenes adicionales:

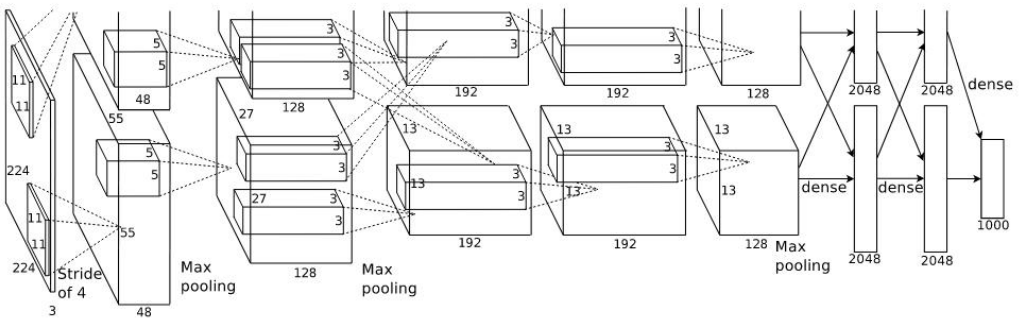
Se consultó a Bing y a Google por cada identidad de la lista en dos oportunidades, la primera con su nombre completo y la segunda con el sufijo *actor*. Resultado: 2000 imágenes por identidad.

3 - Purificación:

Para cada identidad se seleccionó un top 50 de sus mejores imágenes tomadas como samples positivos. Luego se ejecutó un clasificador SVM lineal del tipo uno-vs-resto por cada identidad, tomando como samples positivos al top 50 de dicha identidad y como samples negativos al top 50 de las restantes. Resultado: 1000 imágenes por identidad.

4 - Remoción de duplicados:

Eliminaron las imágenes duplicadas (encontradas por diferentes motores de búsqueda) y también los duplicados *cercanos*, que solo se diferenciaban por tener distinto balance de colores o alguna leyenda.



Arquitectura AlexNet

5 - Filtrado manual:

Teniendo 1000 imágenes por identidad se aplicó un método semi-supervisado entrenando a una CNN para ser capaz de diferenciar las 2622 identidades utilizando la arquitectura AlexNet, definida por Krizhevzky, Sutskever y Hinton[3]. Posteriormente se le entregó a los *anotadores* bloques de 200 imágenes que solo se aprobaban si su pureza era 95% o más.

Imágenes presentes en el dataset obtenido

El dataset quedó compuesto por 982.803 imágenes, de las cuales 95% son frontales y el 5% restante de perfil.

Respecto a la construcción y arquitectura de la CNN, se crearon dos clasificadores diferentes y utilizaron 3 variantes de las 5 arquitecturas propuestas por Simonyan & Zisserman[4]. El primer modelo de la CNN ajustó por cada imagen un vector de pesos y luego -al ser abordado como un problema de clasificación de N-posibilidades siendo N = 2662-, mediante una capa final completamente conexa (*FC* de ahora en más) utilizó un predictor lineal por cada identidad que asociaba cada peso a una única identidad. Finalmente, descartó el clasificador lineal y solo retuvo los vectores de pesos ya que permitían el reconocimiento de caras mediante comparaciones usando la distancia Euclideana.

El segundo modelo fué creado para optimizar la razón de ser de la CNN, que es la identificación de caras utilizando la distancia Euclideana, y por ende no fué abordado como un problema de clasificación con N-posibilidades sino buscando minimizar las diferencias entre cada peso. Este modelo utilizó los vectores de pesos obtenidos en el modelo anterior y los tuneó mediante un esquema de entrenamiento llamado *triplet-loss* que tiene dos diferencias fundamentales con el modelo inicial:

1. La cantidad de clases N =/= 2662 identidades, es 1024.
2. El ajuste del vector de pesos se realiza con un triplete *(a, p, n)* siendo:
   a) Una cara la *a* (anchor/anclaje).
   b) *p* los ejemplos positivos de la identidad asociada a la cara.
   c) *n* los ejemplos negativos.

Con respecto a las variantes de arquitectura, se basan en la versión A de Simonyan & Zisserman[4] que consta de 11 bloques donde c/u contiene un operador lineal seguido las funciones de activación ReLU o max-pooling. Los primeros 8 son convolucionales -el operador es un banco de filtros lineales- mientras que los últimos 3 son FC y su cantidad de filtros es igual a la cantidad de entradas que recibe la CNN. Los primeras dos capas FC poseen 4096 salidas mientras que las salidas de la última FC dependen del clasificador utilizado: si es el primero son 2662 y si es el segundo son 1024. Respecto a las variantes B y D, estas agregan 2 y 5 capas convolucionales respectivamente a la arquitectura de A. La entrada de todas las variantes de la CNN es la misma, reciben una imagen cuyo tamaño es una matriz cuadrada de 224x224 valores.

La arquitectura A fué entrenada desde cero y sus resultados tomados como puntos de partida para las variantes B y D gracias al agregado de 2 y 5 capas convolucionales respectivamente (como se dijo antes), inicializando los pesos de las mismas al azar, entrenándolas y luego re-entrenando la red por completo con una velocidad de aprendizaje menor a la de A.

El rendimiento de estas arquitecturas fué testeado sobre un SLI[5] de 4 GPUs Nvidia Titan Black de 6Gb de RAM y utilizando el dataset LFW. Por último, se comparó su rendimiento contra los de los métodos que se consideran *estado del arte* y que fueron testeados contra los datasets LFW e ITF.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Arquitecturas de Simonyan & Zisserman[4].

Se detectó que la arquitectura B provee una pequeña mejora respecto a los resultados de A mientras que D, a pesar de ser la más compleja, no presentó mejoras que justifiquen su utilización por sobre B. Respecto a los dos clasificadores, el segundo que utilizaba *triplet-loss* no solo introdujo una mejora de 1.8% respecto al primero sino que también redujo el ratio de error un 68%, convirtiéndolo en un factor clave.

La conclusión más importante es que se obtuvieron resultados muy similares -y en su mayoría superiores- a los de los métodos considerados *estado del arte* aplicando un entrenamiento apropiado, sobre un dataset nuevo con pocas imágenes (comparado a los de las grandes corporaciones) y con una arquitectura no muy compleja, lo que convierte a esta CNN en algo muy prometedor.

Crítica:

Decidí hacer énfasis en la explicación de la construcción del dataset ya que pocas veces se encuentra información disponible acerca de lo que piensan los autores durante el proceso de consolidación; me parece de vital importancia un aporte de tales características ya que brinda un verdadero *insight* sobre las decisiones tomadas y el porqué de las mismas. Por ejemplo, en el paso 1 la eliminación de las identidades presentes en los datasets *LFW* e *YTF* pudo ser obviada para evitar una reducción considerable de personas así como también la eliminación de duplicados idénticos y duplicados cercanos pudo ser realizada antes del paso 2 (recolección de imágenes adicionales con Google y Bing).

Respecto al rendimiento de las 3 arquitecturas contra el dataset LFW, que la arquitectura D no haya presentado mejoras por sobre B probablemente se deba al agregado de 3 capas convolucionales las cuáles son muy sensibles al setteo de parámetros -como la velocidad de aprendizaje- y su manipulación es crítica. Tal vez los autores no realizaron los suficientes intentos así como tampoco consideraron entrenarla de cero y es por eso que no presentó una mejora sustancial.

Esta nueva CNN forma parte ahora de lo que se considera *estado del arte* si utiliza el método de *triplet-loss* y ha logrado elevar la vara ya que con un dataset de 2.6M de imágenes han superado, en el rendimiento sobre YTF, a FaceNet + Alignment[6] que fué entrenada con 200M de imágenes. Lo mismo sucedió con la CNN Fusion[7] sobre el dataset LFW, que fué entrenada com 500M de imágenes y alcanzó un 98.37 de accuracy mientras que esta CNN llegó al 98.95.

Las CNNs han revolucionado el mundo de visión por computadora y es algo muy alentador que los autores hayan decidido construir y abrir un dataset que no posea nada en común con los ya existentes ya que permite que la comunidad interesada puede beneficiarse del mismo, algo que a fin de cuentas favorece la investigación y perfeccionamiento de las mismas.

Referencias:

[1] - Labeled Faces in The Wild: http://vis-www.cs.umass.edu/lfw/index.html

[2] - YouTubeFaces dataset: https://www.cs.tau.ac.il/~wolf/ytfaces/

[3] - Krizhevsky, Sutskever & Hinton: ImageNet Classification with Deep Convolutional Neural Networks: https://kr.nvidia.com/content/tesla/pdf/machine-learning/imagenet-classification-with-deep-convolutional-nn.pdf

[4] - Simonyan & Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition: https://arxiv.org/pdf/1409.1556.pdf.

[5] - Scalable Link Interface: https://en.wikipedia.org/wiki/Scalable_Link_Interface.

[6] - Schroff, Kalenichenko & Philbin: FaceNet: A Unified Embedding for Face Recognition and Clustering: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Schroff_FaceNet_A_Unified_2015_CVPR_paper.html

[7] - Taigman, Yang, Ranzato & Wolf: Web-Scale Training for Face Identification: https://openaccess.thecvf.com/content_cvpr_2015/papers/Taigman_Web-Scale_Training_for_2015_CVPR_paper.pdf