

# APLICACIÓN DE VISIÓN ARTIFICIAL PARA LA CALIFICACIÓN AUTOMÁTICA DE PRUEBAS ESCRITAS

Freddy Bolaños Martínez<sup>1</sup>, Eliana Isabel Arango Zuluaga<sup>2</sup>, Mónica Aydé Vallejo Velásquez<sup>3</sup>

<sup>1</sup>Ph. D. en Ingeniería Electrónica. Profesor Asociado Universidad Nacional de Colombia. Grupo de Automática de la Universidad Nacional (GAUNAL). Medellín–Colombia. fbolanosm@unal.edu.co.

<sup>2</sup> Ph. D. en Ingeniería Electrónica. Profesora Asociada Universidad Nacional de Colombia. Grupo de Automática de la Universidad Nacional (GAUNAL). Medellín – Colombia.

<sup>3</sup> Ph. D. en Informática (C). Profesora Asistente Universidad Nacional de Colombia. Grupo de Automática de la Universidad Nacional (GAUNAL). Medellín–Colombia. mavallejov@unal.edu.co

## RESUMEN

Este artículo describe una aplicación de visión artificial para la automatización del proceso de calificación de exámenes escritos del tipo de selección múltiple. Una cámara digital es usada para capturar fotografías de las hojas de respuesta, que luego son adecuadas y procesadas con el fin de determinar la nota del estudiante. La adecuación de las imágenes se hace por medio de algoritmos heurísticos de filtrado y redimensionamiento de baja complejidad, con miras a portar la aplicación a una plataforma móvil en el futuro. Los resultados obtenidos son prometedores, con una tasa de aciertos de alrededor del 97 %, y con buena tolerancia a condiciones cambiantes de iluminación y orientación en la toma de las fotografías.

**Palabras clave:** Visión artificial; procesamiento digital de imágenes; automatización.

Recibido: 4 de Agosto de 2017. Aceptado: 21 de Junio de 2018

*Received: August 17th, 2017. Accepted: June 21st, 2018*

## AN ARTIFICIAL VISION APPLICATION FOR AUTOMATICALLY ASSESSING WRITTEN TESTS

### ABSTRACT

*This paper describes an artificial vision application aimed at the automation of the written exams grading process, for the multiple-choice exams kind. A digital camera is used for capturing images from the answer sheet, which are then adapted and processed for the sake of assessing the final grade of the test. The image processing is based on filters and resizing heuristic algorithms with low complexity, which aims at implementing the obtained application on a mobile platform. Obtained results are promising, with a success rate of about 97 %, and with a suitable tolerance regarding changing conditions of lighting and orientation of the taken images.*

**Keywords:** *Artificial vision; digital image processing; automation.*

*Cómo citar este artículo:* F. Bolaños Martínez, E. I. Arango Zuluaga, M. A. Vallejo Velásquez, "Aplicación de Visión Artificial para la Calificación Automática de Pruebas Escritas", Revista Politécnica, Año 14, vol. 14, no. 26, pp. 65-74, 2018. <https://doi.org/10.33571/rpolitec.v14n26a6>

## 1. INTRODUCCIÓN

En los últimos años, la masificación de la electrónica de consumo ha permitido la disponibilidad de dispositivos electrónicos de altas prestaciones, tales como teléfonos inteligentes, plataformas embebidas, y cámaras digitales. Esta disponibilidad facilita la incursión de la automatización y el control en aplicaciones cada vez más complejas, e incluso de naturaleza crítica [1]. Igualmente, se han venido popularizando los sistemas de sensado y automatización basados en procesamiento digital de imágenes y visión artificial [2]. En aplicaciones de automatización, el procesamiento de las imágenes permite, por ejemplo, la mejora de algunos de sus atributos (luminosidad, contraste, rango dinámico, entre otros). Usualmente, luego del acondicionamiento y la mejora de la imagen, existe una etapa de procesamiento de la misma, que se encarga de extraer información útil con diferentes propósitos (clasificación, monitoreo, alarmas, sensado, seguridad, identificación, etc) [3].

Uno de los elementos clave en los sistemas de visión artificial es la segmentación, que aparece de manera recurrente en múltiples aplicaciones [4, 5, 6]. La segmentación permite extraer secciones de interés de una imagen dada, por lo cual es posible eliminar información superflua, simplificar los algoritmos de visión artificial, y cambiar la representación de la información de interés, en otra más adecuada de acuerdo con cada problema. Otras operaciones de uso frecuente en el ámbito del procesamiento de imágenes y la visión artificial son la detección de bordes [6, 7, 8], el reconocimiento de patrones [9, 10], y la toma de decisiones con base en información visual del entorno [11, 12].

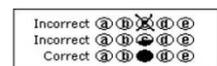
Este trabajo describe una aplicación heurística que procesa imágenes de hojas de respuesta tomadas contra un fondo contrastado. La aplicación se basa en operaciones simples de procesamiento y de visión artificial, para determinar la nota de la prueba a partir de las selecciones de las opciones que se han marcado en la hoja de respuesta. La simplicidad de las operaciones es una condición deseable en la aplicación, ya que se desea implementarla en una plataforma móvil, cuyo desempeño y consumo de potencia son críticos. En principio, la aplicación ha sido desarrollada en la plataforma Matlab [13], que por su facilidad de programación, permite una migración sencilla a otro

lenguaje o plataforma de desarrollo. El resto del documento está organizado de la siguiente manera: La Sección 2 describe la aplicación, su estructura y sus componentes principales. La Sección 3 muestra los resultados obtenidos con el uso de la aplicación y las métricas usadas para estimar su desempeño. Las conclusiones y trabajo futuro se presentan en la Sección 4.

## 2. DESCRIPCIÓN DE LA APLICACIÓN

La aplicación fue desarrollada usando el entorno de programación de Matlab, que usa como entrada una fotografía tomada a la hoja de respuestas del examen. La Figura 1 muestra el formato de la hoja usada para las pruebas de la aplicación. Como se verá más adelante, dicha plantilla puede cambiarse fácilmente por otra (por ejemplo por una con más opciones de respuesta) simplemente cambiando el proceso de entrenamiento de la aplicación, para reconocer las nuevas posiciones de las opciones de respuesta.

Examen de prueba



Rellene completamente la opción para la solución de su examen:

- |   |   |   |
|---|---|---|
| 1. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 21. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 41. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 2. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 22. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 42. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 3. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 23. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 43. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 4. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 24. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 44. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 5. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 25. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 45. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 6. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 26. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 46. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 7. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 27. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 47. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 8. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 28. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 48. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 9. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E  | 29. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 49. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 10. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 30. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 50. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 11. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 31. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 51. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 12. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 32. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 52. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 13. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 33. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 53. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 14. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 34. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 54. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 15. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 35. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 55. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 16. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 36. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 56. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 17. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 37. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 57. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 18. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 38. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 58. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 19. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 39. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 59. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |
| 20. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 40. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E | 60. <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="radio"/> D <input type="radio"/> E |

Figura 1. Plantilla usada como hoja de respuestas

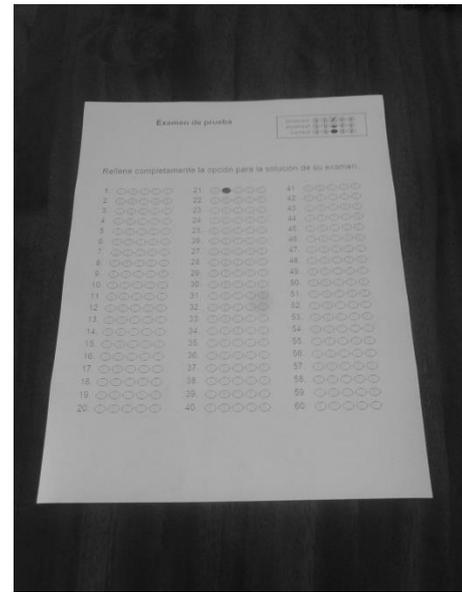
Las imágenes de las hojas de respuesta, similares a la plantilla que se muestra en la Figura 1, pueden ser tomadas en condiciones diversas (orientación, iluminación, resolución), pero siempre deben

aparecer contra un fondo contrastado. Esto es porque se pretende implementar la aplicación en una plataforma móvil, en la que no se pueden garantizar condiciones uniformes para la toma de las imágenes de entrada. La condición con respecto a que la hoja de respuestas esté contrastada contra el fondo, es consecuencia de la necesidad de eliminar la información superflua de la imagen (fondo), y para ello se requiere diferenciar muy bien la región de interés del resto de la fotografía.

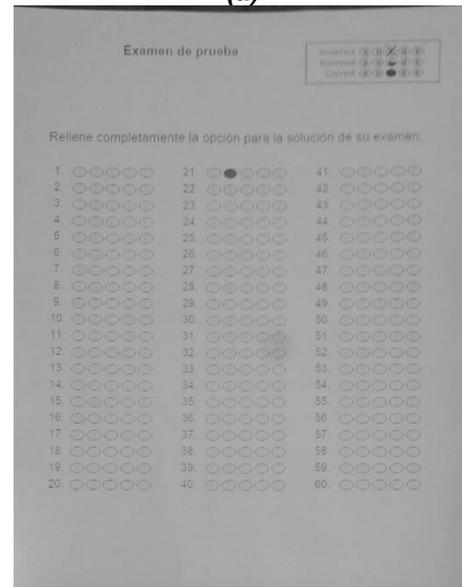
La aplicación puede dividirse en tres etapas fundamentales, cuya entrada corresponde a la imagen contrastada descrita anteriormente. En primer lugar, es necesario efectuar la extracción de la región de interés de la imagen contrastada. Dicha región se refiere a la sección de la imagen que corresponde exclusivamente a la hoja de respuestas. Al final de esta primera etapa, la imagen coincide exactamente con los bordes de la hoja de respuestas, de modo que es posible efectuar una segmentación y binarización de las subregiones de interés, asociadas con las respuestas marcadas en la plantilla. Finalmente, es necesario efectuar un proceso de comparación y detección, con el objeto de determinar si se ha marcado la opción correcta para cada pregunta. Cada una de las tres etapas descritas se explica en detalle a continuación.

**Extracción de la región de interés**

Esta podría considerarse la etapa crítica de la aplicación, ya que se elimina la información superflua del fondo contra el cual se toma la imagen. El propósito de esta etapa es lograr que los bordes de la imagen procesada y los de la hoja de respuesta (que contiene la información de interés) coincidan. La Figura 2 muestra un ejemplo de imagen antes (a) y después (b) del proceso de extracción.



(a)



(b)

**Figura 2. Ejemplo de la etapa de extracción. a) Antes de la extracción. b) Después de la extracción**

La etapa de extracción mostrada en la Figura 2 inicia con una binarización de la imagen de entrada. Dicha binarización consiste en la cuantificación de la información de la imagen a solo dos posibles niveles, representando la ausencia total de luminosidad (color negro asociado con el fondo), y la luminosidad completa (color blanco asociado con la imagen de interés). Debe tenerse en cuenta que

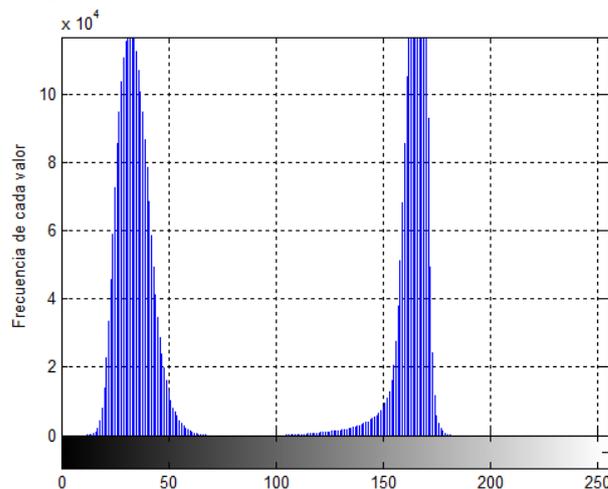
la binarización de la imagen se hace únicamente con el propósito de efectuar la extracción, ya que la información original de la fotografía es conservada para etapas posteriores. Uno de los formatos más usados para la representación de imágenes en color, es el RGB (del inglés: *Red, Green, Blue*), que usa 8 bits para cada una de sus componentes de color, es decir: Rojo (*R*), Verde (*G*) y Azul (*B*). Esto quiere decir que un solo pixel de la imagen puede tomar 1 entre 256 posibles combinaciones de color. Tal cantidad de información (alrededor de 16 millones de colores distintos) no es necesaria para establecer los límites de la imagen de interés (hoja de respuestas).

Por esta razón, la imagen en color es llevada a una representación de dos niveles (binaria) que permite diferenciar fácilmente qué parte de la imagen corresponde al fondo y qué otra parte de la misma corresponde a la hoja de respuestas. Dicho cambio en la cantidad de niveles discretos para una señal, es referido comúnmente como cuantificación [14], y en este caso se desarrolló en dos procesos sucesivos. En primer lugar la imagen en color fue transformada en una imagen en escala de grises, en donde solo es importante la información de luminosidad de cada uno de los pixeles. Si *R*, *G* y *B* son los valores de intensidad de color para un pixel determinado, entonces el valor del nuevo pixel (*P*) en escala de grises se puede determinar de la siguiente forma [15]:

$$P = \frac{R + G + B}{3} \quad (1)$$

Debido a que cada componente de color está representado como un número entero de ocho bits, el rango dinámico para *P* se encuentra entre los valores 0 y 255. Los valores de *P* cercanos a cero representan condiciones de poca luminosidad en la imagen, mientras que los valores cercanos al máximo (255), indican una cantidad importante de luminosidad. El proceso que lleva la imagen a una representación binaria con el propósito de hacer la extracción, debe establecer un umbral entre los dos valores extremos (0 y 255), con el fin de redondear (cuantificar) todos los pixeles de la imagen a estos dos posibles valores. Existen varios criterios reportados en la literatura para escoger el umbral apropiadamente [15]. En general, si la región de interés está muy contrastada contra la información superflua del fondo, la definición de umbral es relativamente robusta ante cambios en las condiciones en las que fue tomada la imagen. La

Figura 3 muestra lo que se conoce como Histograma, para la imagen de ejemplo mostrada en la Figura 2.a. Un histograma describe la frecuencia de aparición de cada uno de los posibles valores para los pixeles, lo que se logra estableciendo la cantidad de pixeles que aparecen para cada uno de los posibles valores en la imagen objeto de estudio. Se ha hecho un acercamiento sobre la figura del histograma de modo que se puedan diferenciar correctamente los dos lóbulos que aparecen en ella. Es decir, por razones de claridad en la figura y de escalamiento, el lóbulo de la derecha aparece recortado en sus valores máximos.

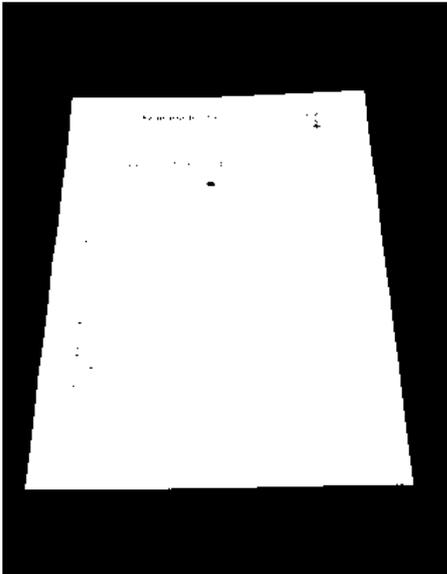


**Figura 3. Histograma de la imagen de la Figura 2.a**

El histograma de la Figura 3 es un gráfico que relaciona los 256 posibles valores de los pixeles de una imagen en escala de grises (eje X en la figura), con la frecuencia o cantidad de apariciones en la imagen de cada uno de dichos valores (eje Y en la figura). Por encontrarse muy contrastada, la imagen de la Figura 2.a. exhibe un histograma con dos lóbulos bien diferenciados. El lóbulo de la izquierda está centrado alrededor del valor 32 y se asocia con la información de poca luminosidad (fondo). El lóbulo de la derecha está centrado alrededor del valor 166 y está asociado con información de alta luminosidad (la hoja de papel o la región de interés).

Claramente la selección del umbral para la binarización se puede hacer escogiendo un punto intermedio entre los dos lóbulos descritos (el que representa el fondo de la imagen y el que representa la región de interés). Una vez seleccionado el valor del umbral, se cuenta con un

criterio simple para la cuantificación a dos niveles: Todos los píxeles cuyo valor en escala de grises supere el umbral, serán llevados a su valor máximo y serán considerados parte de la región de interés. Por su parte, todos los píxeles con valores por debajo del umbral serán considerados información del fondo y no se incluirán en la imagen de salida. La Figura 4 muestra los resultados del proceso de binarización para la imagen de la Figura 2.a.



**Figura 4. Imagen binarizada**

Una vez binarizada la imagen original, es necesario componer o sintetizar una nueva imagen a partir de la información que corresponde solamente a la hoja de respuestas. Para esto es necesario conocer las fronteras o los límites de dicha información dentro de la imagen cuantificada. Como se puede asumir razonablemente que la imagen de interés está compuesta por cuatro lados rectos, lo que se hace es determinar las posiciones de las esquinas del cuadrilátero que la hoja de papel representa en la imagen binaria. Una vez determinadas las esquinas de la región de interés, se pueden obtener las ecuaciones de las cuatro rectas entre cada par de puntos, lo que servirá como criterio para la extracción.

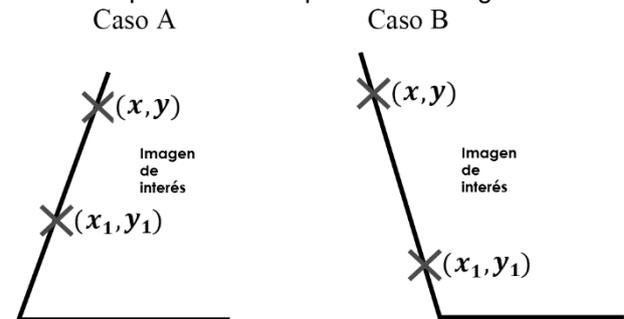
Para la detección de las posiciones de las esquinas de la imagen de interés se propone un algoritmo que se encarga de encontrar la esquina inferior-izquierda de la hoja de papel en la imagen binarizada. Dicho algoritmo luego se extrapolará para la detección de las demás esquinas de la región de interés, en donde por medio de un

cambio de coordenadas, ese mismo algoritmo se usa para la detección sucesiva del resto de las esquinas de la hoja. El algoritmo de detección busca el punto más abajo y más a la izquierda de color blanco dentro de la imagen binarizada. Como las esquinas de la imagen de interés pueden ser oblicuas, se desarrolló un algoritmo basado en deltas de desplazamiento en las dos direcciones, para hallar las coordenadas de las esquinas. La definición de dichos deltas se muestra a continuación:

$$\Delta x = x_1 - x \tag{2}$$

$$\Delta y = y_1 - y \tag{3}$$

En las Ecuaciones (2) y (3),  $(x, y)$  representa la posición del píxel que en un momento dado es considerado el de la esquina inferior-izquierda de la imagen de interés. Por otro lado, el par  $(x_1, y_1)$  representa un nuevo candidato a convertirse en la esquina de la imagen de interés. Claramente la búsqueda de la esquina inferior-izquierda de la imagen solo se hace sobre píxeles de alta luminancia (valor 255 o color blanco en la imagen binarizada), y se evalúan dos posibles casos para decidir si se debe considerar al nuevo píxel como el que corresponde a la esquina de la imagen. La Figura 5 ilustra los dos casos que son considerados en la búsqueda de la esquina de la imagen.

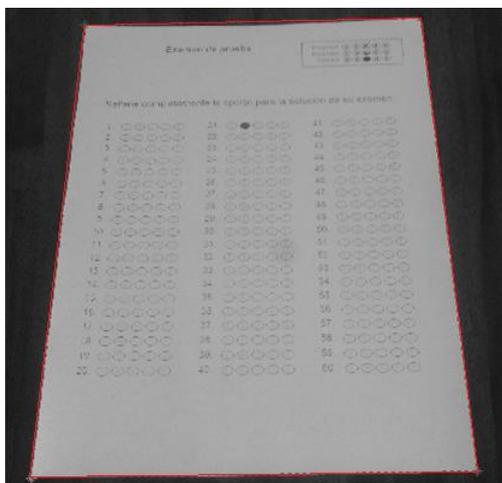


**Figura 5. Condiciones para la detección de esquinas**

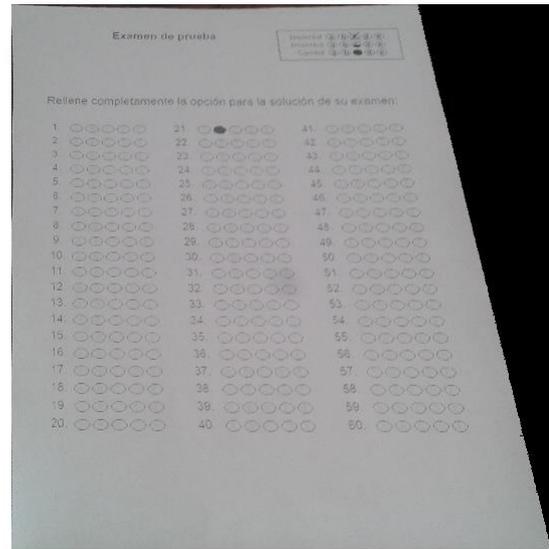
En la figura, el ejemplo del Caso A satisface simultáneamente que  $\Delta x \leq 0$  y que  $\Delta y \geq 0$ , por lo que se puede decir que el nuevo punto evaluado está más abajo y a la izquierda que el que era considerado el de la esquina de la imagen. En el Caso B se satisface al mismo tiempo que  $\Delta x \geq 0$  y  $\Delta y > 0$ , lo que sirve como una segunda condición para la detección de la esquina de la imagen de interés. Una vez establecidas las coordenadas de las 4 esquinas, es posible trazar rectas entre cada par de puntos, delimitando así la región de interés dentro de la imagen original. La Figura 6 muestra la superposición de la imagen de la Figura 2.a, con la

selección que algoritmo hizo de las esquinas y las rectas de los límites de la región de interés.

Una vez conocidas las ecuaciones de las rectas que hacen las veces de frontera entre el fondo de la imagen y la región de interés, la etapa de extracción culmina con una composición o síntesis de todos los píxeles de la imagen original que se encuentren dentro de los límites establecidos por dichas rectas. Como el cuadrilátero que conforma la hoja de respuestas puede estar ubicado de manera oblicua dentro de la imagen original, la imagen sintetizada puede tener algunas zonas oscuras, reflejando por ejemplo, que el ancho de la imagen de interés es diferente en diversas zonas de la imagen original. La figura 7 muestra el resultado del proceso de extracción. A partir de las ecuaciones de las rectas de frontera, el algoritmo extrae filas de píxeles de la imagen original y las acomoda sucesivamente en la imagen sintetizada. La figura también ilustra esas zonas en negro de la imagen extraída, resultado de la posición oblicua de la hoja de respuestas en la imagen original. El paso de la imagen mostrada en la Figura 6, a la Figura 7, se consigue por medio de una transformación de coordenadas, en donde la esquina superior izquierda de la segunda figura (coordenadas  $x = 1$ ,  $y = 1$ ), corresponde a la esquina de los bordes detectados en la etapa anterior (esquina de la hoja contrastada). Dicho cambio de variables se efectúa aprovechando las ecuaciones de las rectas obtenidas en el proceso de detección de bordes.



**Figura 6. Detección de bordes**



**Figura 7. Síntesis de la imagen extraída para la imagen de la Figura 2.a**

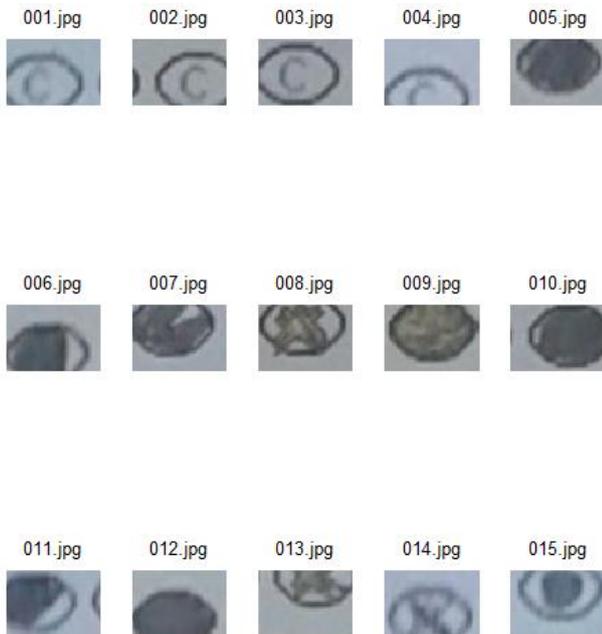
Para completar el proceso de extracción es necesario eliminar las zonas en negro de la imagen sintetizada. Esto se logra redistribuyendo de manera uniforme los píxeles que se encuentran en cada fila de la imagen sintetizada y rellenando los píxeles negros que quedan por medio de un proceso de interpolación. El resultado final de la extracción, puede visualizarse en la Figura 2.b.

### **Segmentación y binarización de las regiones de interés**

Debido al proceso de extracción explicado previamente, los tamaños de las imágenes con la información de interés dependerán de la posición de la hoja de respuestas dentro de la imagen original. Esto obliga a que las dimensiones de la imagen se operen de manera normalizada, para garantizar que independientemente de los resultados del proceso de extracción, las posiciones de cada uno de los óvalos de la hoja de respuesta, se puedan ubicar de manera unívoca.

El algoritmo para estimar la nota del examen se concentra en la región de la imagen extraída en donde está ubicado el óvalo de la respuesta correcta, es decir, el óvalo que debería estar marcado en la hoja de respuestas. Para cada una de las respuestas que deben ser evaluadas el algoritmo extrae la región o área de interés, para que la siguiente etapa determine si el óvalo ha sido marcado correctamente o no. Las coordenadas

normalizadas de la región de interés dentro de la imagen extraída se calculan promediando las posiciones del centroide de dicha región, para diferentes imágenes, tomadas en diferentes condiciones. El centroide se calcula como el promedio aritmético de las coordenadas de todos los píxeles que corresponden a la región de interés (el óvalo de la respuesta). La Figura 8 muestra las imágenes de interés obtenidas para la respuesta 21-C de 15 ejemplos diferentes de imágenes de entrada.



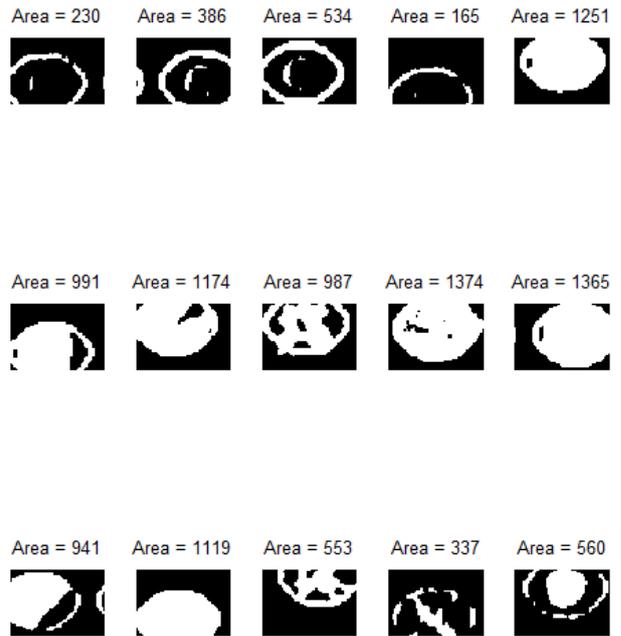
**Figura 8. Regiones de interés de la respuesta 21-C**

La Figura 8 muestra cómo a pesar de la variabilidad de las condiciones de entrada en las diferentes imágenes originales el método del centroide permite aislar de manera aproximada la posición de la opción que debe ser evaluada en la hoja de respuestas. En este caso la posición a evaluar en el algoritmo de detección corresponde a las posiciones promedio del centroide de cada una de las imágenes involucradas.

**Decisión con respecto a la respuesta**

Una vez aislada la región del óvalo, es necesario determinar si la opción asociada a dicha región ha sido seleccionada por quien presentó el examen. Para hacer esto se recurre nuevamente a la binarización de la región de interés y al cálculo de su negativo. De esta forma, el área de píxeles en

blanco de la región de interés se convierte en un criterio para decidir si el óvalo de la opción correcta ha sido debidamente llenado. Dicha área corresponde a la suma de todos los píxeles que aparecen en blanco en el negativo de la región de interés (si el óvalo ha sido rellenado, el negativo tendrá muchos valores de blanco). Una vez calculada la información de área una simple comparación contra un valor de umbral permite decidir si la respuesta fue rellenada de manera correcta. El umbral de decisión se calcula a partir de un área mínima (una mínima cantidad de píxeles rellenados en el óvalo), con lo que se considera que la opción ha sido sombreada por quien presentó la prueba. La Figura 9 ilustra el resultado de binarización para cada una de las regiones de interés de la Figura 8 y muestra los valores de área calculados en cada caso.



**Figura 9. Binarización y negativo de la región de interés**

La Figura 9 sugiere que escogiendo un valor umbral de alrededor 900 píxeles de área, es posible tomar la decisión apropiada con respecto a si se ha rellenado o no la respuesta correcta. Aquellos valores de área por debajo del umbral se tomarán como respuesta incorrecta, en caso contrario se asumirá que se ha llenado la respuesta correctamente.

Debe recordarse que las imágenes extraídas de la primera etapa pueden tener diferentes tamaños y que las coordenadas del centro de la región de interés están normalizadas con respecto a la cantidad de filas y de columnas de la imagen. Por esta razón y para que el criterio de decisión por umbral sea robusto, es necesario que la región de interés siempre tenga el mismo tamaño, independientemente de las dimensiones de la imagen extraída. Lo anterior explica por qué todas las imágenes correspondientes a la zona de interés son redimensionadas a un tamaño de 40 filas por 55 columnas de píxeles, de modo que el uso del área como criterio de decisión no dependa de la imagen de entrada original.

### 3. RESULTADOS OBTENIDOS

Como ya se había mencionado, la implementación de la aplicación se hizo en la plataforma Matlab, versión 7.8 (R2009a), ejecutándose sobre un equipo Intel Core I7 con 6 GB de memoria RAM y sistema operativo Windows 8.1. En cuanto a las tres etapas referidas anteriormente, la de extracción de la imagen de interés es la que consume una cantidad de tiempo considerable comparada con las otras dos, y por lo tanto es la que condiciona el desempeño de la aplicación, en términos del tiempo de respuesta de la aplicación.

Dado que el tiempo que consume la etapa de extracción depende de la resolución de la imagen de entrada y de condiciones tales como la posición y el tamaño de la hoja de respuestas dentro de la imagen original, perfilar el desempeño de la aplicación puede resultar difícil. Sin embargo, se logró determinar que para una imagen de 8 Megapíxeles (la máxima resolución con la que se ensayó la aplicación), el tiempo que toma la extracción de la imagen de interés es de alrededor de 10 segundos. Esta medida es un valor promedio obtenido usando las rutinas de medida de tiempo de ejecución que ofrece la plataforma Matlab.

En cuanto a la etapa de extracción en sí misma, se logró determinar que siempre que el fondo estuviera bien diferenciado de la imagen de la hoja, es posible extraer la imagen de interés sin errores. Para determinar en qué casos puede fallar la aplicación, se hicieron pruebas con diferentes combinaciones de colores entre la hoja de

respuestas y el fondo. El resultado de esta sucesión de pruebas se muestra en la Tabla 1.

**Tabla 1. Resultados obtenidos para diferentes situaciones de contraste**

Caso	Contraste (Fondo / Hoja)	Extracción de la información
1	Cian/Amarillo	Incorrecta
2	Cian/Blanco	Correcta
3	Azul/Amarillo	Correcta
4	Azul/Blanco	Correcta
5	Azul/Verde	Correcta
6	Violeta/Naranja	Correcta
7	Violeta/Amarillo	Correcta
8	Violeta/Blanco	Correcta
9	Verde/Amarillo	Correcta
10	Verde/Blanco	Correcta
11	Naranja/Rojo	Incorrecta
12	Naranja/Blanco	Correcta
13	Naranja/Amarillo	Incorrecta
14	Rojo/Blanco	Correcta

Con respecto a las etapas de segmentación de la región de interés y de la decisión, se usaron un total de 45 imágenes de entrada (15 hojas de respuesta, 3 imágenes diferentes por cada una). Para la selección del umbral de binarización y del centroide de la posición de interés en la imagen extraída, se efectuó un proceso de entrenamiento sobre 15 del total de las imágenes. Dicho proceso de entrenamiento equivale a escoger los mejores valores de comparación para binarizar la imagen, y las coordenadas más adecuadas para ubicar el centro del óvalo en cada una de las opciones a considerar en la hoja de respuestas. Al final, el umbral de binarización y la posición del centro de la región de interés corresponden a los valores promedio de los datos del entrenamiento.

En cuanto a la efectividad de la aplicación para detectar de manera correcta las respuestas de la hoja, del total de 45 imágenes de entrada originales, se detectaron 24 positivos verdaderos, 11 negativos verdaderos, 9 falsos negativos y 1 falso positivo. A partir de una modificación de la definición dada en [16], se puede calcular la sensibilidad de la aplicación como:

$$S = \frac{VP + VN}{VP + VN + FP + FN} \quad (4)$$

Donde  $VP$  corresponde a los casos de detección verdaderos positivos,  $VN$  corresponde a los

verdaderos negativos y *FP* y *FN* corresponden a los casos falsos positivos y falsos negativos, respectivamente. En estas condiciones y con los datos descritos anteriormente, el cálculo de la Ecuación (4) da como resultado una sensibilidad de 77.78 %. Sin embargo, debe tenerse en cuenta que los casos de falsos negativos corresponden todos a situaciones como las mostradas en las últimas tres imágenes de las Figuras 8 y 9, en donde el óvalo de la opción correcta no se llenó completamente. Dado que el criterio para decidir si se ha marcado o no la opción de la respuesta correcta, está basado en el área oscura de la imagen, se requiere, como es usual en estos casos, que quien diligencia la hoja de respuestas rellene completamente los óvalos de las opciones que desea marcar. Bajo estas restricciones no habría ningún caso de falso negativo y la sensibilidad se eleva a 97.78 %.

#### 4. CONCLUSIONES

Se ha desarrollado una aplicación en visión artificial para la automatización del proceso de calificación de exámenes basados en hojas de respuestas de selección múltiple. Los algoritmos usados en dicha aplicación son bastante simples, lo que facilita una implementación en diferentes plataformas, incluyendo las plataformas móviles. La sensibilidad, definida como la cantidad de decisiones correctas sobre el total de pruebas puede llegar casi al 98 %, si se aplican las restricciones regulares para el llenado de este tipo de hojas de respuesta.

Sin embargo, la sensibilidad de la solución propuesta podría mejorarse aún más, si se adoptan ciertos mecanismos simples, que se mencionan a continuación como trabajo futuro, y que serán explorados en una versión posterior de la aplicación descrita:

- Sugerir un protocolo de uso de la aplicación en donde se regularicen las condiciones de la toma de las imágenes (por ejemplo, adoptar una distancia cuasi constante para la toma de las imágenes, o usar marcadores de alto contraste para los óvalos).
- A partir de las coordenadas estimadas para la posición del óvalo a evaluar, efectuar una etapa adicional de procesamiento que considere el área del óvalo completo.
- Establecer un valor de umbral de decisión adaptable para establecer si el óvalo fue rellenado de manera correcta.

- Operar con diferentes resoluciones de la imagen en diferentes etapas del proceso, con el objeto de mejorar el desempeño de la aplicación y los tiempos de respuesta.
- Finalmente, se planea la implementación de una aplicación ejecutable en una plataforma para dispositivos móviles y además efectuar un perfilado del desempeño de la aplicación, de modo que se pueda estimar el costo en tiempo de procesamiento de las imágenes a partir de su tamaño o resolución.

#### 5. AGRADECIMIENTOS

Los autores agradecen a la Universidad Nacional de Colombia, Sede Medellín, por su apoyo para la realización del presente trabajo. Igualmente se les agradece a los ingenieros Leonardo Restrepo y Carlos Andrés Ruiz, por su colaboración en la primera etapa del desarrollo de la aplicación.

#### 6. REFERENCIAS BIBLIOGRÁFICAS

- [1] Sangiovanni-Vincentelli, A., "Quo Vadis, SLD? Reasoning About the Trends and Challenges of System Level Design," in Proceedings of the IEEE , vol.95, no.3, pp.467-506, March 2007.
- [2] Samad, T., "Building Control and Automation Systems," in Perspectives in Control Engineering Technologies, Applications, and New Directions , 1, Wiley-IEEE Press, 2001, pp.393-416.
- [3] Gonzales R., Woods R., "Digital imagen Processing", Addison-Wesley Publishing Company, Massachusetts 2nd Edition 2002.
- [4] Xiaolian Deng, Yuehua Huang, ShengQin Feng, Changyao Wang, "Adaptive threshold discriminating algorithm for remote sensing image corner detection", Image and Signal Processing (CISP), 2010 3rd International Congress on , vol.2, no., pp.880,883, 16-18 Oct. 2010.
- [5] Chernuhin, N.A, "On an approach to object recognition in X-ray medical images and interactive diagnostics process", Computer Science and Information Technologies (CSIT), 2013 , vol., no., pp.1,6, 23-27 Sept. 2013.
- [6] Xin Sun, Xiaoxiao Wang, "Study of edge detection algorithms for lung CT image on the basis of MATLAB", Control and Decision Conference (CCDC), 2011 Chinese , vol., no., pp.810,813, 23-25 May 2011.
- [7] Pham-Minh-Luan Nguyen; Jae-Hyun Cho; Sang Bock Cho, "An architecture for real-time hardware co-simulation of edge detection in image processing using Prewitt edge operator," in Electronics, Information and Communications ICEIC), 2014

- International Conference on , vol., no., pp.1-2, 15-18 Jan. 2014.
- [8] Kasi, M.K.; Rao, J.B.; Sahu, V.K., "Identification of leather defects using an autoadaptive edge detection image processing algorithm," in High Performance Computing and Applications (ICHPCA), 2014 International Conference on , vol., no., pp.1-4, 22-24 Dec. 2014.
- [9] Wang, H.; Peng, D.; Wang, W.; Sharif, H.; Wegiel, J.; Nguyen, D.; Bowne, R.; Backhaus, C., "Artificial Immune System based image pattern recognition in energy efficient Wireless Multimedia Sensor Networks," in Military Communications Conference, 2008. MILCOM 2008. IEEE , vol., no., pp.1-7, 16-19 Nov. 2008.
- [10] Fujun Ren; Xinhua Zhang; Long Wang, "A new method of the image pattern recognition based on neural networks," in Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on , vol.7, no., pp.3840-3843, 12-14 Aug. 2011.
- [11] Massari, M.; Ceriani, E.; Rigolin, L.; Bernelli-Zazzera, F., "Optimal path planning for planetary exploration rovers based on artificial vision system for environment reconstruction," in Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on , vol., no., pp.987-992, 24-28 July 2005.
- [12] Cesetti, A.; Frontoni, E.; Mancini, A.; Zingaretti, P.; Longhi, S., "Vision-based autonomous navigation and landing of an unmanned aerial vehicle using natural landmarks," in Control and Automation, 2009. MED '09. 17th Mediterranean Conference on , vol., no., pp.910-915, 24-26 June 2009.
- [13] Mathworks Inc. MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States. 2013.
- [14] Proakis J. G.; Manolakis D. G., "Digital Signal Processing (3rd Ed.): Principles, Algorithms, and Applications". Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 1996.
- [15] González, R. Woods, R. "Tratamiento Digital de Imágenes". Ed. Addison. Wessley. USA. 1996.
- [16] Koydemir, H. C.; Gorocs, Z.; Tseng, D.; Cortazar, B.; Feng, S.; Chan, R. Y. L.; Burbano, J.; McLeod, E.; Ozcan, A. "Rapid imaging, detection and quantification of Giardia lamblia cysts using mobile-phone based fluorescent microscopy and machine learning", in Lab On Chip Journal, Vol 15, pp. 1284 – 1293. 2015.