

Informe Tarea 1 - ¿Qué hora es?

Kaori Kanno

Pontificia Universidad Católica de Chile

La capacidad de modificar imágenes ha ganado gran importancia en las últimas décadas debido a que cada vez más personas tienen la capacidad y las herramientas para hacer esto. Entender cómo modificar imágenes es tanto relevante para sacar conocimiento de estas como para reconocer cuándo estas presentan alteraciones y cuáles son estas.

En esta tarea haremos un pequeño pre-procesamiento de 6 imágenes dadas de un reloj, y mediante la generación de patrones de dígitos seremos capaces de definir un algoritmo que reconozca la hora que aparece en este.

Solución propuesta

Primero se umbralizó la imagen para quitar ruido usando la función `umbralizacion`, de modo que quedó en valores de negro (0) o blanco (255).

Ya que algunas imágenes estaban mirando al reloj en ángulo, se usó el código del colab de la clase 05, la interpolación lineal `geo_transformation`, además de parte del código que permite hacer la deformación de la imagen, cambiando la figura geométrica a la que se debe transformar de un círculo a un cuadrado, esto ya que utilizamos las esquinas del reloj como referencia para modificar la imagen. Para esto último se utilizaron algunas funciones de numpy que permiten hacer lo visto en la clase 06, es decir, calcular el gradiente en cada línea de la imagen, de modo de detectar bordes. Usando finalmente `argmin` y `argmax` se pudieron encontrar los puntos máximos y mínimos de estos bordes.

Con el reloj ya enderezado pasamos a la fase de segmentación, esto trazando una recta vertical en la imagen y guardando los valores en que se detecta el reloj (tono de 255) y donde se deja de detectar, tanto en dirección horizontal como vertical. Luego solo se usó la mitad superior de esta imagen ya que el resto no poseía información de interés.

Ya con esto, se segmentó cada dígito y el periodo horario en imágenes distintas que se analizaron por separado para extraer la información de cada uno. Para este último valor se definió una función que buscaba dónde de todo el segmento de imagen (pues este texto es pequeño), se encontraba la información de la franja horaria. Luego al detectarlo, se recreaba la imagen como un array de 0s y 1s, donde 0 es fondo y 1 es letra. Se reajustaron los tamaños de arrays, y con ello se hicieron comparaciones con arrays predefinidos mediante la función `detect_am_pm`. Estos arrays am o pm se hicieron basados en algunos de los arrays producidos por las mismas imágenes utilizadas aquí, de tal manera que probablemente solo funcionen bien con estas.

Para el primer dígito, ya que solo estaba la opción de 0 o 1

para estas imágenes, se usó la función `contar_ceros`, la cual definía si el valor era un 0 o un 1 en base a la porcentaje de píxeles de color. Para los otros valores se usó una idea similar a la de la función `detect_am_pm`, generando una matriz plantilla para cada dígito.

Experimentos realizados

Para este trabajo se usó el dataset entregado, el cual posee 6 imágenes distintas del mismo reloj en distintos ángulos y luminocidades a distintas horas.

Para umbralizar, se realizó un histograma de las 6 imágenes, donde un umbral de 0.1 surgía como valor aceptable para todos los casos. Los resultados se ven tras usar la función `umbralizacion`. Para segmentar la imagen se tomó la línea `y=500`, esto pues en las 6 imágenes resultaba contener al reloj. Se tomaron así los bordes superior e inferior de la imagen y con esta información se pudieron detectar los bordes horizontales. Para rotar la imagen se tomaron las esquinas, resultando esta una buena decisión para todas las imágenes salvo la sexta, pues para esta una de las esquinas no es bien tomada por el código. Para segmentar los números se llegó a que dividir la imagen según proporciones permitía que en cada imagen resultante existiera un dígito distinto de la hora del reloj. Las proporciones mostradas en la tarea funcionaban de forma correcta para este trabajo, como se muestra en las imágenes n1 a n4 e "imagen recortada" en la tarea. Para detectar los números en cada imagen se decidió volver cada una de las imágenes recién mencionadas en arrays de 0s y 1s y compararlos con plantillas hechas a partir de estos mismos arrays, de forma que se ajustan bien a estos valores y solo estos.

Finalmente con todo esto, se logró correctamente leer el periodo horario y cada dígito del reloj, por lo que la última línea del código entrega la hora de la imagen que se le inserte.

Conclusiones

La simetría es una buena característica para definir la segmentación en imágenes como las presentadas en este trabajo. El lograr algoritmos automatizados que logren hacer predicciones o análisis requiere siempre de un pre-procesamiento de las imágenes que permita que el algoritmo trabaje siempre bajo la misma premisa. En este trabajo se logró un algoritmo que automatice la lectura de la hora para estas 6 imágenes. A futuro, sería ideal generar un algoritmo capaz de hacer esto en cualquier imagen que se tome de este reloj.