UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Imagen Digital

Looking for Something

Curso 2019/2020

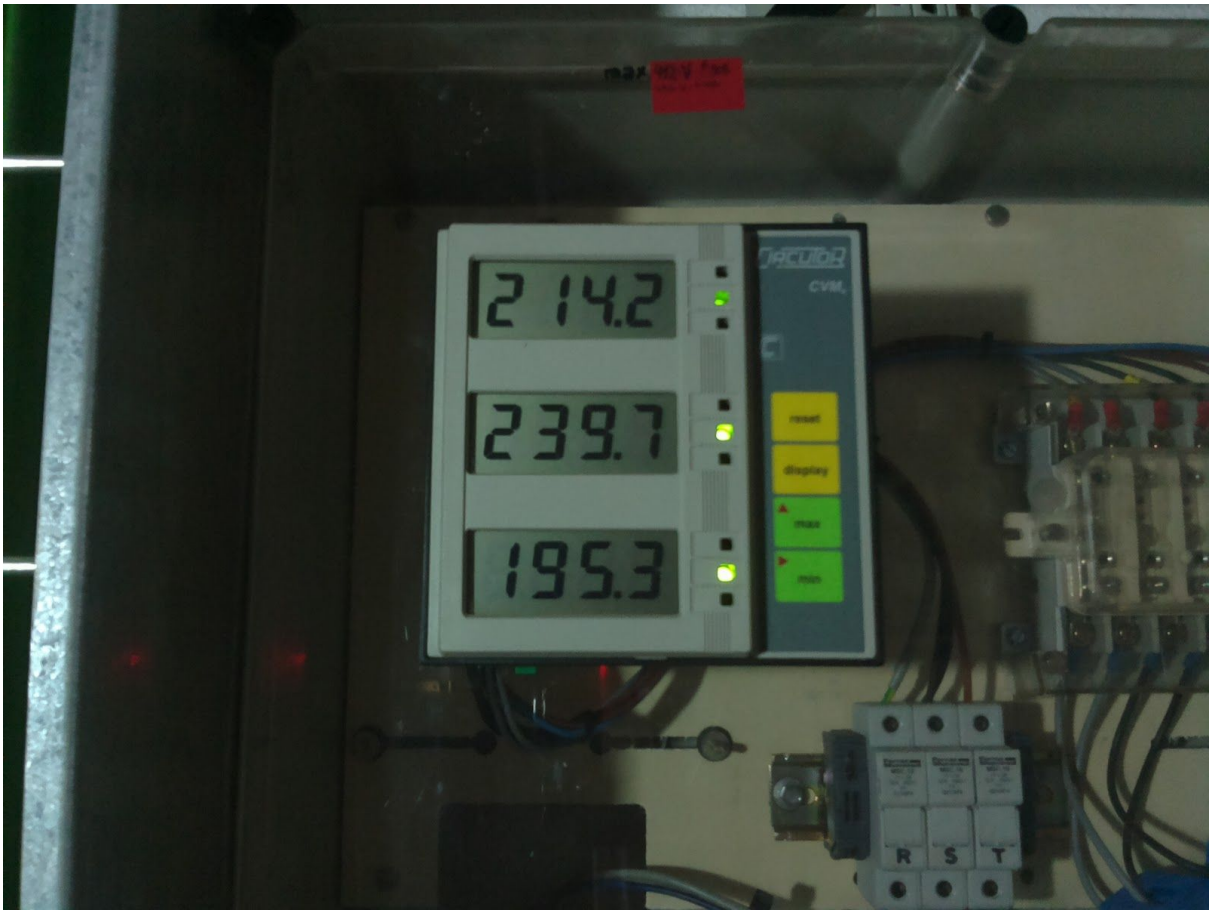
José María Crespo Sánchez

Gonzalo Jiménez Hinchado

Contenido

Introducción

El objetivo de esta práctica es reconocer un patrón determinado en una imagen dada. Para iniciarnos en los problemas de *match template*, se ha prácticado con el clásico ejemplo de Lena que el profesor ha proporcionado. Tras ello, se ha ampliado dicho problema para ajustarse a los requerimientos de la práctica en cuestión. En este trabajo, se parte de 12 imágenes contadores, en los cuales aparecen tres series de números en cada uno. Con una serie de imágenes de los 10 distintos números que pueden aparecer en cada contador, se ha reconocido cada uno de dichos números con el fin de digitalizar esos valores e imprimirlos en un fichero. A continuación, se muestra un ejemplo de una imagen de un contador y otra de un número con el cual se debía hacer el match template:



Definición del problema

En esta entrega el problema que se nos plantea es encontrar los rectángulos donde se encuentran los contadores con sus cifras dentro. Una vez encontrado el rectángulo donde se encuentran los contadores, se deben identificar los números que aparecen en ellos, analizarlos y mostrar la solución por pantalla. Además, la solución será extraida en un archivo .txt dentro de la raíz del directorio, con la siguiente estructura para cada línea:

Ruta_de_la_captura_analizada contador_1 contador_2 contador_3

De esta forma, el principal problema es la búsqueda de los números en la imagen, localizando su posición para ordenarlos de forma que la composición de números encontrados concuerde con el orden del número de la imagen.

Solución planteada

La solución que planteamos para este problema es la utilización de un paquete de multibúsqueda de templates en una imágen, llamado MTM (Multi-Template Matching), al cual le pasamos como parámetro una lista de todos los templates proporcionados, y una imagen en la que buscar, además de otros parámetros que precisan la búsqueda. Esto combinado con el recorte de los contadores en la imagen original facilita la solución del problema.

Lo primero, es cargar las imágenes template en una lista de pares de datos, con el nombre del template, en este caso el valor de la cifra que representa ese template (0, 1, 2, 3, 4, etc) y el archivo de imagen que utilizaremos para buscar en la imagen con el contador, en escala de grises en ambos casos.

Una vez cargados los patrones a buscar y la imagen del contador, el algoritmo utiliza la función de MTM *matchingTemplates* a la cual se le proporcionan:

- Lista de patrones: lista con todos los templates a buscar en la imagen.
- La imagen sobre la cual se van a buscar.
- Método de búsqueda: en nuestro caso usamos TM_CCOEFF_NORMED.
- Número de objetos a buscar: que aplicado a cada uno de los tres recortes de la imagen original da lugar a 4 cifras por búsqueda.
- Threshold: índice de acierto en la búsqueda.
- MaxOverlap: dado que el algoritmo puede encontrar un patrón indefinidas veces sobre la misma cifra, debemos indicarle la distancia mínima entre coincidencias, de forma que sólo guarde una ocurrencia del patrón sobre cada cifra.

Una vez ejecutado el método, la solución se almacena en un variable. Con esta variable representaremos en la imagen los patrones encontrados, las cuales se mostrarán en las imágenes de los recortes de los contadores.

Desafortunadamente, esta solución contiene datos ordenados por porcentaje de acierto, no por orden de búsqueda, por lo que hay que ordenar los datos de la solución. En nuestro caso, localizamos la coordenada horizontal (x) y lo ordenamos de menor a mayor, de forma que el primer número tendrá una coordenada horizontal menor que los siguientes. En el vector ordenado se guardan pares con la coordenada horizontal y el valor de la cifra encontrada.

Este método devuelve la imagen con los patrones encontrados y un vector que contiene la solución ya ordenada.

Tras aplicar el método, se representa la imagen ya procesada en la interfaz de usuario, y se guarda la salida en el archivo “out.txt”.

Este proceso puede ser hecho siguiendo una serie de pasos:

1. Seleccionar la imagen a analizar.
2. Cortar la imagen y extraer los 3 contadores.
3. Analizar las imágenes de los contadores en busca de los patrones, y extraer los números de cada uno.

O bien, se puede hacer todo el proceso a todas las imágenes almacenadas en un solo paso, mediante el botón “Global Process”. Tras su ejecución se habrán analizado todas las capturas de contadores y contrastado con todos los patrones de cifras, dando lugar a un archivo llamado “out.txt” que almacena la solución tal y como se explica previamente en esta documentación.

Salida

El resultado de la ejecución es guardado en un fichero *out.txt*. En dicho fichero figuran todos los números digitalizados de la captura analizada en ese instante junto a su nombre en una sóla línea. El formato de impresión es el siguiente:

Ruta_de_la_captura_analizada contador_1 contador_2 contador_3

Este fichero recoge todos los resultados de una ejecución, desde la primera extracción de números hasta que el programa es cerrado. En caso de ejecutarse el global process, el resultado de las 12 imágenes sera escrito en el fichero, siendo su ruta más breve que la ejecución de capturas individual.

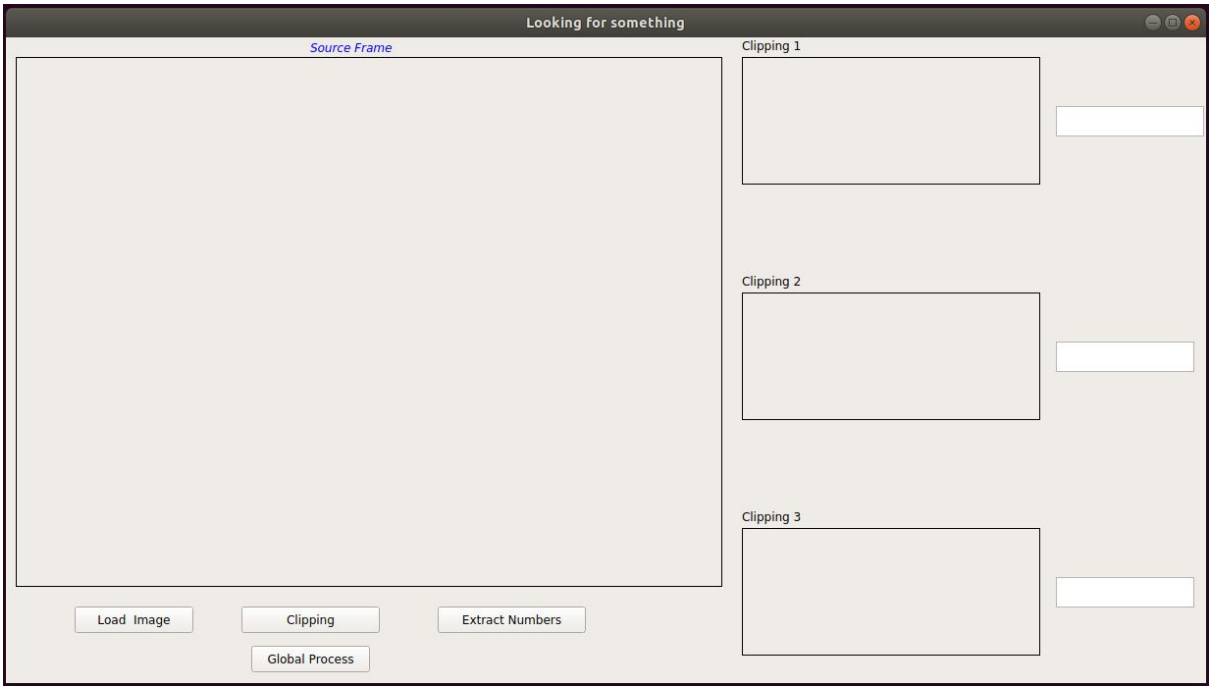
El archivo *out.txt* es sobreescrito con cada ejecución del programa, de tal manera que siempre contenga únicamente la información de la última vez que la aplicación fue lanzada.

Instrucciones de funcionamiento

Para ejecutar el programa es necesario escribir la siguiente instrucción en un terminal de comandos:

$ python main.py

A continuación, se abre una ventana creada con la herramienta QtCreator.



En la captura anterior, se puede observar que hay cuatro botones en la parte inferior. Cada uno de ellos es explicado a continuación:

- **Load Image:** Este botón despliegua una ventana con la cual seleccionar la captura que deseamos analizar y la muestra en el marco principal denominado *Source Frame*.
- **Clipping:** Este botón recorta los tres contadores de la captura y los muestra en los tres marcos situados en la mitad derecha de la ventana, denominado *clipping 1*, *clipping 2* y *clipping 3*. Además, reconoce los números que aparecen y dibuja un recuadro azul por cada uno de ellos reconocido.
- **Extract Numbers:** Este botón extrae los números de las imágenes recortadas y los muestra en los tres espacios a la derecha de las imágenes a las que corresponden. Además, tras extraer los números, imprime la salida en el fichero *out.txt*.
- **Global Process:** Este botón ejecuta las tres acciones anteriores con la única excepción de que no da la opción de elegir que imagen analizar, se realiza el proceso con las 12 imágenes de contadores de que disponemos.

Para cerrar la aplicación únicamente hay que cerrar la ventana pulsando la ‘x’ situada en la parte superior derecha de la misma.

Conclusiones

Por medio de esta práctica, hemos profundizado más en el uso de OpenCv con Python y en QtCreator. Ha resultado una práctica muy interesante, aunque cabe destacar que nos ha resultado complicado en un principio. Tras una primera introducción, el trabajo ha sido más sencillo de lo esperado. Aunque hemos tenido graves problemas en el reconocimiento de cuadrados, para recortar los contadores de las imágenes, y hemos decidido omitir dicha parte del código.