Computer Vision

GUILLERMO GARCIA OTIN

November 2020

1 Problem analysis

Esta tarea tiene la principal dificultad de no tener mucho *labeled data*. Por otro lado, las imágenes son de diferentes dimensiones, algunas muy grandes de hasta 2000 píxeles. Dadas las características de la tarea se proponen dos soluciones:

- Traditional Computer Vision: En esta solución, se implementa un detector de color mediante la librería *OpenCV*. Primero, las imágenes se transforman al espacio HSV donde se puede llevar a cabo el *thresholding* de color de una manera más eficaz. Una vez se ha ajustado el *thresholding* de azules con las 10 *labeled images*, se procede a separar las camisetas azules de las que no lo son. Para esto, se calcula el el tamaño del área azul detectada y se compara con el área de la imagen. Se necesita hacer de forma relativa a la dimensión de la imagen ya que, como se ha explicado anteriormente, las imágenes tienen tamaños diferentes. Por último, se clasifican las imágenes como camiseta azul o no comparando este parámetro con un *threshold* calculado de manera empírica en las *labeled images*.

- Deep Learning: En esta solución se aplican redes neuronales, las cuales son el *state-of-the-art* en tareas de *Computer Vision* como reconocimiento ó detección de objetos. Aunque las características de este problema (poco *labelled data*, mucha variación en los tamaños de imagen y el patrón de diferenciación esta en los píxeles propiamente dichos) no lo hacen adecuado para la utilización de redes neuronales, se explora la posibilidad de su utilización. Modelos para la diferenciación de objetos basados en color mediante redes neuronales han sido también llevados a cabo ppor [1, 4]. Se explora la clasificación con redes neuronales de manera que el problema sea más generalizable y sin tanto trabajo de ajuste de parámetros manual, sin embargo, la detección con el primer método arroja relativamente buenos resultados por las características enunciadas anteriormente. En este método se propone lo siguiente para solventar los problemas:

  - La falta de *labeled data* se solventa usando el clasificador del primer método con un margen pequeño, de manera que las imágenes clasificadas como azul serán con mucha probabilidad azules. Las

imágenes de camisetas no azules se consiguen del subset restante aleatoriamente. Además, las imágenes labeled se añaden a las azules seleccionadas por el primer modelo. Es decir, usamos el primer método a modo de *labeller* con la esperanza de que la red neuronal pueda generalizar a camisetas azules que el primer método fallaba. Adicionalmente, se utiliza *data augmentation* con transformaciones que no afectan al color de la imagen. Las imágenes son giradas aleatoriamente.

- El problema de diferentes sizes de las imágenes se solventa reescalando a (500 x 500) todas las imágenes. Se ha estudiado mantener la forma original de las imágenes y extraer mediante *Regions of Interest pooling* representaciones constantes a todas las imágenes para después clasificarlas con *fully conected layers*, sin embargo se ha considerado que reescalar las imágenes es una opción igualemente válida porque el patrón de clasificación es el color.

- Después del proceso de entrenamiento, los embeddings de las imágenes serán clusterizados en dos clusters, por lo tanto necesitamos embeddings con información suficiente como para que puedan diferenciarse camisetas azules de no azules. Por eso, y tal y como demostraron [2, 3], arquitecturas neuronales triplets y siamesas son capaces de aprender embeddings mucho más discriminadores, que habituales redes neuronales entrenadas para la tarea de clasificación. Además, estas redes tienen la ventaja de poder formar muchos pares/trios de entrenamiento ya que estos son formados con todas las distintas posibilidades de combinar las imágenes entre sí (crecimiento cúbico para redes triplets y cuadrado para siamesas). Se elige por tanto redes triplets para este problema.

- El siguiente problema es la elección de la arquitectura idónea. [1] demostraron que las capas más cercanas al comienzo de la red neuronal son más sensibles al color. Esto tiene sentido ya que a medida que la red es más profunda se aprenden patrones más complicados. Por tanto, se utiliza la propia red que se analiza en [1], Alexenet, preentrenada en Imagenet para hacer de *feature extration*. Seguidamente se hace un max pooling a modo de reducción, a continuación se reorganiza en una dimensión (*flatten*)

y se aplica *dropout* de 0.5 para evitar *overfitting*, finalmente se aplica una *fully conected layer* con 16 dimensiones de salida, la cual será la dimensión de nuestros embeddings. Durante el entrenamiento la distancia entre los embeddings de las imágenes de camiseta azul se hará más próximo en términos de distancia y más grande a los del otro tipo.

2 Models validation

La evaluación de este problema también presenta un desafío, ya que, no se cuenta con etiquetas para su comprobación directa. Por un lado, comprobar una a una las imágenes visualmente es un coste humano súper elevado y, por otro lado comprobar muy pocas imágenes nos dará un estimación errónea. Para solucionar esto, se propone acompañar los resultados del intervalo de confianza, el cual nos dirá la precisión del método y la seguridad de la estimación.

A continuación se presentan los resultados del primer método con intervalo de confianza 0.9: Precisión: 0.9137. Recall: 0.9431. F1-Score: 0.92816

El segundo método arroja malos resultados incluso con los datos seleccionados (0.56), por lo tanto solo se presentan los datos con intervalos de confianza del primer método.

3 Final summary

Con más tiempo y data los métodos se pueden mejorar de la siguiente manera: Método 1: Al tener más data y más tiempo se puede ajustar perfectamente el intervalo de azules que se quiere detectar. Además, si tenemos más labeled data se puede hacer un programa que automáticamente compruebe la precisión, recall y f1-score y varié los parámetros de color de manera que los maximice. De esta manera se conseguiría un programa totalmente autónomo sin inspección manual.

Método 2: Con más tiempo y data se puede entrenar una red neuronal completa y que los parámetros aprendan a discernir el color apropiadamente. Además con más tiempo para ajustar los hiperparámetros la red aprenderá de manera asegurada

Otra solución consistiría en primero localizar la camiseta y después extraer su color. Sin embargo, como la mayoría de imágenes contienen la camiseta en el centro, en primer plano y con un background uniforme, se ha decidido no resolver el problema con este método.

References

[1] M. Engilberge, E. Collins, and S. Süsstrunk. "Color representation in deep neural networks". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 2786–2790. DOI: 10.1109/ICIP.2017.8296790.

[2] R. Hadsell, S. Chopra, and Y. LeCun. "Dimensionality Reduction by Learning an Invariant Mapping". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. 2006, pp. 1735–1742. DOI: 10.1109/CVPR.2006.100.

[3] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[4] Mingyang Zhang, Pengli Wang, and Xiaoman Zhang. "Vehicle Color Recognition Using Deep Convolutional Neural Networks". In: July 2019, pp. 236–238. ISBN: 978-1-4503-7150-6. DOI: 10.1145/3349341.3349408.