

Proyecto visión: Control de calidad y posicionamiento de la línea de producción de tortas

Pablo Adrián Moreno Sierra

pabmosie@upv.es

Automatització i robòtica (33424), 2016, ETSINF, Universitat Politècnica de València

Resumen

Una línea de producción industrial ha desarrollado una línea de producción de tortas con el fin de aumentar su productividad. Actualmente la elaboración de las tortas se hace automáticamente. Ahora, se desea desarrollar un sistema de detección de tortas a través de una cámara, que nos permita averiguar la posición de las tortas buenas para que el robot pueda cogerlas. Para ello, primero debemos analizar todas las tortas para comprobar si están en buenas condiciones, esto es, que no están quemadas, crudas, rotas, agujereadas o deformadas. Y además, en un apartado de ampliación, se ha tenido que identificar el centro de diversas tortas completas en la misma imagen.

Para este proyecto se utilizará el software OpenCV de reconocimiento de imágenes en Microsoft Visual Studio utilizando C++.

Palabras Clave:

OpenCV, reconocimiento de imágenes, control de calidad, control de posicionamiento.

0. Preparación de las imágenes

En todos los casos de tortas unitarias, lo primero que se hace con las imágenes es limpiarlas, es decir, dejar únicamente la torta con el fondo negro y otra imagen binaria de solo blancos o negros.

Primero se calcula la imagen binaria, para ello se aprovecha que el fondo es azul, por lo que se hace un *threshold* con el canal azul de la imagen, y esto nos da la imagen binaria que se ve a continuación:

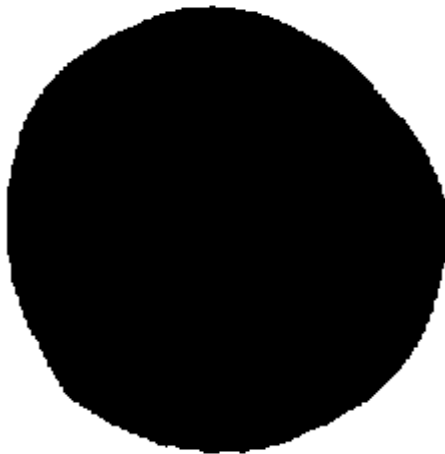


Figura 1: Imagen binaria

Esta imagen binaria se resta al vector de la imagen original, lo que da la imagen limpia (sin el fondo azul) como se puede ver a continuación:

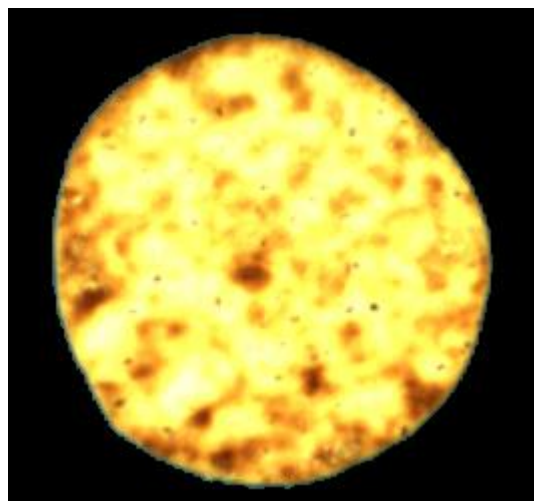


Figura 2: Imagen limpiada

Estas dos imágenes se guardan en memoria para poder utilizarlas para los siguientes análisis de calidad y posicionamiento.

1. Detección de tortas deformadas

Las tortas deformadas, son aquellas que les falta gran parte de la torta:

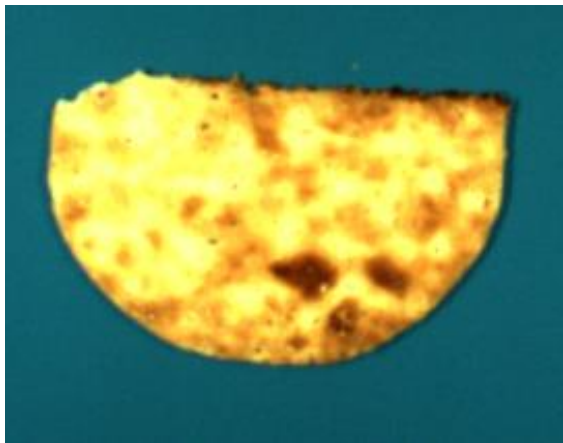


Figura 3: Torta deformada

Para identificar este defecto, se utiliza la imagen binaria para identificar las formas más fácilmente, y se utilizan las funciones *Canny* y *findContours* para identificar el contorno de la torta y calculamos el área de esta. Con el área vemos si esta es menor a un cierto umbral, en tal caso, estará deformada.

Esta torta es posiblemente la más sencilla de distinguir, además se podría comprobar de más maneras como viendo la circularidad de la forma, o incluso comparando su *minEnclosingCircle* con el *minAreaRect*.

2. Detección de tortas agujereadas

Las tortas agujereadas, son aquellas que tienen algún agujero en el interior de la torta:

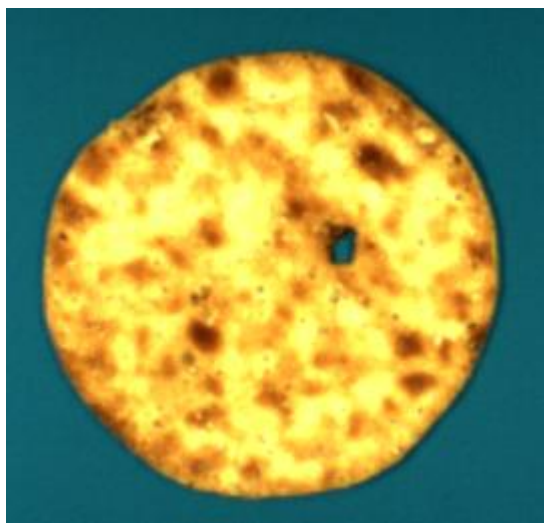


Figura 4: Torta agujereada

Para identificar estas tortas, se utiliza la imagen binaria para identificar las formas más fácilmente, y se utilizan las funciones *Canny* y *findContours* para identificar el contorno de la torta y ver los diferentes contornos que podamos encontrar en la imagen.

Con el vector *hierarchy* que proporciona la función *findContours* se puede ver la jerarquía de los contornos encontrados en la imagen, de manera que, se puede ver si un contorno está dentro de otro. Por lo que utilizando esto se pueden encontrar agujeros dentro de la torta.

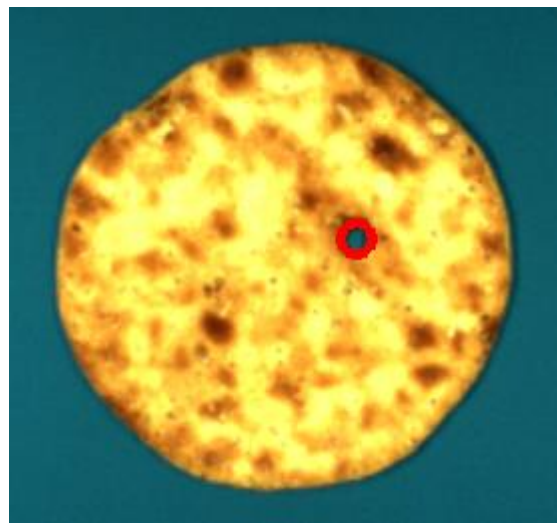


Figura 5: Detección de torta agujereada

3. Detección de tortas rotas

Las tortas rotas se caracterizan por no ser circular debido a un “mordisco” de la torta que puede ser más o menos grande:



Figura 6: Torta rota

Este análisis es bastante complejo debido a que algunas tortas tienen un “mordisco” muy pequeño, lo que lo hace difícil de encontrar.

Para identificar este tipo de tortas, en primer lugar, se utiliza la imagen binaria para identificar las formas más fácilmente, y se utilizan las funciones *Canny* y *findContours* para identificar el contorno de la torta y además se utiliza la función *minEnclosingCircle* para formar un círculo con su contorno y así poder obtener su punto central y trabajar con él.

Se analiza todo el contorno, calculando la distancia euclídea entre dos puntos, del teorema de Pitágoras de forma manual. Midiendo así la distancia entre cada punto del contorno y el punto central. Se analizan las distancias de los puntos con una separación del 1% de los puntos de contorno, es decir, si tenemos 600 puntos en el contorno, el 1% es 6, por lo que se compara el punto 1 con el punto 7, el 2 con el 8, y así sucesivamente. Se comparan las distancias de estos dos puntos al centro y si la diferencia de distancia es mayor al 5%, se considerará que esa torta está rota debido a que hace un “escalón” ligeramente pronunciado y si este defecto ocurre al menos una segunda vez, indica que hay un “escalón” de subida, lo que sería un “mordisco”.

4. Detección de tortas crudas

Las tortas crudas, son aquellas que son bastante claritas, por lo que están poco cocinadas:



Figura 7: Torta cruda

Para identificar esta torta, se utiliza la imagen limpia (la imagen sólo de la torta con el fondo negro) y se realiza un histograma de los colores.

En dicho histograma, se analiza la media del color verde, si esta media es menor a un cierto umbral, se considerará que esta torta está cruda.

Se han encontrado más formas de ver si una torta está cruda mediante el histograma, como por ejemplo, comprobando que la media de los rojos sea menor a un cierto umbral, como en el caso de los verdes, o incluso comprobando si los máximos niveles de verdes o rojos están también en unos determinados umbrales.

5. Detección de tortas quemadas

Las tortas quemadas, son aquellas que tienen partes bastante oscuras, por lo que están demasiado cocinadas:

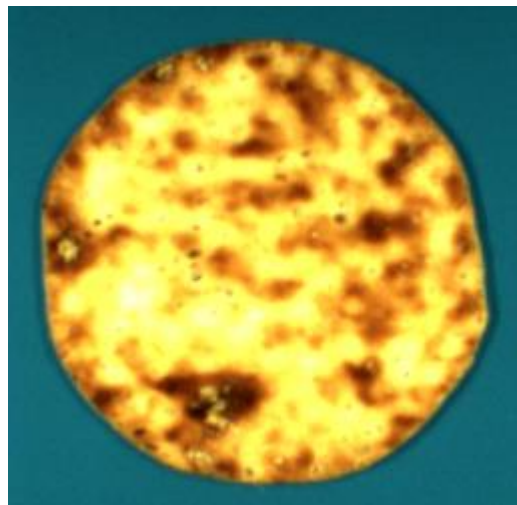


Figura 8: Torta quemada

Para identificar este defecto, se utiliza la imagen limpia (la imagen sólo de la torta con el fondo negro) y se realiza un histograma de los colores.

En dicho histograma, se analizan varios umbrales como son, el mínimo rojo ha de ser mayor que en tortas bien cocinadas, y las medias de verdes y rojos han de estar entre unos intervalos.

6. Resultados de tortas indeterminadas

En este apartado podemos ver los resultados del conjunto de 12 tortas indeterminadas y como estas son clasificadas junto con su tiempo de procesamiento en segundos.

Nombre de imagen	Clasificación	Tiempo procesamiento
I1	Rota	0.016
I2	Deformada	0.016
I3	Deformada	0.016
I4	Rota	0.016
I5	Agujereada	0.020
I6	Buena	0.021
I7	Rota	0.017
I8	Quemada	0.021
I9	Quemada	0.021

I10	Rota	0.017
I11	Rota	0.017
I12	Buena	0.021

- **Indeterminada 1:**

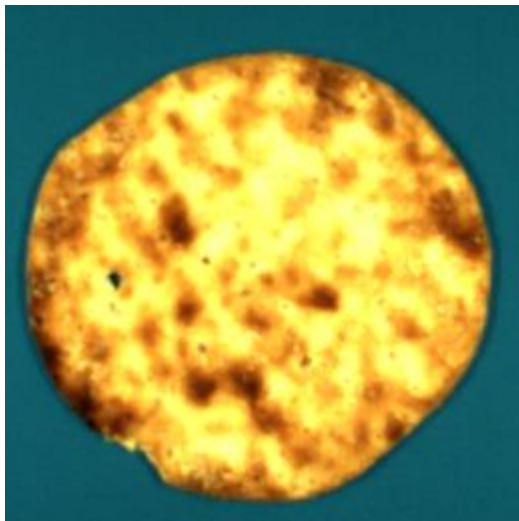


Figura 9: Torta indeterminada 1

Esta torta se determina como rota dado que abajo izquierda tiene un leve “mordisco”, y se realiza antes el análisis de roturas que de agujeros, de lo contrario se detectaría antes como agujereada como se puede ver en la imagen siguiente:

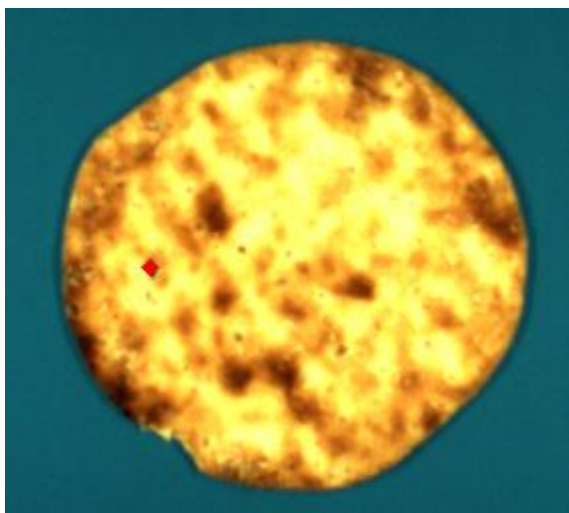


Figura 10: Torta indeterminada 1 se detecta agujereada

No se ha considerado por tanto que eso sea un fallo y no se ha cambiado.

- **Indeterminada 2:**

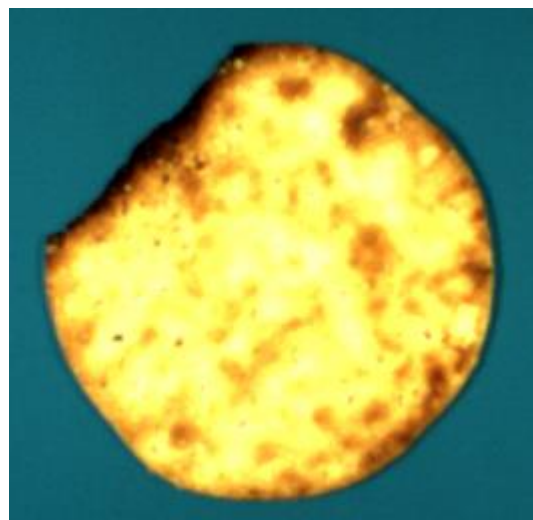


Figura 11: Torta indeterminada 2

Esta torta está claramente deformada.

- **Indeterminada 3:**

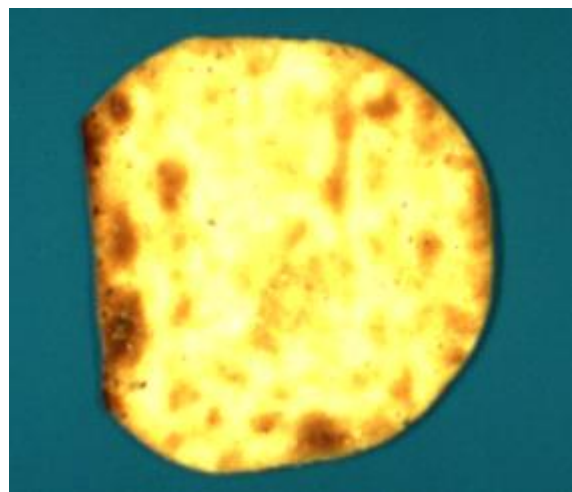


Figura 12: Torta indeterminada 3

Esta torta también está claramente deformada.

- **Indeterminada 4:**

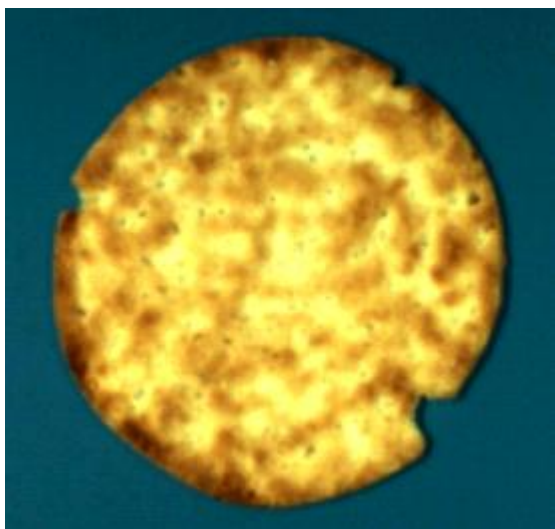


Figura 13: Torta indeterminada 4

En esta torta podemos identificar dos “mordiscos”, uno a la izquierda y otro a la derecha, por lo que ha sido correctamente identificada como rota.

- **Indeterminada 5:**

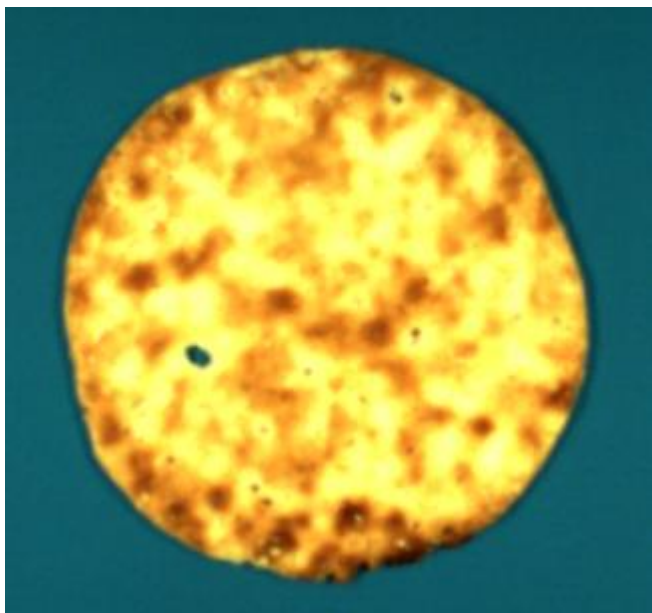


Figura 14: Torta indeterminada 5

En esta torta podemos apreciar un agujerito en la parte izquierda, y como podemos ver ha sido detectado como agujero:

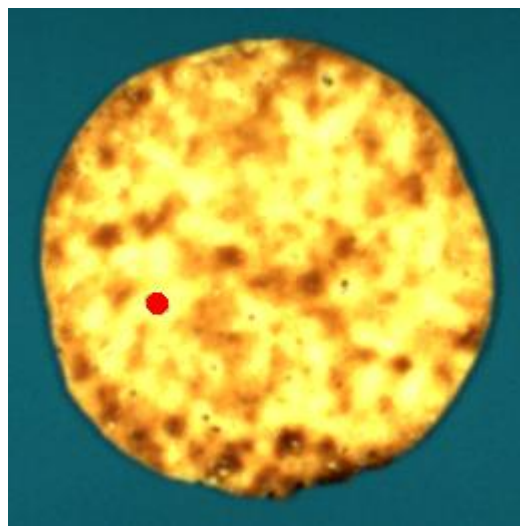


Figura 15: Torta indeterminada 5 se detecta agujereada

- **Indeterminada 6:**

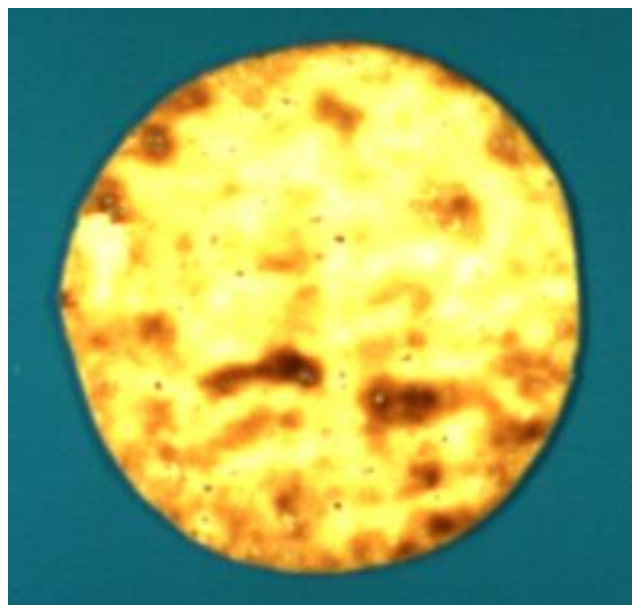


Figura 16: Torta indeterminada 6

Esta torta se ha considerado como buena ya que aparentemente no presenta ningún defecto de forma y no parece estar ni cruda ni quemada, ya que las crudas son muy blancas y las quemadas son bastante más oscuras.

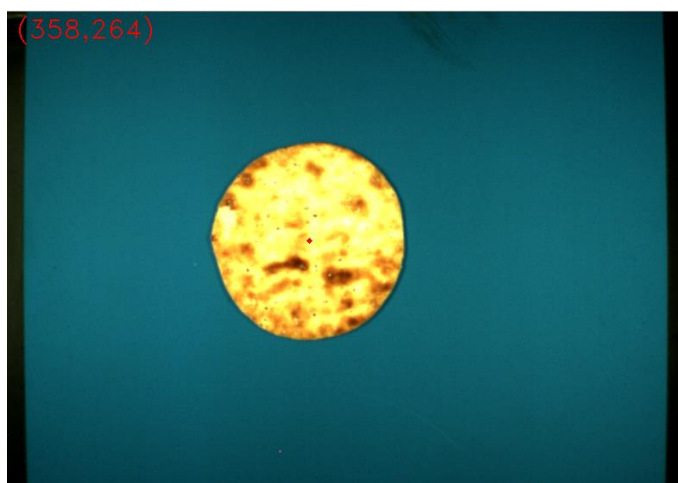


Figura 17: Torta indeterminada 6, resultado

- **Indeterminada 7:**

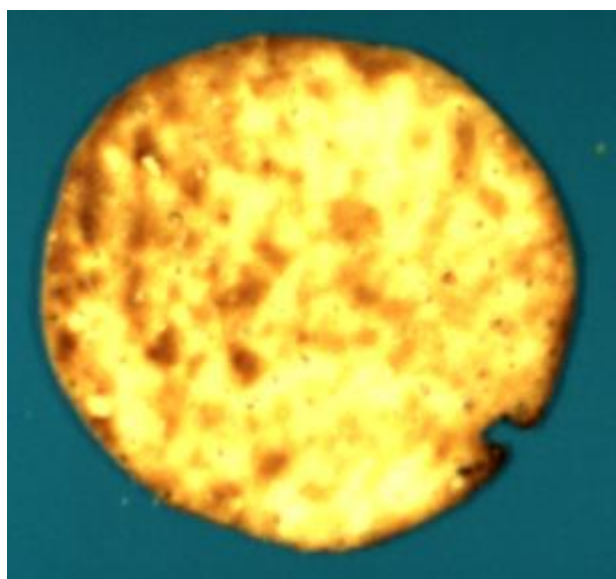


Figura 18: Torta indeterminada 7

Esta torta tiene una rotura a la derecha, por lo que ha sido correctamente identificada como rota.

- **Indeterminada 8:**

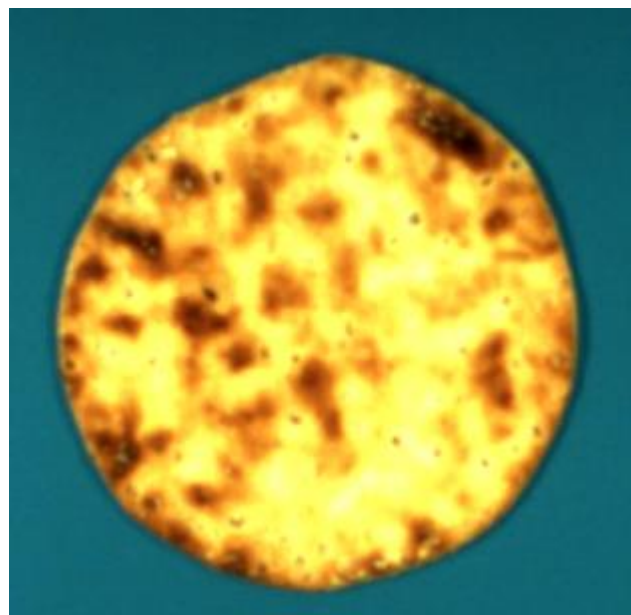


Figura 19: Torta indeterminada 8

Esta torta parece un poco quemada, especialmente en la parte superior derecha, por lo que se cree que ha sido correctamente identificada como quemada.

- **Indeterminada 9:**

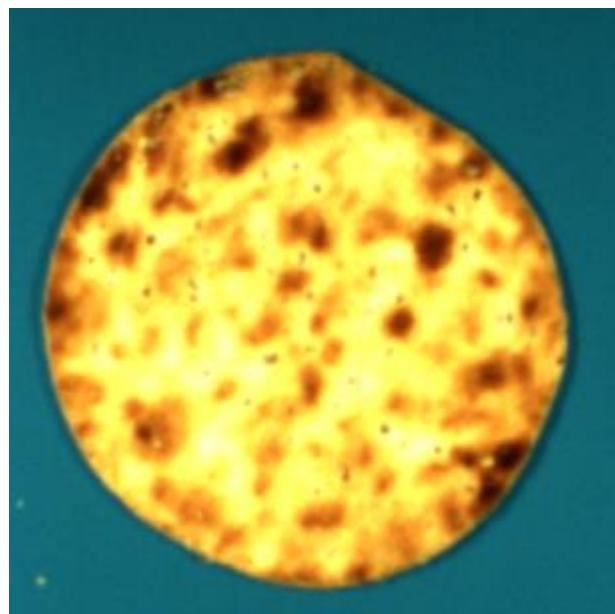


Figura 20: Torta indeterminada 9

En esta torta podemos ver diferentes trozos quemados, por lo que se ha considerado como quemada correctamente.

- **Indeterminada 10:**

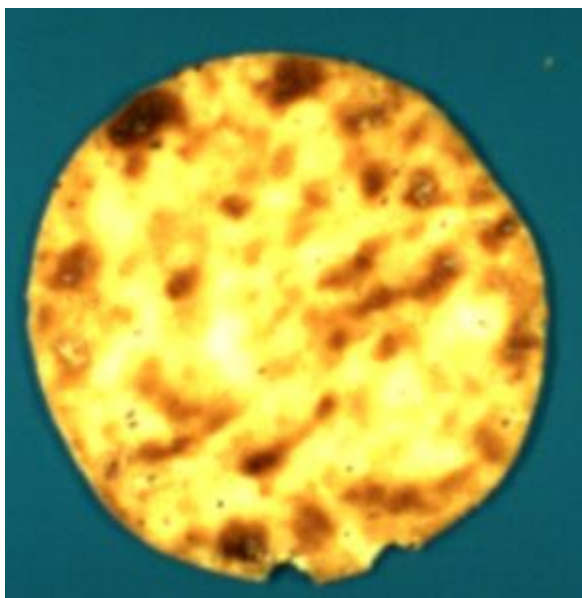


Figura 21: Torta indeterminada 10

Esta torta tiene en la parte de abajo dos muescas, por lo que ha sido procesada como rota.

- **Indeterminada 11:**

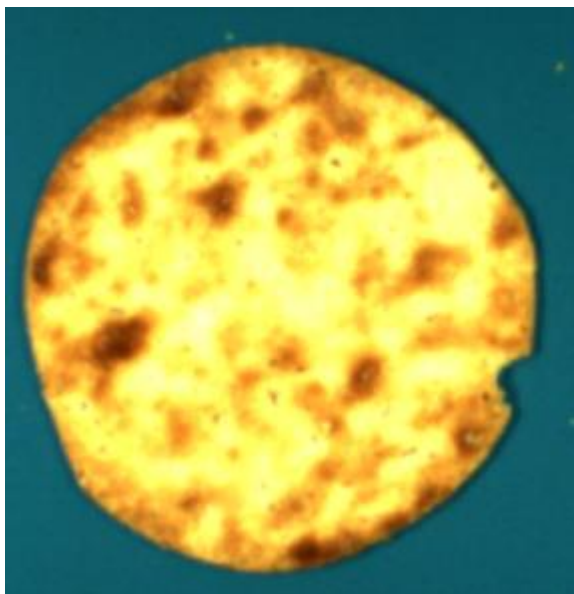


Figura 22: Torta indeterminada 11

Esta torta tiene un “mordisco” a la derecha por lo que ha sido detectada como rota correctamente.

- **Indeterminada 12:**

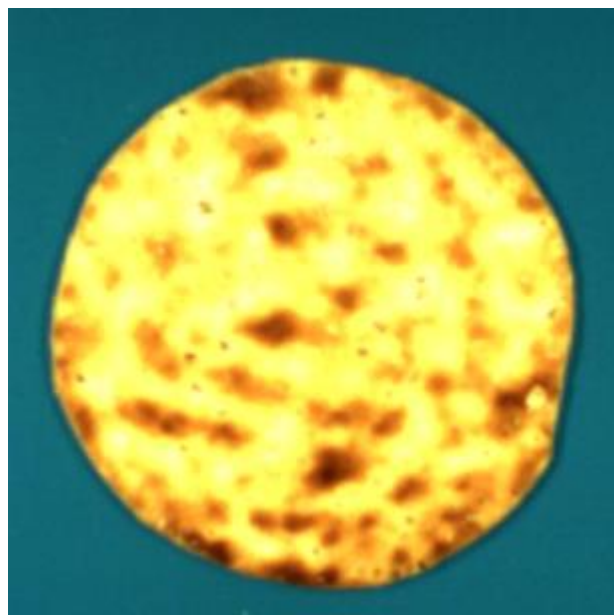


Figura 23 Torta indeterminada 12

Esta torta se ha considerado como buena. Se ha llegado a pensar que podía ser cruda por un poquito blanquecino que tenía, pero comparando con las imágenes crudas parece no tener nada que ver si la comparamos por ejemplo con la torta cruda 1, por lo que se considera correcto el resultado:

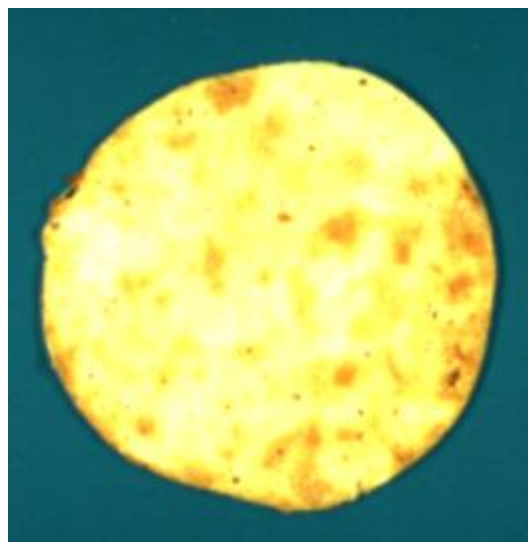


Figura 24: Torta cruda 1

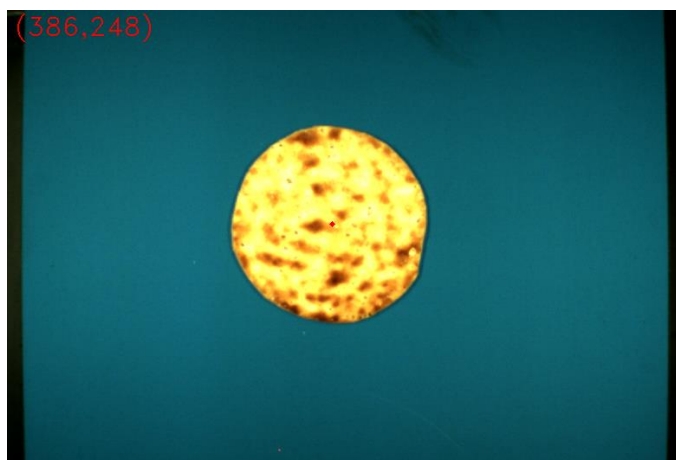


Figura 25: Torta indeterminada 12, resultado

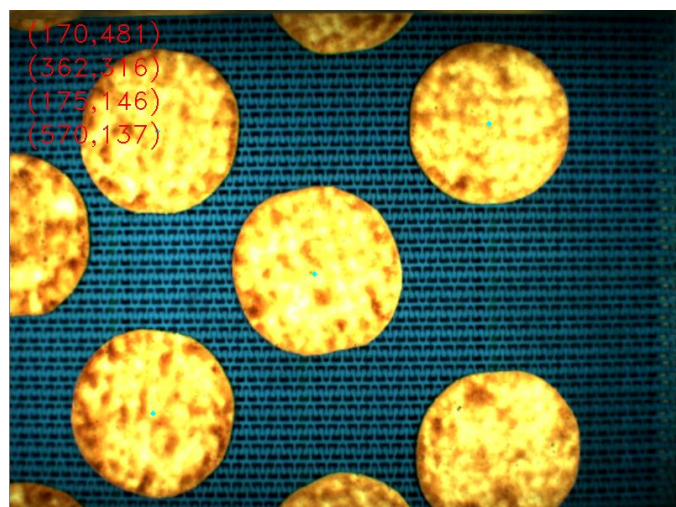


Figura 26: Torta ampliación 1 (Imagen06-22-2016 - 17.11.25.469)

7. Ampliación voluntaria

En esta ampliación hay que ubicar el centro de todas las tortas que salen totalmente completas en las imágenes.

Para la realización de esta ampliación se ha obtenido la imagen binaria de estas tortas y se ha utilizado como en los reconocimientos de formas anteriores, las funciones *Canny* y *findContours* para ello. Y posteriormente se ha utilizado la función *minEnclosingCircle* para hallar su círculo y obtener el centro de cada torta.

También se podía haber utilizado la función *HoughCircles*, pero no era tan precisa para calcular los centros. Y tras probarla en varias ocasiones, se desestimó.

Nombre de imagen	Número de tortas completas detectadas	Tiempo de procesamiento
Imagen06-22-2016 - 17.11.25.469	4	0.015
Imagen06-22-2016 - 17.11.47.569	4	0.014
Imagen06-23-2016 - 14.31.41.956	3	0.014
Imagen06-23-2016 - 14.33.12.250	4	0.014
Imagen06-23-2016 - 14.36.33.505	3	0.016

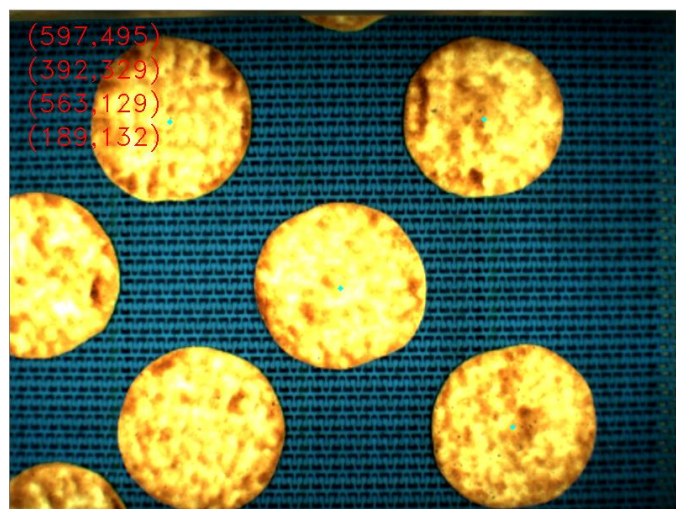


Figura 27: Torta ampliación 2 (Imagen06-22-2016 - 17.11.47.569)

En esta imagen de arriba se puede observar como aparentemente una de las tortas no ha sido seleccionada, pero esto es porque esta torta no está completa del todo y el contorno no se cierra, esto podía haber sido solventado fácilmente, pero se procuró ser rigurosos en esto:

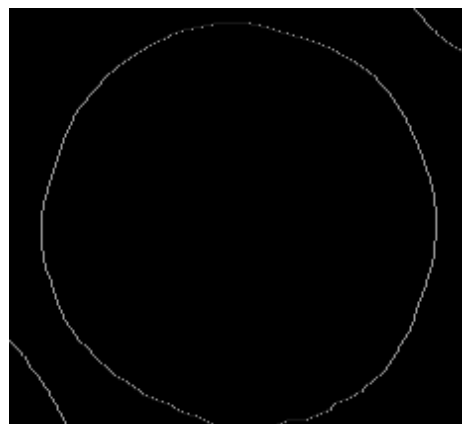


Figura 28: Torta no completa

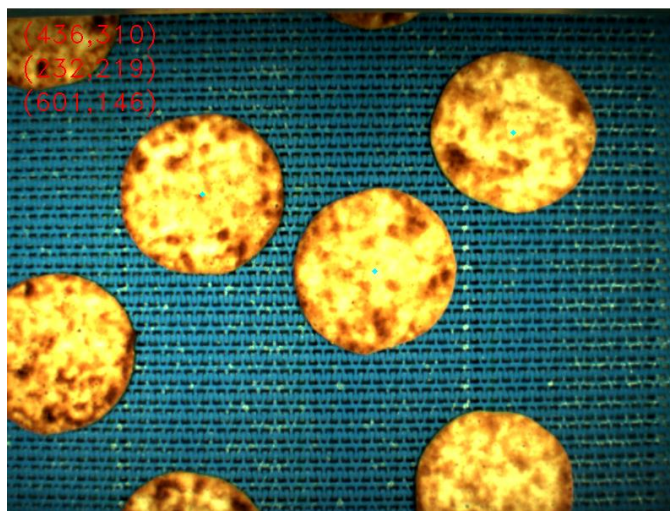


Figura 29: Torta ampliación 3 (Imagen06-23-2016 - 14.31.41.956)

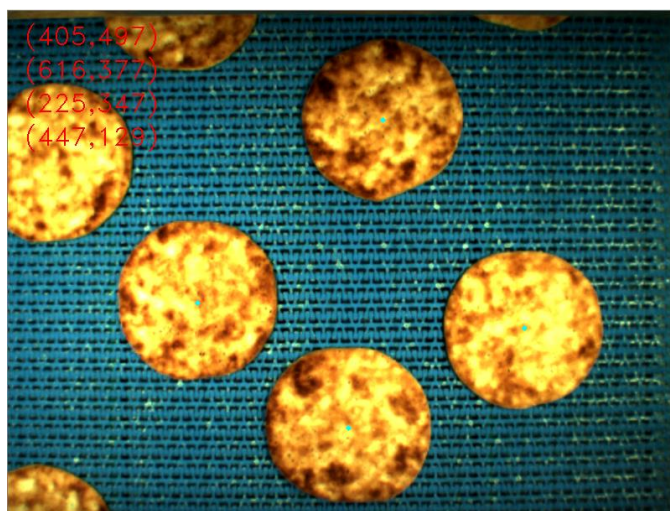


Figura 30: Torta ampliación 4 (Imagen06-23-2016 - 14.33.12.250)

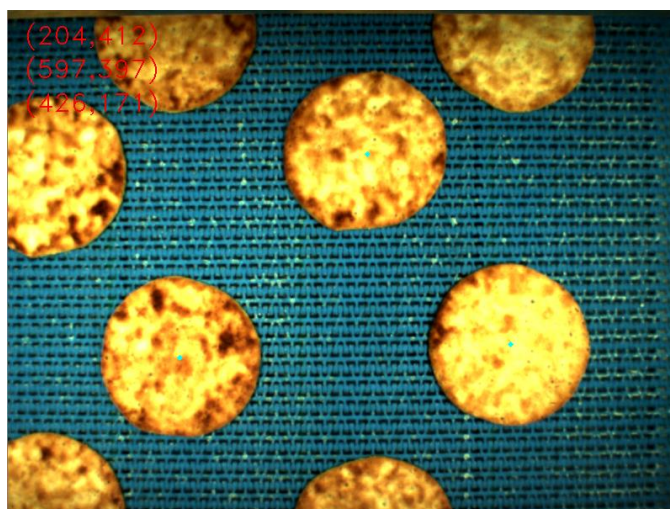


Figura 31: Torta ampliación 5 (Imagen06-23-2016 - 14.36.33.505)

8. Conclusiones

Este proyecto ha estado muy bien para aprender a trabajar con OpenCV ya que ha sido un proyecto muy realista. Lo malo que ha tenido es que dado lo apretado que está todo en la asignatura, junto que el resto de asignaturas del máster, se ha complicado en una tarea muy dura, ya que tampoco lo hemos podido avanzar a penas en clase y por tanto consultar dudas allí salvo un par de días. También hubo problemas con interrupciones que retrasaron muchísimo este trabajo.

A la hora de identificar las tortas, había casos simples que se podían hacer de varias maneras, pero por el contrario, por ejemplo, la identificación de las tortas rotas ha sido muy difícil para poder identificar las pequeñas muescas, no he encontrado ninguna función de OpenCV que ayudara a resolver esto fácilmente, ni con las áreas o perímetros de las tortas puesto que estas a veces son un poquitín más grandes o pequeñas, ni comprobando la circularidad porque a veces las muescas son muy pequeñas y no se detectan. También, en la ampliación que parecía bastante sencilla, como el fondo no es uniforme, ha sido complicado a la hora de distinguir la forma correctamente.

En conclusión, recomiendo que este trabajo se haga en próximos cursos, pero con más tiempo en clase para realizarlo, porque si no se complica bastante.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo de la UPV, especialmente gracias a Martín Mellado Arteché, profesor y responsable de la asignatura Automatización y robótica y Ángel Soriano, el mentor en el tema de visión artificial, y que además, ha ayudado a solventar algunos problemas importantes con el OpenCV.