Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Trabajo de Fin de Máster
Máster en Ingeniería de Sistemas e Informática

Reconocimiento robusto de texto en imágenes de dispositivos móviles

Ana Belén Cambra Linés

Directora: Ana Cristina Murillo Arnal

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza
Septiembre de 2013

RESUMEN

El procesamiento automático de imágenes tiene gran interés en el desarrollo de nuevas tecnologías y aplicaciones basadas en información visual. Hasta hace poco, estas tareas han estado limitadas a realizarse en ordenadores con gran capacidad de cómputo, debido a los altos requerimientos de los algoritmos utilizados. Sin embargo, estas limitaciones van desapareciendo gracias a las últimas generaciones de teléfonos móviles, los *smartphones*, que poseen capacidades de procesamiento mucho más altas. En particular, dentro del campo de la visión artificial y en particular en temas de reconocimiento automático, una tarea que se ve muy beneficiada de la portabilidad a dispositivos móviles es la detección y reconocimiento de texto, ya que se han generado nuevos ámbitos de aplicación.

Con este trabajo de fin de máster se propone mejorar un sistema base existente de reconocimiento de texto en imágenes. El sistema base consiste en una aplicación para móviles capaz de extraer el texto de carteles rectangulares presentes en una fotografía capturada con el móvil. Actualmente existen muchos reconocedores de caracteres, llamados *OCRs* (del inglés *Optical Character Recognition*), que permiten extraer el texto de una imagen pero sus buenos resultados están muy condicionados a cómo se presenta el texto dentro de dicha imagen. Se requiere que el usuario enfoque con mucha precisión dónde se encuentran los textos a leer. Esta situación es una gran restricción y sobretodo muy poco realista y robusta, además de no permitir aprovechar estas tecnologías para, por ejemplo, dar servicios a personas con problemas de visión. Aunque el prototipo tomado como base para este trabajo consigue mejorar los resultados obtenidos por un *OCR* convencional, sigue presentando limitaciones para el uso en escenarios generales. En particular, se va a realizar una evaluación exhaustiva del prototipo, y se va a diseñar e implementar mejoras que reduzcan las limitaciones actuales que presenta, para conseguir un reconocimiento más robusto.

Dado que el campo donde se enmarca este trabajo es una rama activa dentro de la visión artificial, han aparecido nuevos enfoques dentro del reconocimiento de texto que obtienen mejores resultados que los tradicionales *OCRs*. Por ello, también se va a diseñar y evaluar la integración de este tipo de enfoques con el trabajo realizado.

Los resultados obtenidos han sido satisfactorios, consiguiendo mejorar el prototipo base. También la evaluación realizada del proceso demuestra que éste consigue mejorar los resultados de otros OCRs existentes, además de mejorar, en determinados casos, los resultados de otras técnicas de extracción de texto más modernas. Con parte de estos resultados se redactó el siguiente articulo: *"Towards robust and efficient text sign reading from a mobile phone"* que fue publicado en el *2nd IEEE Workshop on Mobile Vision* llevado a cabo junto con el *ICCV 2011*.

Índice general

1. Introducción **1**
1.1. Introducción 1
1.2. Trabajo relacionado 1
1.3. Objetivos 3
1.4. Sistema Base 3
1.4.1. Problema a resolver 4
1.4.2. Prototipo base 5
1.5. Estructura de la memoria 7

2. Reconocimiento robusto **8**
2.1. Generación de hipótesis 9
2.1.1. Inserción de rectas 10
2.1.2. Agrupamiento de esquinas 11
2.2. Evaluación de la probabilidad 12
2.3. Proyección de las hipótesis rectangulares 13
2.4. Selección de hipótesis finales 15

3. Evaluación y comparación **17**
3.1. Evaluación de las mejoras 17
3.2. Comparación con otros *OCRs* 18
3.3. Nuevos enfoques 19

4. Conclusiones **23**
4.1. Conclusiones 23
4.2. Trabajo futuro 24

Bibliografía **26**

Anexos **28**

A. Análisis de tiempo de procesamiento del OCR **28**

B. Publicación de los resultados **30**

C. Detalles de las pruebas realizadas **39**
C.1. Imágenes utilizadas 39

Capítulo 1
Introducción

1.1. Introducción

El procesamiento automático e inteligente de datos ha tenido y sigue teniendo un gran interés dentro del campo científico-técnico. En concreto, la rama dedicada al campo de la visión artificial se ha visto muy beneficiada por la aparición de una amplia gama de potentes dispositivos móviles, proporcionándole un nuevo campo de aplicación. Hasta hace poco, las tareas de procesado automático e "inteligente" de imágenes estaban destinadas a realizarse en ordenadores, debido a los requerimientos de cálculo de los algoritmos utilizados. Pero ahora, dada la calidad de las cámaras integradas y la creciente mejora de la capacidad de cómputo de los *smartphones*, se ha producido un gran aumento en el desarrollo de aplicaciones que realizan tareas relacionadas con la visión artificial.

Más concretamente este trabajo pretende estudiar y mejorar un sistema ya existente que permite extraer texto de señales que aparecen en imágenes capturadas desde un teléfono móvil. El sistema base consiste en un prototipo desarrollado para *iOS*. Para más detalle de sistema base utilizado como punto de partida para este trabajo consultar la sección 1.4 de este mismo documento.

Hay que destacar que el propósito de este trabajo no se centra en transformar el prototipo en una aplicación final. Por un lado se pretende realizar una definición más formal del proceso diseñado así como estudiar y mejorar varios aspectos detallados más adelante. Por otro lado, se pretende comparar el proceso con otras técnicas existentes en el estado del arte dentro del campo de la detección de texto en escenas generales.

1.2. Trabajo relacionado

En el campo del reconocimiento de texto, los métodos basados en *OCR* [1] fueron propuestos hace mucho tiempo pero, todavía se siguen encontrando nuevas contribuciones ampliando: su campo de aplicación, robustez o aumentando su rendimiento. Inicialmente las técnicas de los *OCRs* estaban orientadas a digitalizar el texto procedente de papel. Hoy en día, hay un montón de aplicaciones en las que sería muy útil poder reconocer el texto en imágenes generales. Por ejemplo, en el caso particular de la lectura para la asistencia,

en la literatura se encuentran varias propuestas dirigidas a desarrollar aplicaciones para discapacitados visuales [2, 3, 4, 5]. Sin embargo, todavía encontramos pocas de estas ideas integradas en los teléfonos móviles, en parte, debido al alto costo computacional de algunas de las propuestas. En este trabajo, se analizan qué técnicas de procesamiento de imágenes pueden ser adecuadas para la mejora de los resultados de este tipo de aplicaciones que se ejecutan en un teléfono móvil. Gracias a la explosión del desarrollo de aplicaciones móviles ya se encuentran aplicaciones comerciales que realizan tareas de visión artificial, como wordlens[1] , que lee texto en una aplicación de realidad aumentada.

Los *OCRs* suelen requerir que el texto presente en la imagen este lo más frontal, cercano y centrado posible, para tener un buen resultado a la hora de extraer el texto, presentando malos resultados en caso contrario, como puede verse en la Figura 1.1. Las investigaciones relacionadas se pueden dividir en dos grupos diferentes. Por un lado, hay un grupo de obras dirigidas hacia una mejora en sí misma del reconocimiento de caracteres, proponiendo nuevas técnicas para los *OCRs*. Incluso nos encontramos con obras capaces de identificar texto muy deformado [6] tal y como se presenta en servicios web como los *CAPTCHAs*.

Por otro lado, hay obras que proponen cómo se pueden "pre procesar" las imágenes de escenas generales para detectar que áreas tienen mayor probabilidad de contener texto y cómo reconocerlo. Tratar de reconocer texto en imágenes sin restricciones es un campo activo de investigación, con conferencias y concursos específicos. Existen trabajos que proponen trabajar con segmentos de la imagen: dividiendo ésta en componentes conectadas para después aplicar diferentes filtros de textura a la imagen [7]; o clasificando a nivel de pixel, como se propone en [8], usando valores de histogramas y gradientes de una región alrededor de cada píxel. Ciertos trabajos intentan detectar texto en vídeos, aprovechando información temporal o técnicas de *tracking* [9], [10]. Otros resultados interesantes hacia la detección del texto en imágenes generales se basan en diferentes técnicas de aprendizaje y clasificación, por ejemplo usando un clasificador basado en AdaBoost [11] o entrenando varios clasificadores SVM [12], para determinar qué regiones tienen mayor probabilidad de contener texto. El sistema base utilizado está relacionado con las propuestas que intentan pre-procesar las imágenes para obtener regiones de interés, que posteriormente pueden ser procesadas con las técnicas estándar de los *OCRs*. Hay dos trabajos muy cercanos a este enfoque [13, 14]. Ambos, utilizan diferentes pasos para corregir la orientación o perspectiva basándose en restricciones geométricas, pero ninguno de ellos integra todos los pasos requeridos en el proceso ni demuestra que son factibles para ejecutarse en un teléfono.

Algunos trabajos recientes proponen pasar el uso de las técnicas típicas de los *OCRs* a técnicas generales de reconocimiento de objetos adaptadas para reconocimiento de texto [15, 16]. El trabajo de [15] propone, una vez que se ha ejecutado el reconocimiento de caracteres sobre una región de interés, tener en cuenta la probabilidad de formar palabra.

[1] http://questvisual.com/

Los autores en [16] presentan un enfoque conjunto de localización y reconocimiento de texto que considera los caracteres como regiones MSER y propone sistemas de entrenamiento con fondos sintéticos. La fase de evaluación del sistema que se propone en este trabajo también tiene en cuenta estos nuevos enfoques.

1.3. Objetivos

El sistema base permite detectar desde la cámara de un móvil, automáticamente, donde hay carteles en una imagen y reconocer su texto mediante un sistema *OCR* convencional. Aunque este prototipo consigue mejorar los resultados obtenidos con *OCRs* convencionales, sigue presentando limitaciones para el uso en escenarios generales. En particular, se van a estudiar estos problemas: el tiempo de ejecución en algunos casos es muy elevado, el método de detección de los carteles produce muchos falsos positivos y el resultado de los *OCRs* convencionales no es correcto pese a haber detectado la existencia del cartel. En este trabajo fin de máster se va a realizar una evaluación más exhaustiva del prototipo y diseñar e implementar mejoras sobre estas limitaciones.

En más detalle, las tareas a realizar son:

- Diseñar un nuevo algoritmo que mejore la detección y evaluación de las hipótesis de los carteles detectados.
- Realizar una formalización del proceso de clasificación y evaluación de hipótesis, para presentar un análisis de probabilidades en lugar de decisiones heurísticas.
- Realizar un estudio exhaustivo de las mejoras del proceso diseñado, comparando con *OCRs* más potentes aunque no estén disponibles para integrarse en la plataforma móvil.
- Diseñar y evaluar la integración del método de pre-procesamiento diseñado con técnicas más recientes de detección/extracción de texto basadas en métodos generales de reconocimiento de objetos. Analizar si estas técnicas son más adecuadas para escenas generales que las técnicas específicas de los *OCRs* (orientadas a documentos escaneados, manuscritos,...).
- Redactar un documento/artículo de investigación para difundir los resultados.

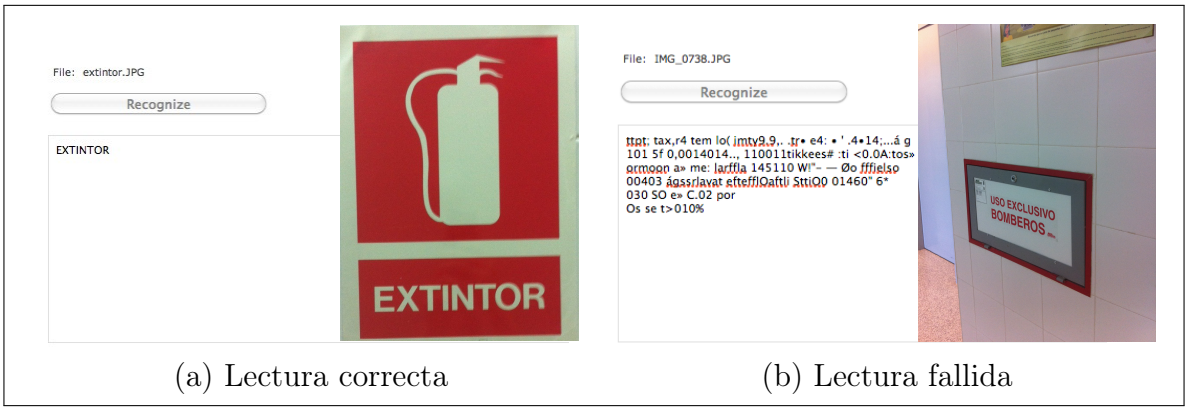
1.4. Sistema Base

El sistema utilizado como base [17] es un prototipo cuyo objetivo general era desarrollar una aplicación que fuera capaz de extraer el texto de carteles rectangulares a partir de una fotografía general de una escena que lo contenga. El proceso se realiza por completo en el propio móvil: desde la captura o elección de la imagen hasta la obtención del texto procedente del cartel, o carteles, existentes en ella.

En primer lugar el usuario adquiere una nueva imagen, y a continuación, se ejecutan diferentes pasos analizando las formas geométricas y su textura, todo esto de forma transparente para el usuario. Finalmente se obtiene el análisis de un *OCR* de las hipótesis con mayor probabilidad de contener texto. Este resultado se da al usuario a continuación, bien leído en voz alta o procesado por un diccionario en línea.

1.4.1. Problema a resolver

Actualmente hay disponibles muchos sistemas capaces de reconocer los caracteres presentes en una imagen, denominados *OCRs* (del inglés *Optical Character Recognition*). Típicamente, estos sistemas presentan un bajo rendimiento en imágenes donde el texto presenta distorsiones provocadas por la perspectiva o cuando no se encuentra correctamente alineado y centrado dentro de la imagen, como puede verse en la Figura 1.1. Para ilustrar esta situación se ha utilizado un *OCR* gratuito disponible en la red[2]. En la primera imagen, el texto del cartel está totalmente de frente, bien encuadrado y sobre un fondo liso, en este caso un folio. Sin embargo, en la segunda aparece un cartel que debido a la inclinación, perspectiva y el "ruido" del fondo (debido al resto de objetos de la imagen), contiene un texto ilegible para el *OCR*. Los resultados de ejecutar el mencionado *OCR* con estas imágenes pueden verse también en la misma figura.



(a) Lectura correcta (b) Lectura fallida

Figura 1.1: (a) Resultados obtenidos por el *OCR* usando un cartel de cerca, frontal y encuadrado y (b) otra desde una mayor distancia, donde el cartel sufre una deformación debido a la perspectiva

Con este simple ejemplo, se observa que los *OCRs* actuales funcionan perfectamente cuando los caracteres a reconocer se encuentran en las condiciones esperadas pero no situaciones generales.

El problema a resolver en el proyecto base se centró en diseñar y desarrollar un proceso mediante el cual, se consiguiera aumentar el rendimiento de un *OCR* ya existente,

[2] *OnlineOCR*: `http://www.onlineocr.net/`

ampliando su campo de aplicación y así poder utilizarlo para extraer texto en imágenes de escenas generales. Para ello se propuso obtener de forma automática aquellas regiones de interés con una mayor probabilidad de contener texto.

En la sección 1.4.2 se describe brevemente los pasos que componen el proceso diseñado y desarrollado.

1.4.2. Prototipo base

En esta sección se resume el proceso que sirve de base para todo el estudio y pruebas realizadas en este trabajo para poder explicar las mejoras realizadas a continuación. La Figura 1.2 recoge los distintos pasos que componen el proceso diseñado. Para más detalle sobre cualquier paso se puede consultar [17].

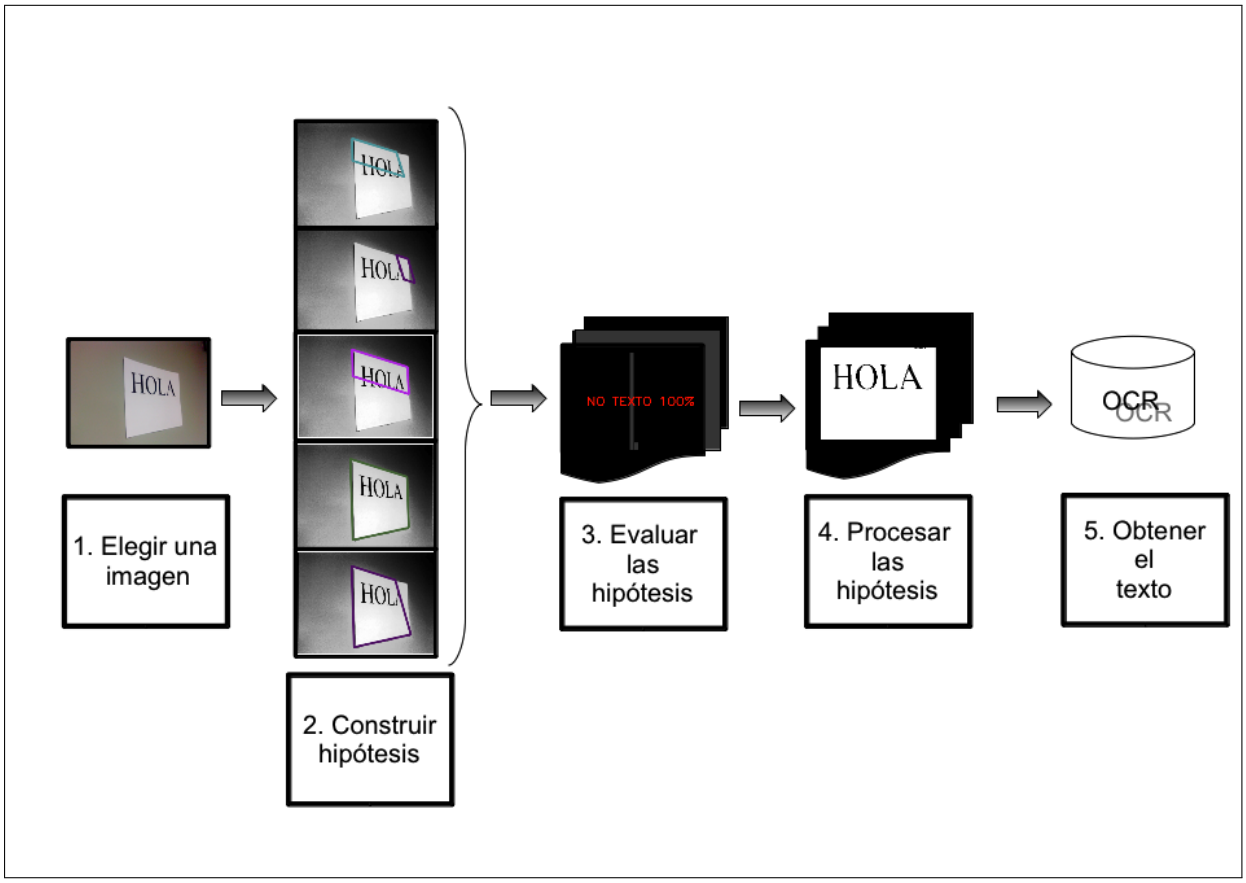


Figura 1.2: Esquema del proceso desarrollado

Búsqueda de rectángulos. Uno de los objetivos ya mencionado de este proceso era liberar al usuario de la imposición de que la imagen a leer, contenga un texto de frente y perfectamente encuadrado. Además, a la hora de diseñar este proceso se adoptó como

restricción que el texto debía estar contenido en un cartel con forma rectangular, dado que la mayoría de los carteles existentes presentan esta forma. Por lo tanto, al desconocer a priori la ubicación del cartel a procesar, y poder ser ésta cualquiera dentro de la imagen, el primer paso es buscar su posible ubicación en ella.

El proceso implementado para detectar los rectángulos consiste en obtener rectas que aparecen en la imagen para a partir de ellas, encontrar esquinas y así, construir hipótesis de posibles rectángulos intentando agruparlas. Con este proceso, se consigue reducir la zona a analizar de la imagen, intentando procesar sólo aquella que puede ser de interés.

Evaluación de las hipótesis. Junto con la posibilidad de que no todas las rectas y esquinas hayan podido ser detectadas correctamente, también es necesario barajar la posibilidad de que no todas las detectadas son de interés. Por tanto, es muy probable que las hipótesis generadas se hayan obtenido de rectas procedentes de otros elementos de la escena como puertas, ventanas, paredes de baldosas o ladrillo..., en lugar de las que corresponden con los bordes del cartel, que son las que interesan. Por esto, es necesario desarrollar un proceso por el cual se pueda evaluar qué hipótesis de rectángulos son más probables de corresponder a un cartel y contener texto. Para realizar esta evaluación, se han diseñado una serie de filtros para analizar el contenido dentro de cada hipótesis rectangular, para determinar si es probable que éste contenga texto o no.

Procesado de hipótesis aceptadas. Llegado a este punto del proceso, se dispone del grupo de hipótesis de rectángulos encontrados en la imagen cuyo contenido es muy probable que sea texto. Con el objetivo de conseguir mejorar el reconocimiento, es necesario procesar estas hipótesis.

Para ejecutar también una rectificación basada en homografías , el paso básico de esta propuesta es la generación de hipótesis rectangulares en la imagen.

Binarización de la imagen. Como paso previo al reconocimiento final de los caracteres, se va a realizar una binarización adaptativa de la imagen para facilitar la tarea al *OCR* y que éste obtenga mejores resultados.

Análisis de trazos rectos en hipótesis proyectadas. Como última comprobación se realiza un análisis de trazos rectos presentes en la imagen final para garantizar que el proceso de binarización no haya eliminado el contenido de las hipótesis que va a analizar el *OCR*.

Integración del proceso desarrollado y un OCR sobre el *iOS* Para completar la aplicación, se ha realizado la integración de un *OCR* de código libre disponible para integrarlo al proyecto implementado para el *iPhone*.

1.5. Estructura de la memoria

La presente memoria describe todo el trabajo realizado para concluir la finalización del Máster en ingeniería de sistemas e informática. En este segundo capítulo se recogen todas las mejoras introducidas sobre el sistema tomado como base. En el tercer capitulo, se realiza una evaluación de los resultados de este proceso y una comparación de éstos frente a otros *OCRs* y otras técnicas más reciente en las que se aplican técnicas de reconocimiento de objetos. En el último capítulo se recopilan las conclusiones obtenidas tras la realización de este trabajo.

Capítulo 2

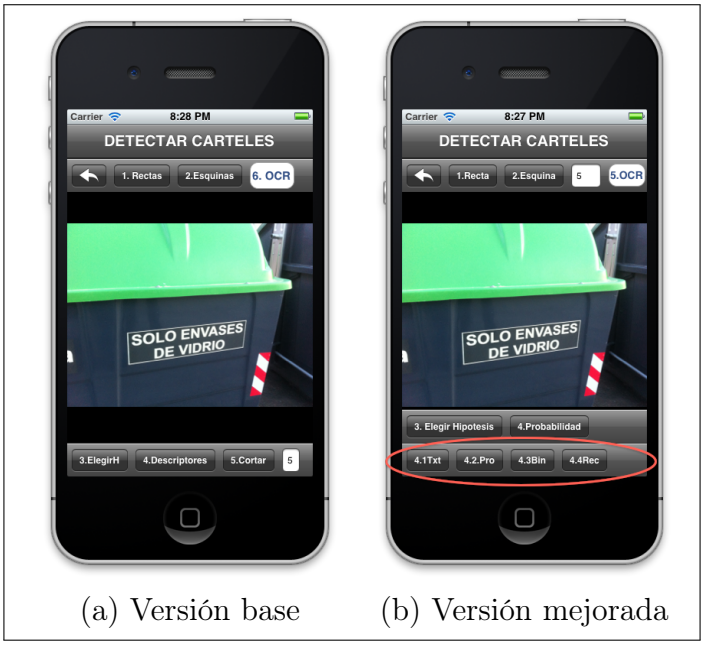
Sistema Robusto de Reconocimiento de Texto

En este capítulo se van a presentar todas las mejoras introducidas a la vez que se va a realizar una descripción más formal del proceso diseñado.

Figura 2.1: Ejemplo de fallo al procesar una hipótesis con el prototipo base. Aunque el cartel ha sido detectado correctamente (izquierda), los fallos en los filtrados posteriores hacen imposible el reconocimiento de su texto (derecha)

Antes de poder introducir ninguna mejora, se realizó una batería de pruebas, con una mayor variedad de imágenes que contenían carteles, con la intención de detectar fallos y descubrir en qué aspectos era posible incluir mejoras. En la Figura 2.1 se puede ver un ejemplo en el que se ha detectado perfectamente una hipótesis, pero sin embargo, su procesamiento no se realiza correctamente.

Para esta tarea, se partió de la interfaz alternativa desarrollada durante la construcción del prototipo base, que permitía mostrar los resultados intermedios obtenidos en cada paso crítico del proceso, como por ejemplo, ver qué hipótesis eran rechazadas por tener una baja probabilidad de contener texto. Aunque esta interfaz proporciona información útil, resulta escasa cuando se pretende analizar por qué en determinadas imágenes no se obtienen mejores resultados. En la Figura 2.2 se observan los detalles añadidos a esta interfaz de depuración sobre los pasos anteriores a proporcionar una hipótesis al *OCR*.



(a) Versión base (b) Versión mejorada

Figura 2.2: (a) Interfaz de depuración existente en el sistema base y (b) la nueva interfaz que permite visualizar más pasos intermedios.

2.1. Generación de hipótesis

Como ya se ha mencionado anteriormente, el proceso pretende detectar carteles existentes en una imagen para conseguir mejorar los resultados que proporciona un *OCR* convencional. El proceso diseñado está orientado a detectar posibles carteles, siendo éstos de forma rectangular. Al no existir ninguna restricción en la imagen a procesar, el primer paso es detectar segmentos rectos y puntos de corte (esquinas) entre todos ellos. Teniendo en cuenta la posición relativa del punto de corte entre dos rectas se puede etiquetar el tipo de esquina detectado como: superior izquierda, superior derecha, inferior izquierda o inferior derecha.

Para generar las hipótesis rectangulares se realiza un agrupamiento de esquinas compatibles. Para considerar que dos esquinas son compatibles se tienen en cuenta: los segmentos rectos que las forman, la distancia entre ellas y su tipo. Que dos esquinas sean compatibles

Tabla 2.1: Número de hipótesis generadas a partir del número de esquinas alineadas correctamente.

Esquinas	***Alineamientos***	***Hipótesis***
4	4	1
4	3	2
3	2	3
2	1	3
1	0	1

significa que pueden formar parte de un mismo rectángulo y por tanto, se considera que están alineadas. Es posible que no se consigan realizar alineamientos completos, es decir, con una esquina de cada tipo. En estos casos, es necesario estimar la forma rectangular de las hipótesis. En la Tabla 2.1 se muestran el número de hipótesis que se generan dependiendo del número de alineamientos conseguidos entre esquinas de distintos tipos.

Debido al elevado número de hipótesis que se pueden llegar a generar según la imagen y escenario utilizado, y con la intención de no sobrecargar el paso final del *OCR*, se limitó el número de hipótesis rectangulares aceptadas. La elección de qué hipótesis deben continuar el proceso depende del número de alineamientos conseguidos. De esta manera, se favorece a aquellas hipótesis que han conseguido realizar cuatro alineamientos de esquinas presentes en las imágenes frente a hipótesis formadas con un número inferior de esquinas encontradas.

A continuación, se detallan todas las mejoras introducidas.

2.1.1. Inserción de rectas

Aunque la extracción de rectas consigue resultados aceptables, en algunas imágenes no consigue extraer suficientes rectas (en la Figura 2.3 se muestran algunos ejemplos).

Para favorecer la detección de esquinas y que posteriormente permita construir hipótesis se ha insertado una nueva recta "virtual", correspondiente a un borde vertical de la imagen. La elección de incluir solo una recta en vez del borde completo de la imagen es para evitar generar una hipótesis que englobe toda o casi toda la imagen. Hay que destacar que esta recta sólo se añade en casos concretos, es decir, cuando existen varios segmentos rectos horizontales que finalizan muy próximos al borde de la imagen.

En la Figura 2.4 se muestra un ejemplo en el que se consigue generar una hipótesis más adecuadas gracias a este paso; en la versión original se detectaba una hipótesis formada por un alineamiento y en la versión final se consigue generar la misma hipótesis pero con cuatro alineamientos.

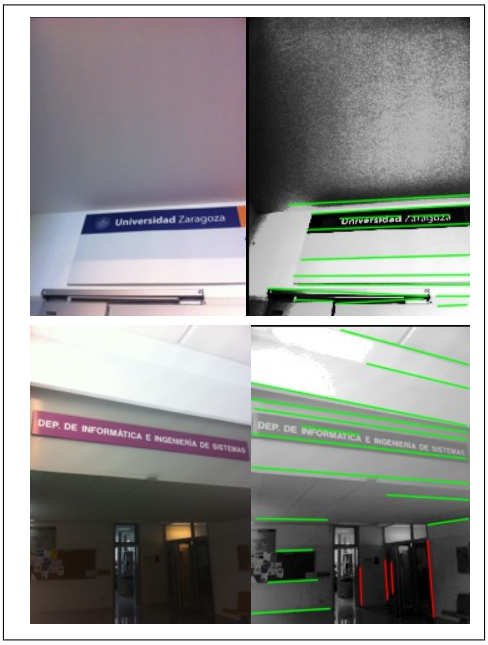
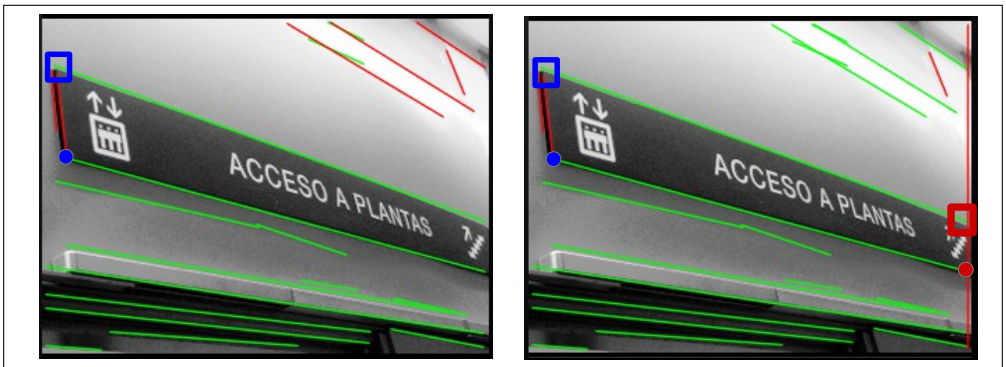


Figura 2.3: Ejemplos de imágenes donde la extracción de rectas no ha obtenido suficientes como para continuar con el proceso.



(a) 1 alineamiento (b) 4 alineamientos

Figura 2.4: En (a) se han detectado 2 esquinas, lo que provoca construir una hipótesis con 1 único alineamiento. Sin embargo en (b), gracias a la mejora conseguida al insertar una nueva recta "virtual", se han detectado 4 esquinas, lo que permite generar una hipótesis completa con 4 alineamientos.

2.1.2. Agrupamiento de esquinas

También se ha llevado a cabo un refinamiento en la agrupación de las esquinas. La existencia muy próxima de esquinas del mismo tipo provoca la generación de hipótesis

Figura 2.5: Agrupación de esquinas muy cercanas y del mismo tipo para disminuir el número de hipótesis solapadas.

muy similares. Lo que ocasiona un procesamiento repetitivo de la misma zona de la imagen sin mejorar los resultados. Es por ello, que se ha considerado necesario agrupar aquellas esquinas cercanas. Para elegir el representante de un grupo de esquinas, se ha atendido a su tipo. Por ejemplo, ante un grupo de esquinas superiores izquierdas, se selecciona como representante la esquina, y por tanto, se eliminan el resto de esquinas, aquella que se encuentra lo más próxima a la esquina superior izquierda de la imagen. De esta manera se pretende generar un menor número de hipótesis, pero que engloben el contenido de las hipótesis que se habrían generado con las esquinas que han sido eliminadas. En la Figura 2.5 se puede observar, en la primera imagen todas las esquinas encontradas y en la segunda, el resultado tras realizar el refinamiento mencionado anteriormente.

2.2. Evaluación de probabilidad de contener texto

Es muy posible que no todas las hipótesis generadas correspondan con carteles y, teniendo en cuenta que el paso donde el *OCR* realiza la extracción del texto es el paso más costoso (representa el 70 % del tiempo de procesamiento total), es preciso seleccionar cuáles de estas hipótesis tienen una mayor probabilidad de contener realmente texto. En el Anexo A puede consultarse con más detalle un análisis del tiempo de procesamiento

del proceso previo al *OCR* y del tiempo de procesamiento de éste.

Para analizar la probabilidad de contener texto se diseñaron una serie de filtros atendiendo a la forma geométrica de las hipótesis así como a analizar la textura de los píxeles dentro del rectángulo. Originalmente, estos valores fueron calculados basados en características observadas y confirmados con una serie de imágenes de prueba. Sin embargo, cuando se realizaron más pruebas con hipótesis reales, se observó que algunos de esos filtros no proporcionaban suficiente información como para determinar si las hipótesis contenían o no texto. Esos filtros correspondían con el valor de la *Kurtosis* de la distribución y con el criterio de que uno de los picos debía ser superior al 40 % del máximo valor encontrado. Con la eliminación de esos filtros se obtuvieron mejores resultados.

Los filtros, finalmente utilizados son los siguientes:

Tipo	expresión	peso
Geometría:	$(r \geq 0{,}5) \wedge (r \leq 10)$	5
Textura:	$(\bar{x} - M_o) \geq 1$	1
	$\sigma \geq 2$	1
	$\exists(Max_1(h) \wedge Max_2(h))$	2

donde $\bar{x}$, M_o y σ corresponden con la media, la moda y la desviación típica respectivamente, h el histograma calculado de la hipótesis y Max_1 and Max_2 corresponden con los 2 picos del histograma

Cada criterio incrementa el valor de los votos v_i de que una hipótesis R_i contenga texto. La probabilidad de que una hipótesis sea del tipo $Text$, se calcula así:

$$P_{ini} = P(R_i|Text) = \frac{v_i}{|V|} \tag{2.1}$$

donde $|V|$ es el número máximo de votos que una hipótesis podría obtener, estableciendo que cada criterio vota de acuerdo al peso mostrado anteriormente. Finalmente, sólo las hipótesis con una $P_{ini} > 0{,}75$ son aceptadas.

Se ha realizado un análisis de los resultados que proporciona este filtro con cerca de 500 hipótesis. Los resultados se recogen en la Tabla 2.2. En ella se puede ver que se consigue una tasa baja de Falsos Negativos (FN), es decir, este filtro no descarta hipótesis que realmente contienen texto.

2.3. Proyección de las hipótesis rectangulares

Las pruebas realizadas con el sistema base [17] confirmaron que un *OCR* proporciona mejores resultados en imágenes donde el texto está lo más frontal, centrado y recto. Para conseguir obtener una vista lo más frontal posible, se utilizó la técnica basada en la estimación de una homografía [18] correspondiente al plano del cartel para conseguir visualizar un cartel rectangular como si éste hubiera sido capturado visto de frente. Para

Tabla 2.2: Resultado de la evaluación de las hipótesis: % de las hipótesis correctamente clasificadas (VP) o no (VN), % de hipótesis clasificadas incorrectamente como que contenían texto (FP) o no (FN).

VP	***VN***	**FP**	***FN***
32,49 %	29,98 %	35,01 %	2,52 %

poder utilizar esta técnica es preciso estimar el tamaño real del cartel encontrado. La heurística utilizada en el prototipo base para estimar la longitud de la arista del rectángulo, *lado*, se calculó teniendo en cuenta el ángulo α:

$$lado = \begin{cases} lado & \text{si} \quad \alpha > 83^{o} \\ lado \cdot 1{,}1 & \text{si} \quad 63^{o} < \alpha \leq 83^{o} \\ lado \cdot 1{,}2 & \text{si} \quad 46^{o} < \alpha \leq 63^{o} \\ lado \cdot 1{,}3 & \text{si} \quad 23^{o} < \alpha \leq 46^{o} \\ lado \cdot 1{,}5 & \text{si} \quad \alpha \leq 23^{o} \end{cases} \tag{2.2}$$

donde *lado* es el tamaño en pixeles en la imagen actual de la arista horizontal y el ángulo α corresponde con el ángulo que forman las aristas horizontal y vertical del rectángulo.

Figura 2.6: Mejora en la nitidez de los caracteres proyectados

Antes de introducir una mejora en la forma de estimar el tamaño del rectángulo, se observó que se obtenían mejores resultados en la proyección de las hipótesis al aumentar el tamaño de la imagen. Originalmente, todas las imágenes utilizadas eran de 640 x 480 píxeles. Aprovechando que las imágenes capturadas por los dispositivos utilizados en el prototipo son de una mayor resolución, de hasta 3264 x 2448 píxeles, se ha optado por utilizar imágenes de 1024 x 768 píxeles. La decisión de utilizar esta resolución y no la máxima proporcionada radica en que la operación de proyectar las hipótesis es algo costosa y dado que el número de hipótesis puede llegar a ser elevado, utilizando esta resolución se han observado resultados aceptables. En la Figura 2.6 se puede observar como utilizando una imagen original de mayor resolución, se consigue aumentar en gran medida la

nitidez de las hipótesis proyectadas, lo que posteriormente permite un reconocimiento de caracteres mejor.

Estimación del tamaño real La segunda mejora introducida ha consistido en mejorar la estimación del tamaño real de las hipótesis. El cálculo del tamaño también se basa en el ángulo que forman las hipótesis rectangulares. Para estimar el tamaño se ha realizado un experimento con hipótesis reales para calcular la relación a partir de casos reales:

1. Se han capturado varias imágenes de un cartel, de tamaño conocido, desde distintos puntos de vista.
2. Se han extraído las hipótesis obtenidas y se ha calculado el ángulo que forman.
3. Al conocer el tamaño real del cartel, se conoce también su ratio de aspecto. Con este valor es posible calcular el factor por el que hay que multiplicar su lado para conservar el mismo aspecto de ratio.

En la Figura 2.7 se muestra una gráfica donde se relaciona el ángulo de cada hipótesis y el factor por el que hay que multiplicar el tamaño del lado para conseguir su tamaño real.

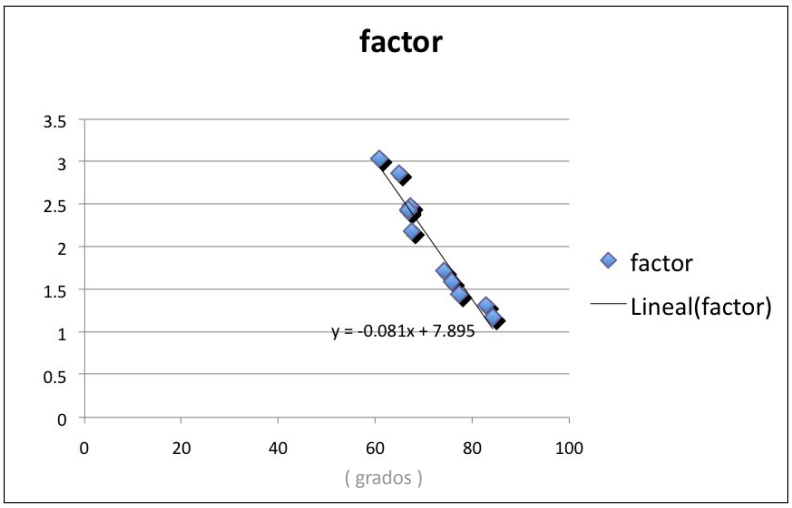


Figura 2.7: Función lineal para calcular factor a multiplicar por el lado

2.4. Selección de hipótesis finales

Llegados a este punto las hipótesis con mayor probabilidad podrían ser procesadas por el *OCR*. Sin embargo, al realizar una binarización previa se obtienen mejores resultados. Esta binarización es propia para cada hipótesis y está relacionada con los picos encontrados en su histograma (calculado durante el análisis de la probabilidad de contener texto).

Durante esta binarización, es posible que se elimine el texto como puede verse en la Figura 2.1. Para evitar pasar este tipo de hipótesis al *OCR*, en la versión original se analizaban los segmentos rectos existentes y en caso de existir más de 10, la hipótesis era válida y en caso contrario, se descartaba. Junto con este análisis también se comprobaba si la hipótesis contenía un posible borde y se procedía a su recorte, consiguiendo eliminar posible ruido.

Figura 2.8: Detección de rectas uniformemente distribuidas a lo largo de la imagen

La mejora introducida en este apartado realiza un análisis de estos segmentos rectos de una manera más completa. Las hipótesis que contienen caracteres correctamente binarizados poseen segmentos rectos verticales distribuidos uniformemente a lo largo de la imagen.Para detectar esta situación, se divide la hipótesis en celdas rectangulares, en concreto en 8 celdas, y se comprueba cuantas de estas celdas contienen segmentos rectos verticales y se calcula una nueva probabilidad, P_{lines}:

$$P_{lines} = \frac{C_i}{|C|} \tag{2.3}$$

donde C_i es el número de celdas que contienen al menos un segmento recto y $|C|$ es el número total de celdas. En la Figura 2.8 puede verse una hipótesis que no contiene texto y otra que sí y cómo se realiza la división en celdas para calcular esta nueva probabilidad. Finalmente, se re-calcula la probabilidad final de cada hipótesis como sigue:

$$P_{final} = P_{ini} * P_{lines} \tag{2.4}$$

El siguiente paso del proceso es extraer el texto de la hipótesis que posee la probabilidad máxima P_{final}, junto con aquellas que superan el 80 % de la máxima obtenida.

Capítulo 3

Evaluación y comparación respecto al estado del arte

En este capítulo se pretende realizar una evaluación del proceso diseñado para justificar que se han conseguido los objetivos planteados en este trabajo. Por una parte se va a evaluar cuánto se ha mejorado el sistema y por otra, se va a demostrar que el proceso diseñado no solo mejora los resultados del *OCR* integrado en el prototipo sino que, el proceso implementado consigue buenos resultados independientemente de él.

3.1. Evaluación de las mejoras

En el capítulo 2 se han presentado las distintas mejoras introducidas sobre el sistema base. Junto con cada incorporación se han mostrado casos en los que se observa que se obtienen mejores resultados parciales. Sin embargo, para medir la mejora global se han repetido las mismas pruebas que se hicieron originalmente con el sistema base. Esas pruebas consistieron en seleccionar imágenes, frontales y no frontales. Para cada imagen se contabilizan el número de caracteres que se han acertado para calcular una tasa de aciertos. Para cuantificar las mejoras realizadas se han repetido dichas pruebas, pero eligiendo únicamente imágenes no frontales, por ser éstas las más difíciles de procesar para el *OCR*, y por tanto, tienen un mayor interés para mostrar los resultados del trabajo realizado. Los resultados de estas pruebas quedan reflejados en la Tabla 3.1, donde se muestra la tasa de aciertos en cuanto a los caracteres correctamente reconocidos. La primera columna corresponde con los resultados obtenidos por el *OCR* original, la segunda

Tabla 3.1: Media μ y desviación típica σ del % de caracteres reconocidos correctamente por el *OCR* utilizado, el sistema base y el sistema final sobre 25 imágenes con carteles oblicuos.

	OCR	Base	Mejoras
	μ | σ	μ | σ	μ | σ
% char	21,44 % | 32,54 %	30,69 % | 39,41 %	47,82 % | 38,86 %

son los resultados del prototipo base y la ultima columna corresponde con la versión final, que contiene las mejoras descritas en este documento.

Con estos datos se puede cuantificar que se ha conseguido una mejora casi un 20 % sobre el sistema base y casi un 30 % sobre el *OCR* original. En el Anexo C se proporcionan los datos obtenidos para obtener las tasas de acierto mostradas en la Tabla 3.1.

De los resultados obtenidos, hay que resaltar, que al igual que existen casos en los que no se consigue detectar las hipótesis, también existen casos en los que aún detectando y binarizando correctamente la hipótesis del cartel, el *OCR* utilizado no consigue extraer el texto correctamente. Un ejemplo se muestra en la Figura 3.1.

Figura 3.1: Ejemplo de fallo al extraer el texto con el *OCR* utilizado

3.2. Comparación con otros *OCRs*

Durante el desarrollo del prototipo se analizaron distintos *OCRs* de código libre(Ocrad[1], Tesseract[2] y Gocr[3]) para su integración en el sistema. En la Tabla 3.2 se muestra un resumen de los aspectos a considerar en su elección. Aunque el *Tesseract* era el que más requisitos cumplía para ser integrado en el prototipo, no siempre proporciona unos buenos resultados como ya se ha visto en el ejemplo de la Figura 3.1. Por ello se ha realizado un nuevo análisis. Este análisis ha consistido en obtener las hipótesis con el sistema desarrollado pero se han utilizado otros *OCRs* para extraer su texto. Estos resultados se muestran

[1]http://www.gnu.org/software/ocrad/
[2]http://code.google.com/p/tesseract-ocr/
[3]http://linux.die.net/man/1/gocr

Tabla 3.2: Evaluación de 3 *OCRs* de código libre. Los criterios a tener en cuenta han sido: el % de caracteres reconocidos correctamente, si soportaban la posibilidad de varios lenguajes y su facilidad para la integración con el resto del sistema.

	Ocrad	Tesseract	Gocr
tasa de aciertos	84 %	77 %	57 %
soporte varios lenguajes		+	
fácil integración	+	++	+

Tabla 3.3: Tasa de acierto para distintos OCRs con y sin el sistema desarrollado (Pre).

	Pre + Tesseract		Tesseract	
	μ	σ	μ	σ
Frontales	74,89 %	32,73 %	36,76 %	38,98 %
Oblicuas	47,45 %	45,46 %	16,44 %	31,34 %
	Pre + GoogleDocs		GoogleDocs	
	μ	σ	μ	σ
Frontales	53,12 %	49,96 %	47,44 %	45,42 %
Oblicuas	30,00 %	48,30 %	4,21 %	13,31 %
	Pre + ABBYY		ABBYY	
	μ	σ	μ	σ
Frontales	76,53 %	34,42 %	77,72 %	41,12 %
Oblicuas	54,29 %	49,85 %	19,00 %	35,17 %

en la Tabla 3.2. Para estas pruebas se han seleccionado imágenes con carteles frontales y carteles que contenían el texto oblicuo. En ella se puede observar que el sistema desarrollado consigue obtener mejores resultados (columna izquierda) que los se obtiene utilizando solo los *OCRs* estudiados (columna de la derecha).

3.3. Nuevos enfoques

El reconocimiento óptico de caracteres se lleva utilizando desde hace mucho tiempo y como se ha podido observar a lo largo de este trabajo todavía presenta limitaciones a la hora de conseguir buenos resultados si no se respetan las restricciones en las imágenes a procesar. Desde hace unos años se están aplicando nuevos enfoques en el campo de la extracción de texto sobre imágenes generales. Aunque estas técnicas, complejas y costosas, quedan lejos de poder ejecutarse en un dispositivo móvil actual, no se pueden excluir de este trabajo, ya que por los resultados que presentan, éstos son mejores que los que ofrece un *OCR* tradicional.

La técnica que se estudia en este parte del trabajo es la propuesta en [15]. Sus autores, proponen utilizar técnicas de reconocimiento de objetos, considerando las palabras como objetos. Para el reconocimiento, el sistema propuesto consta además de un conjunto de palabras, *lexicon*, que serán las posibles palabras a detectar. Junto con los detalles del

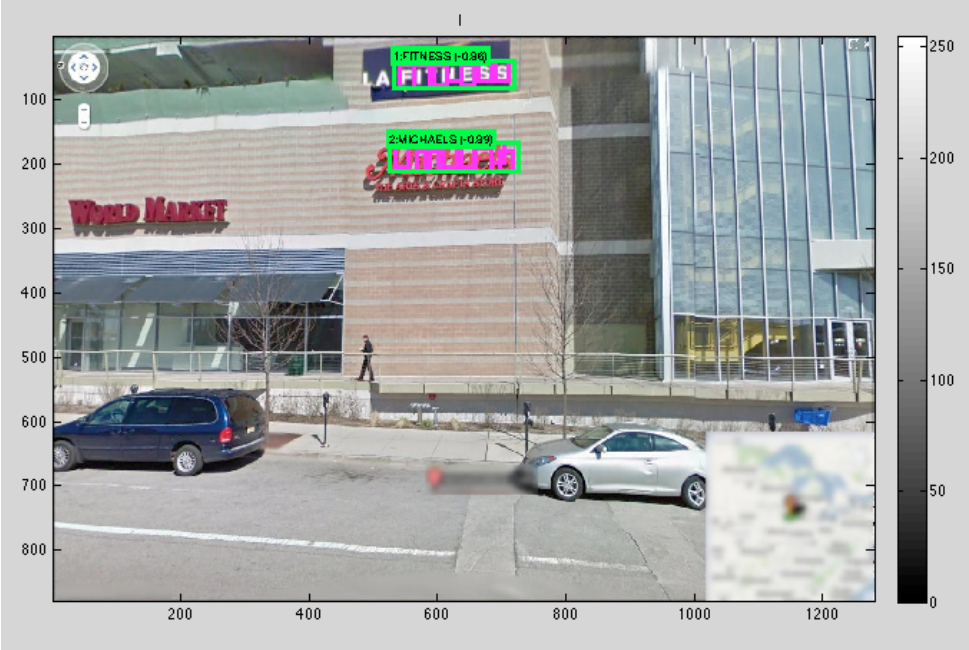


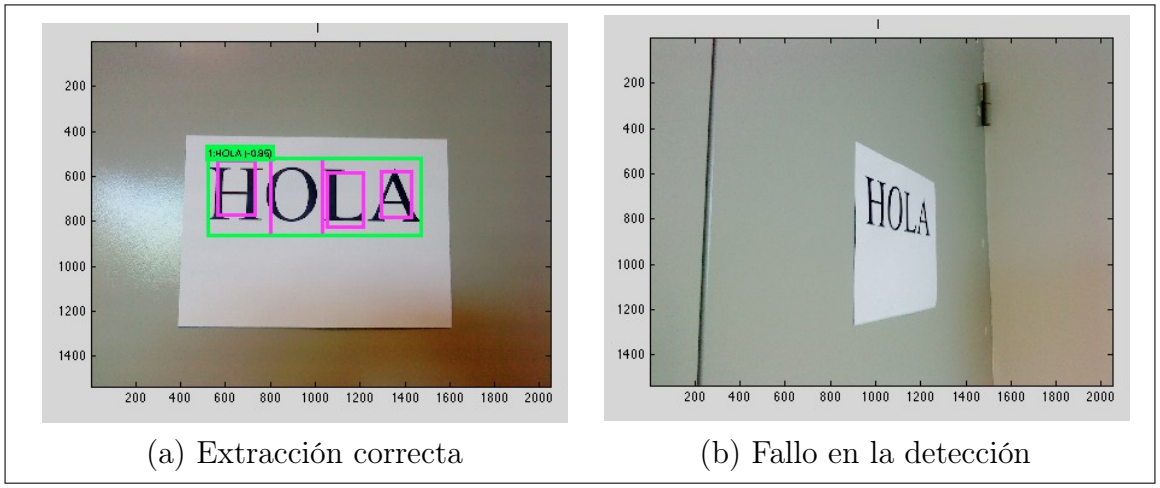
Figura 3.2: Resultado obtenido con el método propuesto en [15]

proceso y bases de datos de imágenes utilizadas; sus autores han dejado disponible[4] el código implementado en *Matlab*. En la Figura 3.2 puede verse un ejemplo del resultado de la ejecución de este método. El *lexicon* utilizado para esta ejecución concreta está formado por las siguientes palabras: *michaels*, *world*, *market* y *fitness*. En el resultado se observa que se han reconocido correctamente 2 palabras, para lo que ha necesitado alrededor de 23,28 segundos. Este sencillo resultado prueba que este método es más robusto que el desarrollado en este trabajo, ya que no existe la restricción de que los carteles tengan forma rectangular y por lo tanto, es mejor en escenas naturales. Sin embargo, en la referente a su ámbito de ejecución, al aplicar técnicas mucho más complejas y costosas, utilizar este sistema en un dispositivo móvil no se considera viable.

Uno de los puntos claves del trabajo realizado (capítulo 2) es que para conseguir mejores resultados en la extracción del texto con un *OCR* es rectificar la deformación provocada por la perspectiva. Para solucionar este aspecto se hace uso de las homografías, técnica aplicable puesto que el texto presente en las escenas suele estar casi siempre sobre un mismo plano (aunque las letras tengan relieve, éstas están apoyadas en el mismo plano). Con las pruebas que se realizaron se observó que las técnicas de reconocimiento de objetos aplicadas a reconocer palabras son más robustas a estas deformaciones. Sin embargo, lo son pero hasta cierto punto, tal y como se puede ver en la 3.3.

Visto que esta nueva técnica es más robusta en la extracción del texto en imágenes, pero falla cuando el texto está deformado debido a la perspectiva, es posible que combinando

[4]http://vision.ucsd.edu/ kai/grocr/



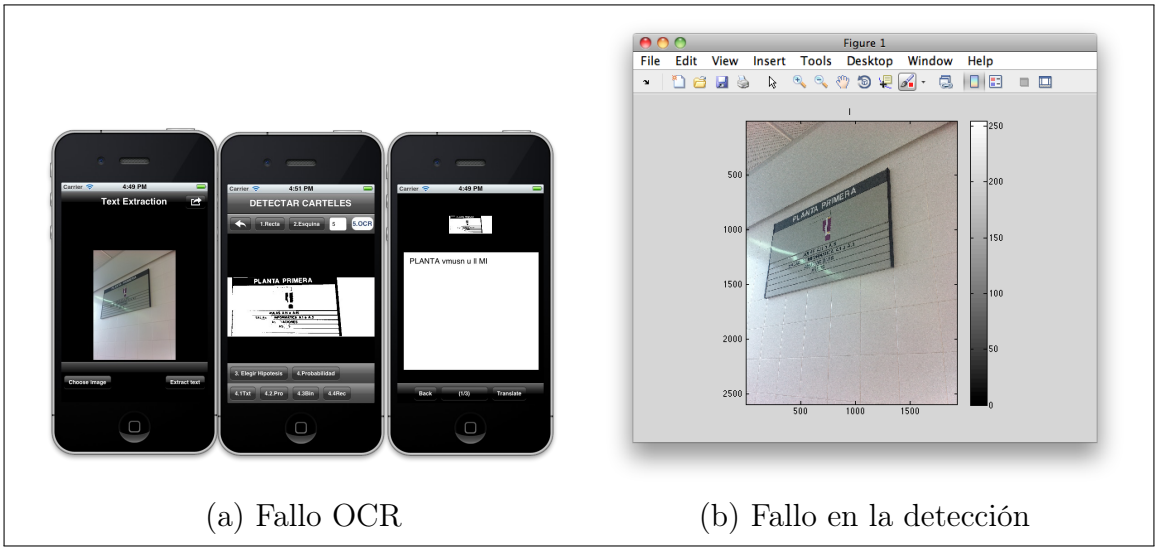
(a) Extracción correcta (b) Fallo en la detección

Figura 3.3: (a) Resultados obtenidos con [15] usando una imagen con el texto frontal y (b) donde el cartel está deformado por la perspectiva.

Tabla 3.4: Media μ y desviación típica σ del % de palabras reconocidas correctamente de las hipótesis generadas a partir del proceso desarrollado. Por un lado el texto ha sido extraído con el *OCR* integrado y por otro con [15].

	OCR μ | σ	Nuevo enfoque μ | σ
% palabras	26,67 % | 28 %	50,00 % | 29 %

los dos trabajos se obtengan mejores resultados. En la Figura 3.4 se ve un ejemplo en el que ninguno de los dos métodos consigue extraer el texto correctamente. Sin embargo, si combinamos ambos, como muestra la Figura 3.5, se puede ver que el texto se consigue extraer correctamente. En la Tabla 3.3 se puede ver el resultado del análisis realizado para comprobar la mejora que se conseguiría en la tasa de aciertos de palabras correctamente detectadas si se combinaran estos dos métodos. En el Anexo C se pueden consultar las tablas de las que se han extraído estos datos.



(a) Fallo OCR (b) Fallo en la detección

Figura 3.4: En (a) se puede ver un caso en el que el *OCR* no ha conseguido extraer todo el texto correctamente de la hipótesis. En (b), la misma imagen tampoco ha sido detectada con [15].

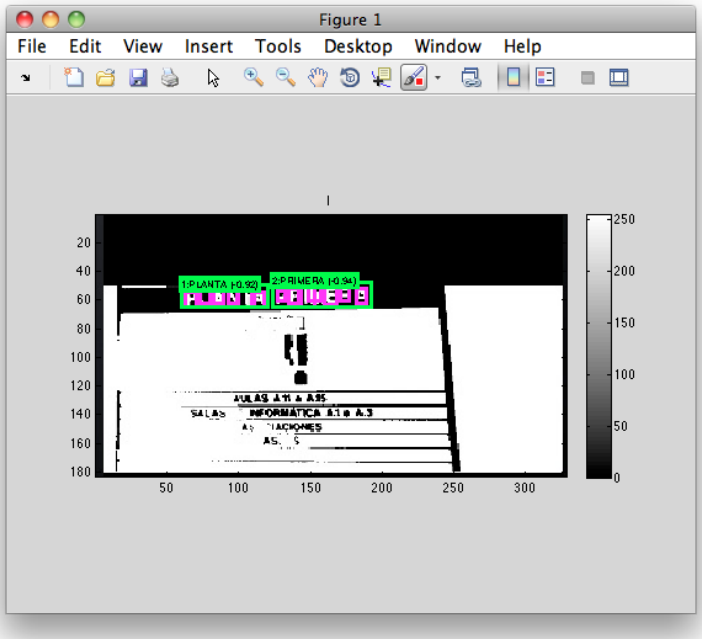


Figura 3.5: Si se combina el procesado propuesto en este trabajo con el sistema de reconocimiento de [15], se consigue reconocer el texto que ninguno de los dos ha conseguido detectar por separado (ver la Figura 3.4.

Capítulo 4
Conclusiones

Este capítulo contiene las conclusiones extraídas tras la realización de este trabajo así como una propuesta de líneas futuras.

4.1. Conclusiones

Este trabajo está dedicado a conseguir extraer texto de imágenes generales capturas desde la cámara de un dispositivo móvil. La primera parte de este trabajo se ha centrado en incorporar mejoras a un sistema base existente [17]. A la vez que se han desarrollado estas mejoras se ha llevado a cabo una definición formal de todo el proceso desarrollado. Con las mejoras realizadas se ha conseguido un sistema de generación de hipótesis más robusto. Al incorporar una recta "virtual" correspondiente al borde de la imagen se ha conseguido mejorar la detección de esquinas. Con el agrupamiento de las esquinas detectadas, se ha reducido el número de hipótesis solapadas, esencial para la eficiencia del proceso. La propuesta de mejora de la evaluación de las hipótesis que contienen texto consiguen reducir el número de hipótesis, reduciendo así el número de falsos positivos a analizar por el *OCR*. Este hecho es importante ya que la extracción del texto llevado a cabo por el *OCR* es el paso más costoso de todo el proceso. Todas estas mejoras han conseguido aumentar la tasa de aciertos en un 30 % sobre la tasa de aciertos del *OCR* utilizado. Para la evaluación del sistema también se han comparado con otros *OCRs*, ajenos al prototipo, donde también se han conseguido mejorar sus resultados.

La segunda parte del trabajo se ha centrado en evaluar si el proceso diseñado consigue mejorar también los resultados de técnicas más recientes en el campo de la extracción de texto. Como resultado principal de todo el trabajo realizado se escribió el siguiente articulo: *"Towards robust and efficient text sign reading from a mobile phone"* [1] publicado en el *2nd IEEE Workshop on Mobile Vision* llevado a cabo junto con el *ICCV 2011*, en Barcelona en Noviembre del 2011. El articulo completo puede consultarse en el anexo B.

[1] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6130223

4.2. Trabajo futuro

La principal línea de trabajo futuro sería integrar todo el proceso propuesto de pre-procesado y re-proyección de las imágenes con las nuevas líneas y enfoques basadas en el reconocimiento de objetos, ya que éstas no se centran en una forma geométrica particular. Tras las pruebas realizadas se ha visto que son técnicas mucho más generales para la extracción de texto en imágenes, pero que aún falta trabajo sobre su eficiencia para poder llegarse a ejecutar en un dispositivo móvil. Además, también se podría trabajar en su robustez frente a deformaciones provocadas por la perspectiva, lo cual parece posible mejorar con las técnicas propuestas en este trabajo.

Bibliografía

[1] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of ocr research and development. *Proceedings of the IEEE*, 80(7):1029 –1058, jul 1992.

[2] N. Ezaki, M. Bulacu, and L. Schomaker. Text detection from natural scene images: towards a system for visually impaired persons. In *Int. Conf. on Pattern Recognition*, volume 2, pages 683 – 686 Vol. 2, aug. 2004.

[3] C. Mancas-Thillou, S. Ferreira, J. Demeyer, C. Minetti, and B. Gosselin. A multifunctional reading assistant for the visually impaired. In *EURASIP Journal on Image and Video Processing*, 2007.

[4] M. Tanaka and H. Goto. Text-tracking wearable camera system for visually-impaired people. In *ICPR*, pages 1–4, 2008.

[5] YingLi Tian, Chucai Yi, and Aries Arditi. Improving computer vision-based indoor wayfinding for blind persons with context information. In *Proc. of Int. Conf. on Computers helping people with special needs*, pages 255–262, 2010.

[6] B. B Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and design of image recognition captchas. *In Proc. of Conference on Computer and Communications Security*, 2010.

[7] Victor Wu, R. Manmatha, and Edward M. Riseman. Finding text in images. In *Proc. of the Int. Conf. on Digital libraries*, DL '97, pages 3–12, New York, NY, USA, 1997. ACM.

[8] Paul Clark and Majid Mirmehdi. Combining statistical measures to find image text regions. In *Proc. of International Conference on Pattern Recognition*, pages 450–453, September 2000.

[9] Christian Wolf and Jean-Michel Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Anal. Appl.*, 6:309–326, February 2003.

[10] Huiping Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *Image Processing, IEEE Transactions on*, 9(1):147 –156, jan 2000.

[11] Xiangrong Chen and A.L. Yuille. Detecting and reading text in natural scenes. In *Computer Vision and Pattern Recognition*, pages II–366 – II–373 Vol.2, 2004.

[12] M. Cord J. Fabrizio and B. Marcotegui. Text extraction from street level images. *City Models, Roads and Traffic (CMRT)*, pages 199–204, 2009.

[13] S. Ferreira, V. Garin, and B. Gosselini. A text detection technique applied in the framework of a mobile camera-based application. *Proc. of Camera-based Document Analysis and Recognition, Seoul*, 2005.

[14] Qixiang Ye, Jianbin Jiao, Jun Huang, and Hua Yu. Text detection and restoration in natural scene images. *Journal of Visual Communication and Image Representation*, 18(6):504 – 513, 2007.

[15] Kai Wang and Serge Belongie. Word spotting in the wild. In *Proc. of the European Conference on Computer Vision*, pages 591–604, September 2010.

[16] Lukáš Neumann and Jiří Matas. A method for text localization and recognition in real-world images. In *Proc. of the 10th Asian Conf. on Computer Vision*, volume IV, pages 2067–2078, November 2010.

[17] Ana Belén Cambra Linés. Interpretación de carteles con la cámara de un móvil, Abril 2011. Proyecto Fin de Carrera.

[18] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

Anexos

Anexo A

Análisis de tiempo de procesamiento del OCR

En este anexo se recogen los datos de las pruebas realizadas para analizar el tiempo de ejecución del *OCR* utilizado.

Durante todo el presente documento se ha hablado de que es importante no sobrecargar al *OCR* con hipótesis que no contengan texto. En la Tabla se recogen los datos obtenidos de medir el tiempo medio de procesamiento de una hipótesis frente al tiempo que tarda el *OCR* en extraer su texto. En ella se aprecia que el tiempo de procesar una hipótesis por el *OCR* supone un 70 % del tiempo total de procesamiento de esa hipótesis. Esta es la razón por la que todo el trabajo desarrollado pretende que el *OCR* sólo analice aquellas hipótesis con mayor probabilidad de contener texto. De esta manera se consigue un tiempo de respuesta aceptable para el usuario final.

Tabla A.1: Tiempo medio, expresado en milisegundos, de procesamiento de una hipótesis frente a la extracción de su texto con el *OCR* utilizado.

Imagen	***Procesamiento***	***OCR***
flores	47,76	386,43
buzon	68,1812	428,08
tercero	142,8595	160,771
seminarios	66,2526	515,594
infra	55,547	42,883
lab	35,49	136,167
prob	53,12	209,76
lab micro	49,54	321,34
pulsador	50,63	95,97
acceso lejos	43,12	88,9
acceso cerca	51,48	226,97
salida cerca	62,14	96,81
vidrio	42,9	59,5
planta baja	41,48	29,711
plan baja sec	52,88	161,513
extintor	41,97	33,037
logronño	59,0105	87,85
uni	41,23	96,94
no usar	74,379	222,671
salida mad	42,32	50,374
diis	53,78	271,25
plan prim	47,88	101,708
plan prim peq	43,39	172,288
edificio	31,24	772,904
manguera	66,89	88,87
plan pri cen	55,79	414,07
aul+pul	47,14	177,46
uso bom frente	58,49	128,82
alquilo	24,19	79,74
pul+man	47,47	107,848
Tiempo Medio	**53,29**	**192,21**

Anexo B

Publicación de los resultados

A continuación se incluye el artículo *"Towards robust and efficient text sign reading from a mobile phone"* , donde se recogen parte de los resultados conseguidos con la realización de este trabajo. El articulo fue publicado en el *2nd IEEE Workshop on Mobile Vision* llevado a cabo junto con el *ICCV 2011*, en Barcelona en Noviembre del 2011.

Towards robust and efficient text sign reading from a mobile phone

A. B. Cambra, A. C. Murillo
DIIS, Instituto de Investigación en Ingeniería de Aragón
University of Zaragoza, Spain
ana.cambralines@gmail.com, acm@unizar.es

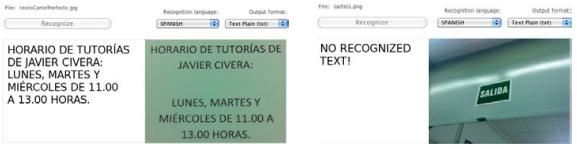
Abstract

Embedded applications on mobile phones are reaching impressive goals thanks to the current powerful smartphones. This work is focused on text recognition applications from mobile phone pictures. Optical Character Recognition (OCR) methods have been developed for a longtime, but they still have poor robustness to process text in general scene images. Our general goal is to study and improve their results, in particular when running locally on a phone. We present a realistic prototype running on iOS, with a light geometry based pre-processing step that helps detecting regions of interest in the image, i.e., likely to contain text-signs. Then, we show how to process and filter these hypothesis to facilitate text recognition by standard OCR methods. This initial version is aimed to rectangular shaped signs to easily take advantage of geometric cues. We demonstrate the performance improvements of including our proposal together with several available OCR libraries. All steps are run locally on the phone in the designed application, which can read or translate the text using additional standard services in the phone.

1. Introduction

There is a broad range of powerful *smartphones* on the market nowadays, and it is getting more and more common to use them for plenty of daily tasks besides standard calling and texting actions: maps, e-mail, agenda, games ... Their common characteristic is that they allow developing and installing new and third party applications that can make use of hardware components, such as GPS localization or integrated camera. As a general goal, this work is aimed to study how easy and feasible is to develop computer vision based applications in these mobile phones, within the particular field of text reading assistance applications.

There are plenty of Optical Character Recognition (OCR) systems, many available as open source libraries. However, they are not usually designed to process general scene images but scanned texts or images zoomed, cropped and centered in the text. We already find several recent commercial applications applying these techniques within mobile devices, making the phone able to "read" text from pictures but with some restrictions for the user. This work presents a proposal towards eliminating these restrictions and improving robustness of this kind of systems, what could increase their fields of application.

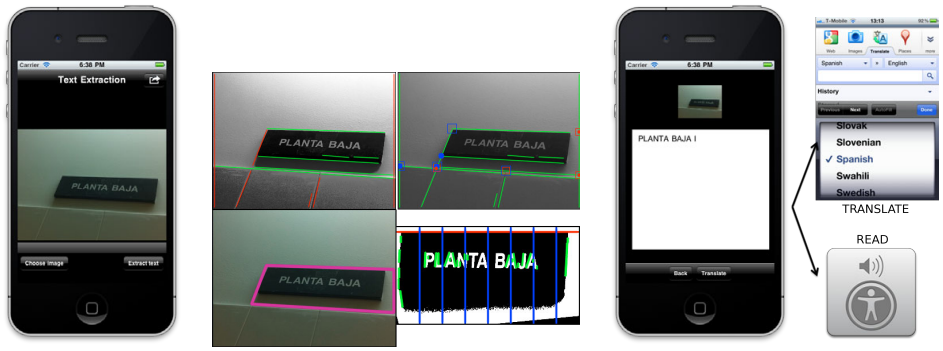


(a) Correct reading (b) Failed reading

Figure 1. (a) Successful recognition results obtained with an *OCR* in a frontal and zoomed view. However, more general scenes with distant text and perspective deformations (b) fail more often.

Typically, OCR methods present low recognition performance when the text in the image suffers perspective distortion or is not properly aligned, centered, zoomed or illuminated, as can be seen in the example in Fig. 1 [1]. Performance would increase if we could automatically obtain regions of interest containing text and process them to avoid those issues from general scene images. Considering the application should be run on a mobile phone brings important differences with regard to multiple related works, detailed next, on text recognition in unrestricted scenes. Simpler and efficient steps seem a requirement to allow the system to run locally on the phone. We start by restricting this initial work to rectangular text-signs, which allows us to make geometric assumptions that help obtaining an efficient prototype running on the mobile phone. Perspective deformation is corrected in plenty of computer vision applications through well known homography estimation and planar region projection, we present here how to apply these ideas in our particular problem. In order too run a homography based rectification, the basic step in this proposal is the gen-

[1] obtained with one of the many OCR available online: *OnlineOCR*, http://www.onlineocr.net/



(a) (b) (c) (d)

Figure 2. Summary of the proposed process. Acquisition (a), pre-processing steps to get a better view (b) that can be read by an standard OCR (c) and then used for different applications (d).

eration of rectangular region hypothesis in the image. This step is also aimed to automatically detect regions of interest, releasing the user from image acquisition restrictions (such as how the image should be acquired or where the text should be located in the image).

This work shows promising results running a simple approach and presents a fully automated text-sign reader prototype running on iOS. Besides presenting good usability and reasonable execution time, our proposal is shown to improve the performance of standard OCR libraries running on the phone. Several improvements and alternatives still have to be explored to generalize the approach to non-rectangular signs. Figure 2 shows a summary of the process developed. First the user acquires a new image (a), and then different steps based on geometric cues and simple texture analysis are run in a transparent way to the user (b). Finally we obtain the OCR analysis (c) only for the most likely hypothesis. This result is then said aloud or processed by an online dictionary (d). In the following sections, we describe and evaluate the details of our approach and the improvements in the performance when integrating it with different available OCR methods.

1.1. Related Work

There are plenty of successful OCR methods [10] proposed for a long time but we still find recent and novel contributions towards new fields of application, robustness or higher performance. Initially, OCR techniques were thought to process scanned texts from paper format to obtain a digital version of them. Nowadays, there are plenty of applications where it would be useful to recognize the text in general images. For example, in the particular case of reading assistance, we find plenty of proposals in the literature towards visually impaired assistance applications [3, 9, 12, 13]. However we still find few of these ideas integrated on mobile phones, partly due to high computational cost of some of the proposals. In this paper, we analyze which image processing techniques may be suitable for improving the results of this kind of applications running on a mobile phone. Thanks to the explosion of mobile application development we already find interesting computer vision based recent commercial products, such as wordlens[2] that reads texts in an augmented reality application.

As already mentioned, OCR methods typically require specific image acquisition (frontal view, zoom, centered ...) to have a good performance recognizing the text, presenting poor robustness to perspective distortions or more cluttered images, as seen in the initial example in Fig. 1. We can divide related research in two different groups. On one hand, there is a group of works towards better character recognition itself, proposing new OCR techniques. We even find works able to identify highly deformed text [18] as presented in web CAPTCHA systems. On the other hand, there are works that propose how to "pre-process" images of general scenarios to detect which image regions are likely to contain text and how to recognize it. Trying to recognize text in unrestricted images is an active field of research, with specific conferences and competitions dedicated to it. We find approaches proposing to work with image segments, dividing the image into connected components areas after applying different texture filters to the image [16]; or classification approaches at pixel-level, as proposed in [2], using histogram and gradient values on the region around each pixel. Certain works try to detect text along videos, taking advantage of the tracking and temporal information [15], [8]. Other interesting results towards detecting text areas in general scenarios include proposals based on different classification and learning techniques, e.g., using an AdaBoost based classifier [1] or training several SVM classifiers [6], to determine which regions are more likely to contain text. Some recent works propose to move from the use of typical OCR techniques to more general object recognition tech-

[2] http://questvisual.com/

niques adapted for text recognition [14, 11]. The work in [14] proposes, once a character recognition step has been run in the region of interest, to take into account the likelihood of full words for the final recognition results. Authors in [11] present a joint text localization and recognition approach that considers the characters as MSER regions and proposes to train the system with synthetic fonts.

Our work is related to the group of proposals trying to pre-process the images to recognize regions of interest, that can be later processed by standard OCR techniques. Two of the closest related works to our proposal can be find in [4, 17]. Both works use different orientation or perspective correction steps based on geometric constraings, but they do not integrate all steps required in the process nor demonstrate an application feasible in a phone.

2. Rectangular hypothesis

As described, this work is focused on reading text-signs and we can expect most of them being of rectangular shape. Therefore, we start the process with a search for possible rectangles in the image as regions of interest. Since there are no restrictions on the image to be processed, our approach detects straight segments and cross points (corners) between them all over the image. Out of these corners, possible rectangular hypothesis are generated and the likelihood of each of them containing text is evaluated.

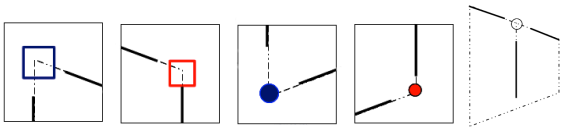
2.1. Straight segments and corner detection

We have implemented a C version[3] of the straight segment detection method proposed in [7]. The obtained segment set is filtered by size, we reject segments below 50 pixels, and by proximity/overlap, we merge segments whose end tips are too close or present big overlap. Then, if we divide the resulting segment set into vertical and others, we can search for cross points between each possible pair taking a segment from each group. To be sure the cross points selected are likely to be part of a rectangular object, we only accept cross points that fall into the image limits and whose distance in the image to the segment tips is below an stablished threshold. According to the relative position of the components of each cross point, we find four different types of corners, as shown in Fig. 3. Part (e) shows an example of particular cases where line segments cross each other in the middle part of one of them, therefore several corners are stored (one per possible type) at that location.

2.2. Rectangular hypothesis generation

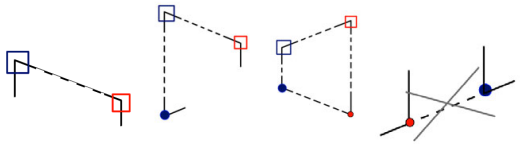
Next step seeks rectangular hypothesis built from the corners found in previous step. First, these corners are grouped with each other trying to get sets made of as many compatible corners as possible. We take into account first a

[3] Available at http://webdiis.unizar.es/~anacris/code/lineDetector.zip



(a) (b) (c) (d) (e)

Figure 3. Types of corners detected: (a) Top-left, (b) Top-right, (c) Bottom-left, (d) Bottom right. (e) Multiple-type corner: TL+TR.



(a) (b) (c) (d)

Figure 4. (a)-(c) Different sets of compatible corners with 2, 3 and 4 corners respectively, and (d) a pair of non compatible corners, due to wrong relative position.

suitable alignment of the line segments generating each corner point; secondly, a distance between the corners, which must be higher than minimum segment length; and finally a correct relative location of the different types of corners (Fig. 4 shows examples fitting or not this criteria).

Corners that have not been included in any compatible group are used to instantiate singleton hypothesis. Therefore we have compatible sets made out of one, two, three or four image corners, to allow the system to be robust against occlusions and line segments missed during the extraction process. All of these sets, except the four-corner ones, need a post-processing to estimate the complete shape of the rectangular region. For each of these matched sets, we estimate how the whole region could be. We take into account the segments that composed each corner to generate the missing sides of the rectangular regions. For same cases, as shown in Fig. 5, more than one hypothesis can be generated from a compatible set of corners.

2.3. Rectangular hypothesis evaluation

Besides taking into account the fact that some rectangular region sides and corners may not have been detected, we should deal with the fact that there can be many rectangular shapes in images that do not correspond to text signs. The final OCR step is the most expensive one (the whole processing time dedicated to one hypothesis is split aproximately in 30% of the time of the prefiltering and 70% for the OCR step), therefore we want to avoid overloading the system running it for all hypothesis. We evaluate the likelihood of each of them containing text to allow the system keep working only with a few top (most likely) candidates.

As basic hypothesis descriptor we compute the gray level histogram, with 16 levels. From prior knowledge we could expect to find two clear local maxima in this histogram, cor-

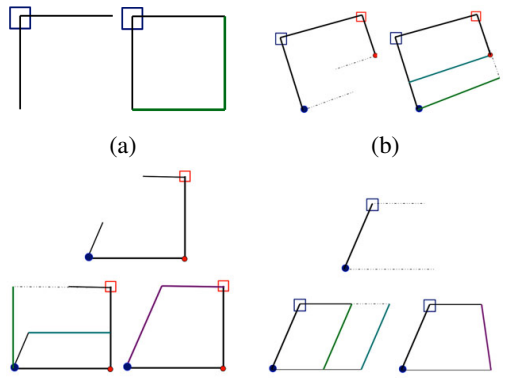


Figure 5. Rectangular hypothesis generated from single corners (a) and from compatible sets of four (b), three (c) and two (d) corners.

responding to the color of the text and background. Moreover, this double-maximum would make that the average and the mode are not too close to each other and that the distribution gets high standard deviation (compared to deviation on homogeneous textures hypothesis). These ideas where confirmed analyzing the values on a reduced set of rectangular hypothesis samples with or without text on them. We also checked on the kurtosis values, that point how disperse/concentrated a distribution is, but the sought range of values for this moment was not clearly segmented for text/non-text.

Then, the likelihood of each hypothesis containing text is obtained using the described histogram cues and a basic geometric cue (r as the ratio between length and width of the rectangular region) as follows:

• We evaluate these expressions for each hypothesis R_i:

Type	expression	weight
Geometry:	$(r \geq 0.5) \wedge (r \leq 10)$	5
Texture:	$(\bar{x} - M_o) \geq 1$	1
	$\sigma \geq 2$	1
	$\exists(Max_1(h) \wedge Max_2(h))$	2

being $\bar{x}$, M_o and σ the average, mode and standard deviation respectively, and h the histogram of the evaluated hypothesis with two local maxima Max_1 and Max_2.

• Each matched criteria increases the number of votes v_i for rectangular hypothesis R_i being a text area. Each criteria votes according to the weight shown in previous step. This way we can control which criteria is more important, for instance depending on how certain we are that it should be always true. These votes are used to estimate a simple likelihood of the hypothesis being of type $Text$, as follows:

$$P_{ini} = P(R_i|Text) = \frac{v_i}{|V|} \quad (1)$$

being $|V|$ the maximum amount of votes that can be obtained.

• Hypothesis with $P_{ini} > 0.75$ are accepted to continue with the rest of the process (except at certain cases with extremely large amount of hypothesis, where only a fixed number of most likely hypothesis are accepted to avoid application overload).

3. Final processing of accepted hypothesis

Once we have selected a set of likely rectangular hypothesis through the geometry cues and the simple likelihood evaluation process, we proceed with a more detailed analysis of the accepted hypothesis, to decide the hypothesis that would be given to the OCR step.

3.1. Frontal projection through homographies

OCR methods are shown to perform better when text is presented in a frontal, zoomed and centered view. Therefore, the key idea in this step is to project the current image to the equivalent that would be seen if the sign would have been seen frontally and centered.

"Virtual" frontal view. We suggest a simple way to estimate how a "virtual" frontal view of the sign will look like, as shown in Fig. 6. The text we are looking for must be contained on a plane in the 3D scene (the sign board) with rectangular shape. Let us suppose ABCD are the corners of the rectangular hypothesis we have instantiated. If it would correspond to an actual rectangular sign, its frontal view should contain parallel and perpendicular edges. Depending on the orientation of sides and angle between hypothesis edges, the process decides which sides to keep as reference and which weight should be applied to the other dimensions. For instance, in the example in Fig. 6, we keep vertical edges as reference because they are closer to their ideal slope. Then, the longest vertical side, $\bar{AC}$ is kept as it is and the horizontal sides are "projected" to be perpendicular to $\bar{AC}$ with an increase on its length inversely proportional to angle α. This angle is always estimated between 0 and 90^o, and we establish a maximum increase of 60% on the side length. Thanks to this guess about how the corners should be projected, we can estimate an homography to automatically project the original view into the virtual one as detailed next.

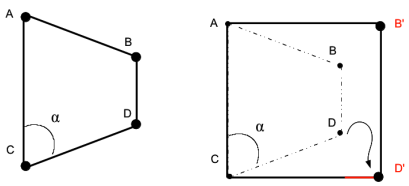


Figure 6. Current view of the rectangular region (left) and goal "virtual" frontal view (right).

Homography estimation An homography, **H**, is a well known projective geometry transform [5], that relates two different planar region projections. We estimate it between the "virtual" image and the original one to obtain a mapping from any pixel **x** that belongs to the sign plane in the original image to the corresponding point **x'** in the other view:

$$\mathbf{x}' = \mathbf{H}\,\mathbf{x}. \tag{2}$$

In general, four corresponding points between the two views of the planar surface are enough to estimate the homography between both views. Therefore, we estimate where the four corners of current hypothesis rectangular region (A, B, C, D) may appear in the virtual frontal view (A', B', C', D'), and estimate the homography with these correspondences. Now we can project the whole hypothesis region according to the estimated **H**. Figure 7 shows two projection examples with images acquired from the mobile phone. We should note that even if the shape we are guessing for virtual view is not accurate, the goal of getting the text in a frontal and homogeneous view (similar size for all characters) is nicely achieved by this simple idea.

Figure 7. Two examples of the projection from an original image (left) to a virtual frontal view (right).

3.2. Re-evaluation of hypothesis likelihood

Finally, we have observed that even though some OCR methods include a binarization step, we can include a more flexible one thanks to the previous steps run in our proposal. Then, a re-evaluation of the likelihood of being text after the projection and binarization steps is run, to try to include the new information that may have been generated there.

Image adaptive binarization. Since we have already cropped the image rectangular region and computed the gray level histogram over the hypothesis region, we analyze its mode to determine a flexible binarization threshold that depends on it.

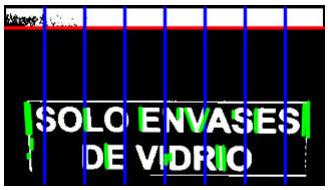


Figure 8. Final evaluation of segments in the hypothesis area. Long segments still found inside the region (top segment, in red) are used to crop it. Distribution of small segments (in green) inside the 8 region cells (marked with long vertical edges) is used to update the hypothesis likelihood of containing text.

Straigh segments analysis. The final likelihood evaluation of containing text for the remaining hypothesis depends on the distribution of straight segments on the hypothesis region. We extract straight segments on the binarized image, accepting segments of five pixel length or more. Two interesting goals are achieved in this part of the process. First, if long segments (that occupy most of the length or width of the rectangular area) are found, they are used to further crop the view. This way we clean a little more the view that will be provided to the OCR. Secondly, we divide the rectangular region in several cells as shown in Fig. 8 and we re-evaluate the likelihood of the hypothesis being text, P_{final}. We combine the information regarding straight segments distribution, P_{lines}, in the hypothesis with the initial estimated likelihood (1) as shown in (3). Then, we select the most likely hypothesis, maximum P_{final}, together with hypothesis with a likelihood within 80% of that maximum.

$$P_{final} = P_{ini} * P_{lines} \tag{3}$$

P_{lines} is computed from the count of cells containing small segments: $P_{lines} = \frac{C_i}{|C|}$, where C_i is the number of cells that contain at least one straight segment and $|C|$ is the number of cells (8 in our experiments). Note that this re-evaluation is conservative, only decreasing the final probabilities of hypothesis being text, but it does not directly reject any, unless there are no straight segments detected. If this happens, we are quite certain that there are no sharp characters or that the binarization step went wrong and all the region in the hypothesis became black or white, losing the details. Besides, in order to present a prototype running on the mobile phone, we need to keep acceptable execution times. As mentioned before the OCR step is the most expensive one and this filter helps processing with it as few hypothesis as possible. The experiments show some examples of the benefits of using this final filter.

3.3. Integration with the OCR step

The hypothesis accepted at this point will finally be processed by an OCR to recognize their text. We have evaluated three of the most used OCR methods available as open

source to be able to integrate them in our prototype: Ocrad[4], Tesseract[5] and Gocr[6]. To choose which one was more suitable for our prototype, we performed some tests to evaluate their results with several images with text signs on them. Results and analysis of these tests are summarized in Table 1, where each column corresponds to a different OCR library. First row shows the average percentage of characters appearing in an image that are correctly identified by the approach; Second and third row summarize two additional criteria, besides performance, of interest for building the prototype: possibility of using different trained model depending on the language to be used and ease to integrate with the iOS prototype, respectively. According to this brief analysis, we decided to use the best compromise option, tesseract, to implement the final prototype described and tested in the following section.

Table 1. Evaluation of three open source OCR libraries.

	Ocrad	Tesseract	Gocr
correct recognition	84 %	77%	57 %
different languages		+	
easy integration	+	++	+

4. Prototype developed and evaluation

We have designed and implemented an iOS application to demonstrate our proposal in the mobile phone. The prototype have been tested with around 200 different images, with different resolutions, acquired from a mobile phone. Our test images are divided in two types, frontal and oblique views, that are usually easier and more difficult to recognize respectively for an standard OCR. Some of the test images are shown in Fig. 12. Our proposal evaluation has been twofold: on one hand, checking the usability of the application; on the other hand, evaluating the performance and improvements observed with the different steps proposed.

4.1. Prototype design and implementation on iOS

The application designed runs three basic stages. First, the user has to choose an image from the *iPhone Photo library* or acquire a new image with the device camera. Second, the proposed processing is applied to the chosen image. This step is transparent for the user, who must just wait until it is finished. Once the text is recognized, besides showing it on the screen, we have implemented two useful possible final applications: one option sends the text to an online translator; the other one just reads the resulting text using the *voiceover* option at iOS. An example of the application running at the different explained stages is shown in Fig. 9. More details on the different steps running on different test cases can be found in the video provided as additional material to this paper (recorded while running the process in the iOS simulator platform).

(a) (b) (c)

Figure 9. Screenshots of an execution of the proposal. Image acquired (a), result after automatic processing (b) and final use of the answer, translation from spanish to english in this case (c).

4.2. Evaluation of hypothesis likelihood estimation

First, we present a brief analysis of the proposed hypothesis likelihood evaluation for containing text. We checked the classification of every generated hypothesis (447) in a few images, with a likelihood acceptance threshold of 75%. A summary of these results is presented in Table 2. True Negatives (TN) in this case are hypothesis that did not contain a text sign and were properly rejected, while True Positives (TP) are hypothesis accepted that did actually contain a text-sign. First column of results shows performance before projecting the hypothesis to frontal virtual views. At that point, critical issue is to avoid false negatives, and we can appreciate that very few of the rejected hypothesis did contain text. Final results shown in the second column correspond to the final results of our proposed process, just before running the OCR step. We can appreciate how the projection and simple line segment analysis proposed in section 3 further improve the results. Accuracy is higher, i.e., we achieve better classification of the hypothesis. Now more hypothesis without text are rejected so the accuracy of the text class recognition increases to 0.90 from 0.63. This last re-evaluation step, subsection 3.2, significantly reduced the amount of hypothesis accepted to be processed in the last OCR step (the percentage of processed hypothesis was decreased from 60% to 18 %). Besides, we observed that none of the hypothesis rejected after this step would be

Table 2. Hypothesis classification accuracy.

		Before projection	After projection and line filtering
No text	$\frac{TN}{TN+FN}$	0.92	
Text	$\frac{TP}{TP+FP}$	0.63	0.90

[4]http://www.gnu.org/software/ocrad/ocrad.html
[5]http://code.google.com/p/tesseract-ocr/
[6]http://jocr.sourceforge.net/

properly read with the OCR step. This re-evaluation just helped to re-rank likely hypothesis. Figure 10 shows an example where the initial texture analysis incorrectly gave a high P_{ini} of being text, but the low amount of segments found decreased its final likelihood to the half.

Figure 10. Re-evaluation step provides better likelihood estimation in cases where simple histogram analysis fails. This example got high P_{ini} but low P_{lines}, properly decreasing the final likelihood of being text because there are very few straight segments (green).

4.3. Evaluation of final recognition performance

In order to demonstrate that the proposed process helps in the aimed applications, we carried out a first detailed evaluation using 10 different images of each type. These tests consisted of calculating the percentage of characters appearing in the image which were correctly recognized using different open and commercial OCR or running the whole process proposed in our prototype before the OCR. Besides the chosen open-source library to integrate in our prototype (tesseract) we also evaluate the improvements obtained when using the OCR available in Google Docs web application and the commercial OCR ABBYY (available for trial version). To run these experiments, we saved the processed hypothesis images that were provided to the OCR step during the execution of our proposal, to provide exactly the same ones to the other OCR methods. Table 4.3 shows the average μ and standard deviation σ in the percentage of image characters correctly recognized for different configurations. First two columns show results with our proposal run together with the OCR and last two columns were obtained just running the OCR. Different rows present results using easier tests (frontal views) compared to harder cases (oblique views). We can see the pre-processing proposed significantly improves character recognition results for most cases. As expected, it especially helps with oblique views thanks to the re-projection to virtual frontal views.

We analyzed in more detail the results using the OCR integrated with our prototype. This second set of tests, consisted of 50 images of each type, frontal and oblique views. We count how many "correct readings" we obtain, considering a good reading when the text in the sign has been recognized clearly enough for a person to understand the meaning. Although this is a subjective criteria, we typically observed in our tests that over 80% of characters recognized can provide a good idea of the meaning. Besides, in most of that cases the "grammar correction" features from text editors are able to suggest the correct word. Figure 11 summarizes the results of this experiment, with clear improvements when incorporating our preprocessing steps.

Table 3. OCR performance with and without our proposal (Pre).

	Pre + Tesseract		Tesseract	
	μ	σ	μ	σ
Frontal	74,89 %	32,73%	36,76 %	38,98 %
Oblique	47,45 %	45,46%	16,44 %	31,34 %
	Pre + GoogleDocs		GoogleDocs	
	μ	σ	μ	σ
Frontal	53,12 %	49,96%	47,44 %	45,42 %
Oblique	30,00 %	48,30%	4,21 %	13,31 %
	Pre + ABBYY		ABBYY	
	μ	σ	μ	σ
Frontal	76,53 %	34,42%	77,72 %	41,12 %
Oblique	54,29 %	49,85%	19,00 %	35,17 %

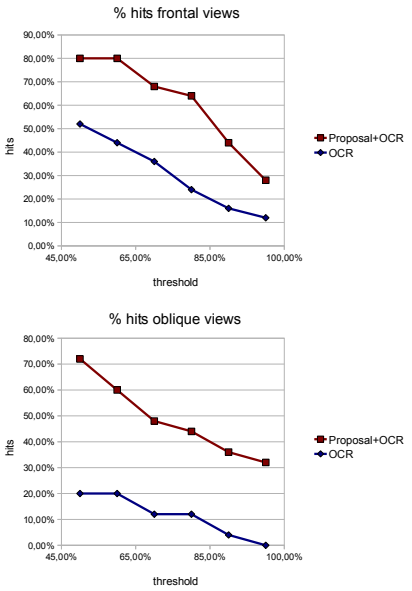
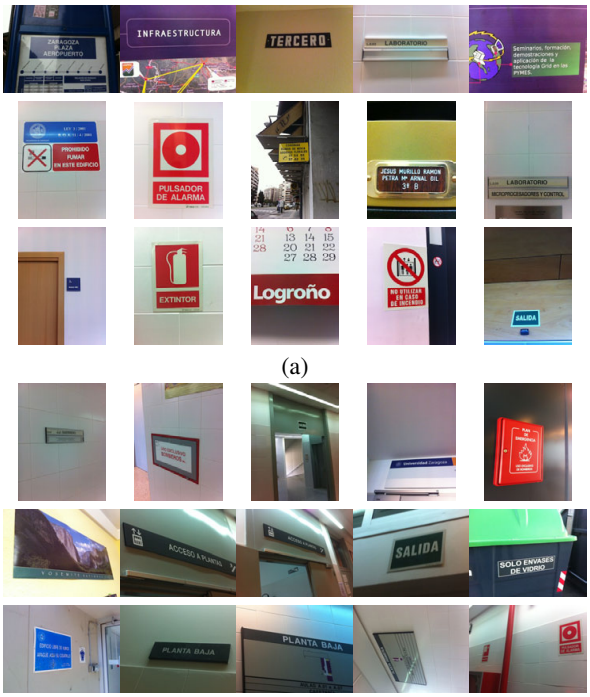


Figure 11. Percentage of tests considered as correct readings (hits), using different acceptance threshold on the % of recognized characters for frontal and oblique views.

5. Conclusions and Future Work

We have presented in detail an approach to read text-signs in images acquired from a mobile phone in general scenes. The whole process is described, evaluated and demonstrated in a functional prototype working on iOS. Many image processing and classification techniques are too costly to run on the mobile phone, so we try to kept this in mind to design our prototype steps. The key ingredients of our proposal are a simple and efficient analysis of the image based on geometric cues. This allows that two additional basic texture analysis steps are enough to filter the



(a)

(b)

Figure 12. Some of the test images used to evaluate the proposal, both from frontal (a) and oblique (b) views.

best regions of interest. Currently our approach is aimed to rectangular text signs, to facilitate some geometric assumptions needed for the homography based image rectification step. We study how to present the selected regions of interest in an optimal way to be processed by an standard OCR and demonstrate the improvements of including our pre-processing before running several standard OCR techniques. Future work improvements can be done towards a more general and more efficient approach. For instance, the approach could be extended to other geometric shapes, finding alternative ways of estimating the homography for rectification. Regarding efficiency, some steps could be optimized by for example detecting area overlap between hypothesis to save some repeated computations.

Acknowledgments

This work has been supported by projects DPI2009-14664-C02-0 and DPI2009-08126.

References

[1] X. Chen and A. Yuille. Detecting and reading text in natural scenes. In *Computer Vision and Pattern Recognition*, pages II–366 – II–373 Vol.2, 2004. 2

[2] P. Clark and M. Mirmehdi. Combining statistical measures to find image text regions. In *Proc. of International Conference on Pattern Recognition*, pages 450–453, September 2000. 2

[3] N. Ezaki, M. Bulacu, and L. Schomaker. Text detection from natural scene images: towards a system for visually impaired persons. In *Int. Conf. on Pattern Recognition*, volume 2, pages 683 – 686 Vol. 2, aug. 2004. 2

[4] S. Ferreira, V. Garin, and B. Gosselini. A text detection technique applied in the framework of a mobile camera-based application. *Proc. of Camera-based Document Analysis and Recognition, Seoul*, 2005. 3

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 5

[6] M. C. J. Fabrizio and B. Marcotegui. Text extraction from street level images. *City Models, Roads and Traffic (CMRT)*, pages 199–204, 2009. 2

[7] J. Košecká and W. Zhang. Video compass. *ECCV '02 Proceedings of the 7th European Conference on Computer Vision*, pages 476–490, 2002. 3

[8] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *Image Processing, IEEE Transactions on*, 9(1):147 –156, jan 2000. 2

[9] C. Mancas-Thillou, S. Ferreira, J. Demeyer, C. Minetti, and B. Gosselin. A multifunctional reading assistant for the visually impaired. In *EURASIP Journal on Image and Video Processing*, 2007. 2

[10] S. Mori, C. Suen, and K. Yamamoto. Historical review of ocr research and development. *Proceedings of the IEEE*, 80(7):1029 –1058, jul 1992. 2

[11] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Proc. of the 10th Asian Conf. on Computer Vision*, volume IV, pages 2067–2078, November 2010. 3

[12] M. Tanaka and H. Goto. Text-tracking wearable camera system for visually-impaired people. In *ICPR*, pages 1–4, 2008. 2

[13] Y. Tian, C. Yi, and A. Arditi. Improving computer vision-based indoor wayfinding for blind persons with context information. In *Proc. of Int. Conf. on Computers helping people with special needs*, pages 255–262, 2010. 2

[14] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. of the European Conference on Computer Vision*, pages 591–604, September 2010. 3

[15] C. Wolf and J.-M. Jolion. Extraction and recognition of artificial text in multimedia documents. *Pattern Anal. Appl.*, 6:309–326, February 2003. 2

[16] V. Wu, R. Manmatha, and E. M. Riseman. Finding text in images. In *Proc. of the Int. Conf. on Digital libraries*, DL '97, pages 3–12, New York, NY, USA, 1997. ACM. 2

[17] Q. Ye, J. Jiao, J. Huang, and H. Yu. Text detection and restoration in natural scene images. *Journal of Visual Communication and Image Representation*, 18(6):504 – 513, 2007. 3

[18] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and design of image recognition captchas. *In Proc. of Conference on Computer and Communications Security*, 2010. 2

Anexo C

Detalles de las pruebas realizadas

En este anexo se muestran las pruebas que contienen los resultados las pruebas realizadas para 25 imágenes. Para estas pruebas se han seleccionado imágenes que contienen carteles oblicuos por considerarse estos los más difíciles de procesar.

En la Figura C.1 se muestran los resultados obtenidos de procesar las imágenes de prueba con el *OCR* utilizado. En la Figura C.2 se muestran los resultados obtenidos utilizando el sistema base y en la Figura C.3 se observan los resultados con las mejoras introducidas.

En la Figura C.4 se muestran los resultados obtenidos de procesar las hipótesis obtenidas de distintas imágenes con carteles oblicuos, por un lado utilizando el *OCR* integrado en el prototipo, y por otro lado, combinando el proceso de pre-procesamiento y re-proyección con nuevos enfoques en el campo de la extracción de texto que aplican técnicas de reconocimiento de objetos.

C.1. Imágenes utilizadas

En la Figura C.5 se muestran las imágenes que han sido utilizadas para las pruebas realizadas y calcular la tasa de aciertos de cada sistema.

OCR	YOSEMITE NATIONAL PARK		ACCESO LEJOS		ACCESSO CERCA		SALIDA CERCA		VIDRIO	
CARACTERES		%		%		%		%		%
TOTAL	20		14		14		6		19	
aciertos	0	0,0%	4	28,6%	0	0,0%	0	0,0%	8	42,1%

OCR	LABORATORIO		Planta baja pequeña		Planta baja seccion		universidad zaragoza		PLAN DE EMERGENCIA USO EXCLUSIVO BOMBEROS	
CARACTERES		%		%		%		%		%
TOTAL	11		10		10		19		38	
aciertos	0	0,0%	0	0,0%	0	0,0%	0	0,0%	25	65,8%

OCR	PLANTA PRIMERA grande		JJ GUERRERO		SALIDA ESQUINA		USO EXCLUSIVO BOMBEROS		PLANTA PRIMERA pequeña	
CARACTERES		%		%		%		%		%
TOTAL	13		10		6		20		13	
aciertos	0	0,0%	0	0,0%	0	0,0%	19	95,0%	0	0,0%

OCR	PROHIBIDO FUMAR EN ESTE CENTRO		ENTRADA		PULSADOR ALARMA AULA A.15		MANGUERA		PULSADOR MANGUERA	
CARACTERES		%		%		%		%		%
TOTAL	26		7		22		8		24	
aciertos	18	69,2%	6	85,7%	8	36,4%	0	0,0%	0	0,0%

OCR	SALIDA seccion		cursos español		SALIDA PUERTA		EDIFICIO LIBRE DE HUMOS APAGUE AQUI SU CIGARRILLO		MANGUERA	
CARACTERES		%		%		%		%	8	%
TOTAL	6		65		6		42		6	
aciertos	0	0,0%	52	80,0%	0	0,0%	0	0,0%	2	33,3%

Figura C.1: Resultados del OCR original.

SISTEMA BASE	YOSEMITE NATIONAL PARK		ACCESO LEJOS		ACCESSO CERCA		SALIDA CERCA		VIDRIO	
CARACTERES		%		%		%		%		%
TOTAL	20		14		14		6		19	
aciertos	0	0,0%	6	42,9%	6	42,9%	0	0,0%	19	100,0%

SISTEMA BASE	LABORATORIO		Planta baja pequeña		Planta baja seccion		universidad zaragoza		PLAN DE EMERGENCIA USO EXCLUSIVO BOMBEROS	
CARACTERES		%		%		%		%		%
TOTAL	11		10		10		19		38	
aciertos	0	0,0%	0	0,0%	0	0,0%	0	0,0%	0	0,0%

SISTEMA BASE	PLANTA PRIMERA grande		JJ GUERRERO		SALIDA ESQUINA		USO EXCLUSIVO BOMBEROS		PLANTA PRIMERA pequeña	
CARACTERES		%		%		%		%		%
TOTAL	13		10		6		20		13	
aciertos	11	84,6%	0	0,0%	0	0,0%	20	100,0%	13	100,0%

SISTEMA BASE	PROHIBIDO FUMAR EN ESTE CENTRO		ENTRADA		PULSADOR ALARMA AULA A.15		MANGUERA		PULSADOR MANGUERA	
CARACTERES		%		%		%		%		%
TOTAL	26		7		22		8		24	
aciertos	13	50,0%	0	0,0%	13	59,1%	0	0,0%	0	0,0%

SISTEMA BASE	SALIDA seccion		cursos español		SALIDA PUERTA		EDIFICIO LIBRE DE HUMOS APAGUE AQUI SU CIGARRILLO		MANGUERA	
CARACTERES		%		%		%		%		%
TOTAL	6		65		6		42		8	
aciertos	0	0,0%	16	24,6%	0	0,0%	37	88,1%	6	75,0%

Figura C.2: Resultados del sistema base

SISTEMA MEJORADO	YOSEMITE NATIONAL PARK		ACCESO LEJOS		ACCESSO CERCA		SALIDA CERCA		VIDRIO	
CARACTERES		%		%		%		%		%
TOTAL	20		14		14		6		19	
aciertos	0	0,0%	8	57,1%	7	50,0%	5	83,3%	19	100,0%

SISTEMA MEJORADO	LABORATORIO		Planta baja pequeña		Planta baja seccion		universidad zaragoza		PLAN DE EMERGENCIA USO EXCLUSIVO BOMBEROS	
CARACTERES		%		%		%		%		%
TOTAL	11		10		10		19		38	
aciertos	0	0,0%	10	100,0%	10	100,0%	0	0,0%	20	52,6%

SISTEMA MEJORADO	PLANTA PRIMERA grande		JJ GUERRERO		SALIDA ESQUINA		USO EXCLUSIVO BOMBEROS		PLANTA PRIMERA pequeña	
CARACTERES		%		%		%		%		%
TOTAL	13		10		6		20		13	
aciertos	0	0,0%	0	0,0%	3	50,0%	20	100,0%	6	46,2%

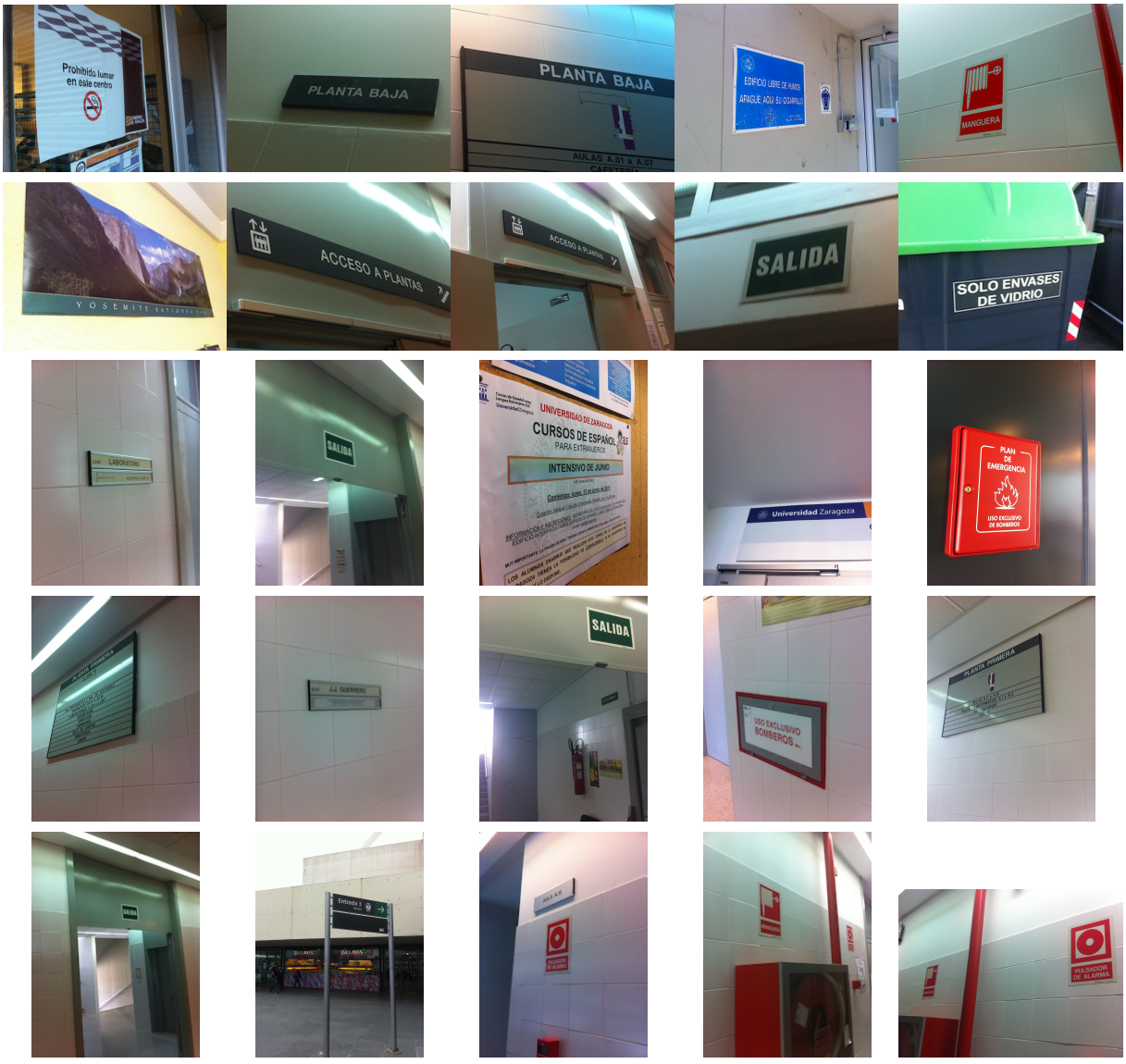
SISTEMA MEJORADO	PROHIBIDO FUMAR EN ESTE CENTRO		ENTRADA		PULSADOR ALARMA AULA A.15		MANGUERA		PULSADOR MANGUERA	
CARACTERES		%		%		%		%		%
TOTAL	26		7		22		8		24	
aciertos	18	69,2%	2	28,6%	13	59,1%	0	0,0%	16	66,7%

SISTEMA MEJORADO	SALIDA seccion		cursos español		SALIDA PUERTA		EDIFICIO LIBRE DE HUMOS APAGUE AQUI SU CIGARRILLO		MANGUERA	
CARACTERES		%		%		%		%		%
TOTAL	6		65		6		42		8	
aciertos	0	0,0%	29	44,6%	0	0,0%	37	88,1%	8	100,0%

Figura C.3: Resultados del sistema con las mejoras realizadas

SISTEMA MEJORADO	PLAN DE EMERGENCIA USO EXCLUSIVO BOMBEROS		PLANTA PRIMERA pequeña		ACCESO LEJOS		ACCESSO CERCA		PLANTA PRIMERA grande	
PALABRAS		%		%		%		%		%
TOTAL	6		2		3		3		2	
aciertos	2	33,3%	0	0,0%	2	66,7%	1	33,3%	0	0,0%
SISTEMA COMBINADO										
PALABRAS		%		%		%		%		%
TOTAL	6		2		3		3		2	
aciertos	2	33,3%	2	100,0%	1	33,3%	1	33,3%	1	50,0%

Figura C.4: Resultado obtenidos para calcular las mejoras que se podrían conseguir combinado el trabajo desarrollado en este documento y con nuevos enfoques de reconocimiento de texto.



(b)

Figura C.5: Imágenes utilizadas para calcular la tasa de aciertos