ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"IMPLEMENTACIÓN DE HERRAMIENTA PARA LA EXTRACCIÓN DE INFORMACIÓN DE DOCUMENTOS DE IMPORTACIÓN UTILIZANDO TECNOLOGÍA OCR"

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES

ESPECIALIZACIÓN MULTIMEDIA

RAMIREZ MERA CARLOS ALFREDO

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTOS

Mi más sincero agradecimiento a Dios por haberme dado la fuerza y sabiduría para superar cada obstáculo que se me ha presentado en mi vida, por haberme dado una familia con personas maravillosas mis padres, mis hermanos, mi esposa y recientemente la bendición más grande que uno puede tener en la vida, mi hijo Carlos Daniel.

Quiero agradecer de manera especial a mis padres, sin los cuales no hubiese podido llegar a donde estoy, gracias a su apoyo incondicional y a los valores que me han inculcado a lo largo de mi vida me han permitido progresar a lo largo de mis estudios y me han motivado a alcanzar mis metas sin importar lo difícil que sea llegar a ellas.

También a quiero agradecer a mi esposa quien en los 4 últimos años que ha pasado a mi lado me ha ayudado a ser mejor persona y a perseverar a pesar que muchas veces quise rendirme, por estar siempre ahí a pesar de que tantas veces quise estar solo y abandonar mis responsabilidades, siempre supo cómo encaminarme y gracias a ella pude seguir adelante en mis estudios, gracias por su confianza y apoyo en los momentos más difíciles.

DEDICATORIA

El presente proyecto lo dedico a mi hijo Carlos Daniel, quien nació hace tres meses y fue el impulso más grande que tuve para poder terminar mi trabajo de grado. Aunque muchas veces vi imposible realizarlo por la dificultad presentada, encontré la fuerza y motivación en él, mi alegría más grande que cambio mi manera de ver la vida. Él me dio el empuje que necesitaba en los últimos días para poder alcanzar mi meta de ser un profesional, un padre del cual se sienta orgulloso y de esta manera poderle brindar un mejor futuro, todo esfuerzo así sea el más grande del mundo, vale la pena si es por él.

TRIBUNAL DE EVALUACIÓN

Daniel Ochoa Donoso	**Daniel Ochoa Donoso**
PROFESOR DE MATERIA INTEGRADORA	TUTOR ACADÉMICO

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Carlos Ramírez Mera

RESUMEN

El presente trabajo de titulación consiste en la implementación de una herramienta que permita extraer información de documentos de importación utilizando tecnología OCR. El objetivo es desarrollar un programa que permita automatizar el proceso de transcripción de texto de un documento de importación escaneado a un formulario web de manera más efectiva y en menor tiempo.

Actualmente este trabajo es realizado por empleados quienes en ciertas ocasiones cometen errores; esto perjudica en el tiempo del proceso y económicamente a la empresa, que debe cancelar un valor para solicitar el reenvío del formulario corregido. Los documentos escaneados muchas veces son capturados de manera incorrecta con una mala calidad en la imagen obtenida, lo cual influye en la interpretación de la información del formulario.

Por esta razón se desarrolló una aplicación permite corregir los problemas relacionados a la calidad de la imagen, además de reconocer y extraer los campos de los documentos de importación, a través de un archivo de texto estructurado que pueda ser utilizado a futuro por cualquier otra plataforma digital.

Fue implementado con tecnología de reconocimiento óptico de caracteres del motor de reconocimiento de Google Tesseract y procesamiento de imágenes mediante algoritmos creados en EMGU CV. Puede ser ejecutado en Windows para lo cual necesita tener instalado el código ejecutable junto a los archivos de estructura del documento a procesar.

Como resultado se tuvo un 90% de aciertos en el texto interpretado en un tiempo promedio de 7 segundos a través de la herramienta desarrollada. Estos resultados fueron generados a partir de una muestra de 10 documentos de importación con texto nítido y legible ubicado dentro de las posiciones definidas en la estructura previamente por el usuario. El porcentaje de efectividad depende de la calidad del documento escaneado y es posible mejorar a futuro mediante implementación de algoritmos de corrección de caracteres.

Palabras Clave: Tecnología OCR, Procesamiento de imágenes, Reconocimiento de texto, Transcripción de texto

ABSTRACT

The present graduation work consists of the implementation of a tool that allows extracting information from import documents using OCR technology.

The objective is to develop a program that automates the process of transcribing text from a scanned import document to a web form more effectively and in less time.

Currently this work is done by employees who sometimes make mistakes. This damages the time of the process and economically to the company, which must cancel a value to request the resubmission of the corrected form. Scanned documents are often captured incorrectly with poor image quality, which influences the interpretation of form information.

For this reason, an application was developed to correct problems related to the quality of the image, as well as to recognize and extract the fields of the import documents, through a structured text file that can be used in the future by any other digital platform.

It was implemented with optical character recognition technology from the Google Tesseract recognition engine and image processing using algorithms created in EMGU CV. It can be executed in Windows for which you need to have the executable code installed next to the structure files of the document to be processed.

As a result, there was a 90% success in the interpreted text in an average time of 7 seconds through the tool developed. These results were generated from a sample of 10 import documents with clear and legible text located within the positions defined in the structure previously by the user. The percentage of effectiveness depends on the quality of the scanned document and it is possible to improve it in the future by implementing character correction algorithms.

Keywords: OCR technology, image processing, text recognition, text transcription

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

En la actualidad existe una gran cantidad de herramientas tecnológicas que han facilitado la realización de tareas de manera más eficiente generando mejores resultados en múltiples campos, tales como: educación, salud, manufactura, comunicación, etc. No obstante, aún existen personas que realizan tareas de manera manual, ya sea por: falta de recursos económicos para adquirir una herramienta tecnológica de ayuda, falta de conocimiento, o el poco grado adaptabilidad a la tecnología que pueden tener los individuos.

Los seres humanos son capaces de realizar múltiples tareas al mismo tiempo, no obstante ejecutar diferentes actividades durante un periodo largo de tiempo, puede conllevar a cometer errores, debido a que el cerebro va perdiendo concentración en las tareas que requieren mayor atención, generando una pérdida de eficiencia en el trabajo realizado.

En las empresas una de las tareas comúnmente realizadas, es la transcripción de documentos “copiar en otra parte algo que ya está escrito”; es decir copiar un texto manuscrito o impreso a un medio electrónico sean formularios, mail, documentos electrónicos, entre otros.

El presente proyecto propone implementar un Sistema utilizando tecnología OCR (Reconocimiento Óptico de Caracteres), que permita automatizar el proceso de transcripción de campos del documento escaneado a un formulario web.

Los Sistemas OCR (Reconocimiento Óptico de caracteres) son herramientas dirigidas a la digitalización de texto, permiten detectar caracteres en imágenes o documentos electrónicos para posteriormente generar datos que podrán ser editados en herramientas de procesamiento de texto, son empleados a nivel empresarial y educacional, otorgando beneficios como el incremento de la competitividad, disminución de costos y mayor calidad en los servicios que ofrecen. En la digitalización de documentos el uso de los Sistemas OCR representa un ahorro importante de tiempo, porque la información de los documentos es reconocida de forma automática en vez de ser ingresados de forma manual.

Entre las principales ventajas del uso de sistemas OCR se encuentran: La reutilización del texto contenido en la imagen, el ahorro de recursos de almacenamiento al transformar una imagen escaneada a un documento de texto, el ahorro de recursos humanos e incremento de la productividad de las personas las cuales podrán enfocarse en otras actividades para mejorar de calidad de servicio.

1.1 Definición del problema

La mayoría de personas son propensas a cometer errores al realizar una o más actividades en su día a día. De igual manera en su jornada laboral los individuos no se encuentran exentos a tener fallos en las actividades realizadas en su trabajo, más aún, si un exceso de carga laboral conlleva a realizar varias tareas con prisa al mismo tiempo.

La problemática de este proyecto se centra en los errores cometidos por el personal de una empresa al transcribir la información de documentos escaneados a un formulario web de la entidad que controla las importaciones en el país. Si el formulario es enviado con errores a la entidad aduanera es devuelto para que se realice su respectiva corrección, una vez corregido se debe realizar una solicitud para volver enviar el formulario, esto implica cancelar un valor por el reenvío de la información corregida. De esta manera afecta el tiempo estimado para realizar esta operación y perjudica económicamente de la empresa, cada vez que se cometen errores involuntarios al llenar los formularios web de manera manual.

Por medio de la tecnología OCR se propone implementar un sistema que permita realizar la transcripción de información de un documento escaneado.

Los Sistemas OCR son reconocidos por su fácil uso y gran utilidad interpretando el texto en imágenes, no obstante pueden tener problemas, como indican varios estudios [1], [2], [3] al reconocer texto debido a sus limitaciones al detectar los caracteres, entre las cuales están:

- La tipografía del texto, la cual se puede convertir en un inconveniente al no poder identificar tipografías poco comunes. Además de esto si el

documento contiene texto manuscrito dificulta el reconocimiento de las letras a interpretar.

- La Distorsión en la forma de los caracteres, la presencia de trazos no propios de la forma original de las letras puede conllevar a una mala interpretación o a no detectar el carácter original.
- Sobre posición de texto, es una distorsión común en la impresión de formularios, puede encontrarse texto sobreimpreso en otro texto. Esto conlleva a pérdida de la forma original de los caracteres y por consecuente no permite el reconocimiento del texto al Sistema OCR.
- La calidad de la imagen es otro de los puntos que afectan en el rendimiento de un intérprete OCR por lo cual, el texto impreso en el documente debe ser bastante claro y la imagen debe tener una buena resolución. Es recomendable trabajar con imágenes de mínimo 300 dpi, esto depende directamente del dispositivo de captura, la mayoría de escáneres en la actualidad pueden generar imágenes con esta resolución.

Se ha realizado diferentes pruebas con los sistemas OCR y se ha encontrado que son capaces de generar resultados con porcentajes de 90-95% de acierto dependiendo de los factores mencionados anteriormente relacionados a la calidad de la imagen.

1.2 Causas

1.2.1 Error humano

El error humano está relacionado a la mala práctica de las personas al realizar una actividad en la que participa directa o indirectamente, generando accidentes o incidentes en el campo laboral.

Existen varias causas por los cuales las personas pueden equivocarse al momento de realizar actividades en el trabajo divididas en dos grupos, análisis similares en [4], [5], [6].

Factores personales: Son causas que se generan a partir de las características de las personas, entre las cuales tenemos:

- Falta de conocimiento.
- Falta de concentración.

- Negligencia.

Factores laborales: Son elementos relacionados al ambiente laboral y el estado de las herramientas tecnológicas utilizadas por los empleados, entre las cuales tenemos:

- Presión laboral.
- Tecnología o equipos inadecuados para realizar un trabajo.

1.2.2 Calidad de imagen

En la actualidad existen sistemas OCR como Adobe Acrobat Pro, Omnipage, ABBY Fine Reader y otras herramientas OCRs Online que permiten detectar y reconocer texto en imágenes, con interpretaciones bastante aproximadas al texto original. Los resultados generados por estas herramientas dependen de la calidad de la imagen y el pre procesamiento correctivo aplicado a los archivos analizados.

La calidad de la imagen es determinada por una buena resolución, una correcta orientación y la nitidez en la impresión del documento.

La calidad del texto impreso puede afectar al reconocimiento de los caracteres, al tener una mala calidad de impresión los mismos no se encuentran completamente cerrados lo cual origina una mala interpretación al momento de analizar el texto de la imagen, casos similares en [7], [8].

Otros factores relacionados a la calidad del texto son manchas, texto sobreimpreso, marcas de agua y presencia de trazos como líneas, puntos o texto manuscrito cercano a los caracteres, este tipo de problemas afectan la morfología de los mismos. Lo cual al ser analizado genera pérdida de información y divergencia en la interpretación del texto.

Los sistemas OCR aplican los correctivos por medio de pre procesamiento de imágenes antes de ejecutar el algoritmo de reconocimiento de texto.

1.2.3 Comparación de áreas de texto de estructura.

Parte del sistema OCR es identificar las áreas de texto con información de interés a extraer del documento. Generalmente los documentos

escaneados pueden contener formas como líneas, imágenes o texto que no deseamos extraer del documento. Por lo cual antes de aplicar el algoritmo de reconocimiento es necesario segmentar la imagen en áreas de texto para poder filtrar el contenido no deseado. Las herramientas de reconocimiento de texto suelen realizar un análisis del contenido del documento, en aplicaciones similares [9], [10], [11]; por esta razón es necesario definir previamente la estructura que represente la ubicación de los campos a extraer de la imagen. Después de obtener las áreas de texto definidas en la estructura se puede presentar errores en los algoritmos de detección, por lo cual es importante tener en cuenta los siguientes escenarios:

Donde AL =Área layout, AD =Área detectada.

- UN AD coincide correctamente con AL.
- Un AL está dividida, más de una AD coincide con AL.
- Una AD contiene más de una AL
- Una AL está parcialmente completa, parte del AL no se encuentra en AD.
- Un AL no fue detectado.
- Un AD no está definido en ningún AL.

Los escenarios en los cuales no coincide un AD con su respectiva AL pueden generar un análisis incorrecto al momento de interpretar el texto del área extraída.

1.3 Efectos

Los diferentes problemas encontrados al implementar un Sistema OCR pueden traer consecuencias futuras a partir de su uso, proceso de adaptación y de cómo puede cambiar e influir en el modelo de negocio.

Con la implementación de un sistema se espera obtener mejores resultados en los procesos los cuales va automatizar. Las ventajas obtenidas deben ser mayor a los inconvenientes que pueda causar, por lo cual es importante minimizar cualquier tipo de inconveniente relacionado a los procesos ya establecidos y los cambios que podría generar la incorporación de un nuevo sistema. De esta manera se puede obtener un mayor beneficio de la herramienta OCR.

En el caso de los OCR puede que no afecten de manera crítica a los procesos de una empresa, pero requieren de mayor precauciones como se expuso anteriormente al momento de obtener la imagen escaneada, en la definición de la estructura, en la calidad de la impresión y orientación al momento de escanearla son limitantes que pueden ser mitigadas al momento de escanear el documento.

Los problemas de coincidencia de áreas de texto detectadas con la estructura definida puede generar una mala interpretación lingüística o extracción de un campo esperado con información de otro campo, por lo cual el sistema debe ser lo suficientemente robusto para detectar y corregir este tipo de errores, y obtener resultados conforme a lo definido en la estructura del documento.

El sistema OCR implementado pretende cubrir deficiencias que otros sistemas pueden tener, al orientar la detección de texto, definiendo una estructura que permitirá dirigir la búsqueda a áreas específicas de la imagen, lo que permite aislar elementos que no son de interés de reconocimiento. Con esto se mejora el reconocimiento y la eficiencia del algoritmo OCR

1.4 Objetivos

1.4.1 Objetivos Generales

- Desarrollar un programa utilizando tecnología OCR, que permita realizar el trabajo de transcripción de texto de un documento escaneado a una plataforma digital.
- Desarrollar un programa que pueda integrarse a otro tipo de interfaz sea móvil, web o de escritorio.
- Desarrollar algoritmos que permitan corregir la calidad del documento escaneado.

1.4.2 Objetivos Específicos

- Extraer la información de la imagen en una estructura de texto editable por el usuario.
- Corregir y mejorar la calidad del texto de documentos por medio de procesamiento de imágenes.

- Evaluar los resultados obtenidos midiendo el porcentaje de aciertos y el tiempo de ejecución del programa desarrollado.

1.5 Soluciones Similares

Actualmente existen diferentes aplicaciones OCR que permite cumplir el objetivo de este tipo de softwares que es extraer un texto de una imagen bastante aproximado al deseado, comparaciones realizadas en [12], [5].

Entre los sistemas OCR más populares actualmente tenemos los siguientes:

- **Adobe Acrobat Pro**

  El lector de documentos PDF de Adobe es una de las herramientas más conocidas y cuenta con su propio sistema OCR, pero solo está disponible para la versión pagada Premium que permite guardar documentos pdf como un documento Word. Una de las desventajas de este sistema es contener saltos de línea segmentación de palabras si el archivo PDF no tiene la suficiente calidad para convertirlo.

- **Omnipage**

  Es un programa desarrollado por Nuance (Multinacional desarrolladora de software), es bastante popular y funciona muy bien para archivos en lote, cuenta con un asistente que automatiza el proceso de OCR para múltiples archivos al mismo tiempo. Puede tener problemas de compatibilidad con ciertos tipos de archivos. El costo de la versión estándar es de 117 dólares.

- **ABBYY Fine Reader**

  Reconocido como uno de los mejores OCR para Windows, cuenta con una gran cantidad de herramientas que empodera y facilita el uso del OCR, permite recrear un maquetado del documento y puede trabajar con imágenes de calidad media, debido a que cuenta con algoritmos de procesamiento de imágenes. Los resultados generados son bastante buenos, el costo de la versión estándar es de 169 dólares.

- **OCRs online**

  Existen varias herramientas online que permiten transformar imágenes a texto, pero la desventaja de estas apps webs es que no son muy confiables y los resultados pueden tener error.

1.6 Solución Propuesta

Se propone implementar un programa que permita detectar y extraer información de documentos escaneados. El sistema debe ser robusto y confiable, incluye un módulo de detección y corrección de errores en la imagen, ejemplos similares en [13], [14].

Se debe analizar los documentos de importación con la finalidad de identificar problemas relacionados a la calidad del documento. De esta manera el programa desarrollado podrá prevenir los problemas encontrados.

La herramienta desarrollada debe ser independiente a una interfaz de trabajo por lo cual a futuro le permitirá integrarse a otras implementaciones.

Puede procesar varios documentos al mismo tiempo, por lo cual puede generar resultados de manera más eficiente que una persona al realizar la tarea de transcripción de texto.

CAPÍTULO 2

2. METODOLOGÍA

En este capítulo se detalla la metodología utilizada para el desarrollo del proyecto. La metodología se encuentra dividida en dos etapas:

- Análisis de documentos y levantamiento de información
- Análisis de diseño y selección de herramientas.

2.1 Análisis de documentos y levantamiento de información

En la primera etapa se revisó y analizó los documentos de importación en colaboración del usuario encargado de transcribir la información del documento escaneado al formulario electrónico, por lo cual se fijó como objetivo recopilar la información necesaria y requerimientos para el desarrollo del proyecto.

Se definió el alcance y las funcionalidades del sistema acorde al análisis del problema, se separó la parte funcional de procesamiento de imagen, motor OCR y la interfaz para tener una mayor flexibilidad al momento de realizar modificaciones o mejoras al sistema.

Se ejecutó una serie de pruebas de la efectividad de extracción de información de la herramienta OCR con varias imágenes modificadas con la finalidad de observar el comportamiento y resultados generados ante los diversos escenarios de variaciones de calidad de imagen definidos, esto permitió establecer las necesidades funcionales del sistema y la división por módulos independientes enfocados a resolver los diversos problemas encontrados.

Los problemas más comunes identificados en los documentos escaneados (Figura 2.1).

- Orientación de texto
- Brillo y contraste de la imagen
- Distorsión de perspectiva de la imagen.
- Desplazamiento en la imagen.

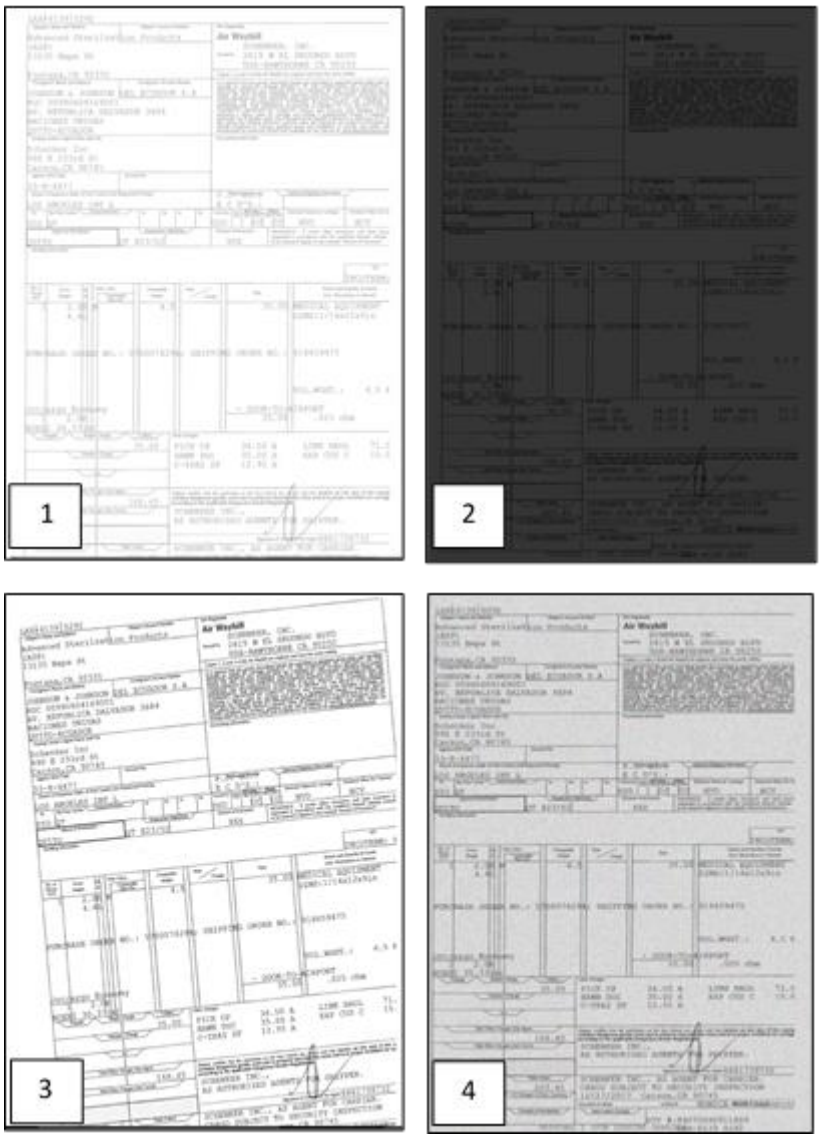

Figura 2.1: Imágenes modificadas 1) 2) Nivel de Brillo 3) Rotación 4) Ruido.

Una vez identificados los problemas se procedió a realizar pruebas con el motor de reconocimiento Tesseract; con el objetivo de evaluar el impacto en los resultados del reconocimiento de caracteres. Se identificó problemas que inciden directamente en los resultados obtenidos de entre los cuales tenemos:

- Orientación incorrecta del texto
- Presencia de sombras no uniforme en diferentes partes del documento.
- Baja resolución de la imagen.

Los resultados obtenidos de las pruebas realizadas, ver tabla 2.1.

Descripción	Porcentaje
Imagen original	95%
Imagen distorsionada	71%
Imagen rotada	5%
Imagen desenfocada	18%
Imagen con ruido pimienta	74%
Imagen con alto brillo	95%
Imagen con bajo brillo	95%
Imagen redimensionada	71%
Imagen modificada en forma	49%

Tabla 2.1: Resultados obtenidos en pruebas de imágenes modificadas

Como se puede apreciar el rendimiento del intérprete OCR se ve afectado directamente por la calidad de la imagen.

A partir de las pruebas realizas el principal problema a corregir es la rotación en las imágenes, lo cual genera pérdida de información y una interpretación errónea. Se realizaron pruebas con imágenes rotadas en diferentes ángulos, para así definir el ángulo de inclinación el intérprete OCR empieza a verse afectado, tal como se muestra en la Tabla 2.3 disminuye con un ángulo mayor a 3 grados.

Descripción	Porcentaje
Imagen rotada 3°	87%
Imagen rotada 5°	11%
Imagen rotada 8°	0%
Imagen rotada 15°	0%
Imagen rotada 180°	1%

Tabla 2.2: Resultados obtenidos en las pruebas con imágenes rotadas en diferentes ángulos.

Las imágenes modificadas por nivel de brillo no tuvieron una afectación tan notable. Aun así, los otros casos de prueba generaron una mala interpretación y pérdida de información. Son casos poco comunes en los documentos, pero es importante tenerlo en cuenta.

Los documentos escaneados con los que trabajará el Sistema deben ser estructurados y mantener la posición de los campos impresos, este tipo de documento contiene cuadriculas que rodean el texto impreso del documento; Lo cual es un inconveniente al momento de procesar la imagen ya que puede existir texto sobreimpreso o texto acompañado de líneas que pueden generar una mala interpretación del OCR.

Se realizaron pruebas del motor OCR en documentos estructurados, se comparó el intérprete al procesar una imagen escaneada que contiene todos los campos comparado con el procesamiento de segmentos de la imagen que contengan únicamente un campo en específico. Se obtuvo mejores resultados a nivel de tiempo en el procesamiento por segmentos de la imagen, cada una de estas secciones como se mencionó anteriormente suelen contener parte de líneas verticales y horizontales de la cuadricula del documento. Es recomendable aislar todo lo que pueda influir en la forma de las letras o interpretación de la palabra para obtener mejores resultados.

Las pruebas realizadas del motor OCR sobre el documento completo y sobre los segmentos de imágenes generaron los siguientes resultados:

Texto Procesado	Documento Original	Documento dividido
LAX-6139 0292	85%	85%
Advanced Sterilization Products	100%	97%
(ASP)	100%	100%
13135 Napa St	100%	100%
JOHNSON & JOHNSON DEL ECUADOR S.A	100%	97%
RUC 00990604169001	100%	94%
AV. REPUBLICA SALVADOR 3684	100%	96%
LOS ANGELES INT L	100%	94%
QUITO	80%	80%
USD	100%	100%
2.0K	100%	75%
MEDICAL EQUIPMENT	100%	94%
203 . 45	100%	100%
10/27/2017	70%	100%

Tabla 2.3: Resultados obtenidos en la interpretación de Tesseract entre un documento completo y uno dividido por campo.

En la Tabla 2.3 se puede observar que si bien, hay campos en los que difiere la interpretación un carácter al texto original, esto puede ser corregido al detectar la presencia de segmentos de líneas en las imágenes (Figura 2.2). Además de generar resultados en menor tiempo y proveer un mejor manejo sobre los campos a extraer al tener un orden a diferencia del análisis por documento completo que genera un texto sin estructura ni orden.

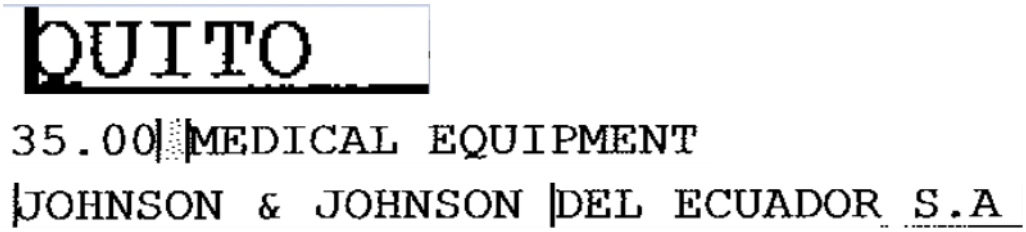

Figura 2.2: Campos de la imagen segmentada

Con finalidades de prueba los segmentos que contenían líneas y puntos que no eran parte de las palabras fueron removidos manualmente con un editor de imagen (Figura 2.3), de esta manera se procedió a evaluar la interpretación del OCR bajo un mejor escenario. Los resultados obtenidos son mostrados en la Tabla 2.4.

Texto Procesado	Original dividido	Segmentos Corregidos
LAX-6139 0292	85%	100%
Advanced Sterilization Products	97%	100%
(ASP)	100%	100%
13135 Napa St	100%	100%
JOHNSON & JOHNSON DEL ECUADOR S.A	97%	100%
RUC 00990604169001	94%	100%
AV. REPUBLICA SALVADOR 3684	96%	100%
LOS ANGELES INT L	94%	100%
QUITO	80%	100%
USD	100%	100%
2.0K	75%	100%
MEDICAL EQUIPMENT	94%	100%
203 . 45	100%	100%
10/27/2017	100%	100%

Tabla 2.4: Resultados obtenidos en la interpretación de Tesseract en segmentos con errores y segmentos corregidos manualmente.

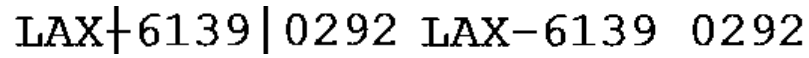

Figura 2.3: Corrección manual de segmentos

De esta manera se pudo establecer que bajo un escenario en el cual los segmentos de imágenes contienen únicamente texto se generan resultados, con un porcentaje de efectividad por encima del 95%.

Adicional a la corrección de calidad del documento también es necesario detectar, extraer, depurar, mejorar e interpretar el texto en las secciones del documento en las cuales es dividido acorde a lo definido previamente por el usuario, para posteriormente generar un archivo estructurado con la información deseada para el usuario.

De esta manera la estructura generada podrá ser consumida por aplicaciones móviles, aplicaciones web o de escritorio.

2.2 Análisis de diseño y selección de herramienta.

En la actualidad existen una gran variedad de programas que permiten realizar reconocimiento de texto en imágenes obteniendo resultados bastantes buenos, la mayoría de estas herramientas tienen un costo por licencia y suelen ser poco flexibles a las necesidades de negocio de las empresas de importación como las listadas en la sección 1.6

El software implementado requiere realizar procesamiento de imágenes para corrección y mejora de la calidad de la imagen acompañada de un motor OCR, que permita interpretar el texto de las imágenes.

Emgu CV

Es un paquete opensource de .NET que contiene más de 500 funciones para procesamiento de imágenes y aplicaciones de visión artificial como reconocimiento de objetos, reconocimiento facial, calibración de cámaras y visión robótica.

Provee un entorno de desarrollo fácil de utilizar y altamente eficiente, gracias a su programación en código C y C++ optimizados, aprovechando las capacidades de procesadores multinúcleo. Puede ser ejecutado en Windows, Linux, Mac OS X, iOS, Android y Windows Phone.

Al estar basado en C permite aprovechar al máximo las operaciones en múltiples procesadores, además contiene una amplia documentación que facilita el uso de sus librerías en el procesamiento de imágenes gracias a sus estructuras y funciones que permiten implementar de manera sencilla diversos algoritmos de procesamiento de imagen.

Se escogió EMGU CV por ser un paquete de código abierto dirigido exclusivamente C#, el cual es una versión mejorada del lenguaje de programación orientada a objetos C++. Lo que permite un manejo más sencillo de las imágenes mediante sus atributos y métodos basados en OPENCV.

Entre las funciones utilizadas de esta herramienta en el proyecto, en [15]:

- Representación y fácil acceso a los valores de intensidad de pixel de la imagen mediante matrices.
- Operaciones lógicas entre matrices.
- Operaciones morfológicas.
- Operaciones de erosión y dilatación.
- Filtros para mejorar calidad de color de la imagen.
- Funciones de aproximación y detección de formas rectangulares.
- Función de HoughLinesP.

La aplicación de las diferentes funciones y estructuras permiten implementar las soluciones a las diversas necesidades del proyecto propuesto como lo son:

- Segmentación de imagen.
- Corrección de calidad de imagen.
- Detección y eliminación de formas geométricas en la imagen.
- Aplicación de máscara a imagen (Máscara definida por Layout).

Tesseract OCR

Es un motor de reconocimiento de texto Opensource distribuido por Google bastante preciso que genera resultados con un porcentaje de acierto por encima del 95%, siendo una herramienta multiplataforma basada en librerías de procesamiento de imágenes de OpenCV.

Fue seleccionado por su alto porcentaje de acierto, mayor al 95% basado en pruebas realizadas por la comunidad que podemos encontrar en su repositorio de Github.

Tesseract, da la opción de detectar el texto en una imagen independiente a la estructura del documento. Permite encontrar las regiones de texto, por medio de líneas de texto del documento para posteriormente proceder a reconocer las palabras que son clasificadas acordes al tessdata definido. Un tessdata es un conjunto de caracteres y fuentes almacenadas en archivos las cuales se puede interpretar los gráficos al motor OCR. Además, contiene un módulo de análisis lingüístico que posibilita clasificar la segmentación de las palabras, agrupándolas acorde a la distancia y ubicación de los caracteres en las áreas de texto.

El training data o tessdata de Tesseract contiene 20 muestras de 94 caracteres en 5 diferentes fuentes, puede ser ampliado mediante entrenamiento, lo cual permite crear nuevas definiciones al tessdata inicial, obteniendo mediante la adición de nuevas formas de letras una amplia colección para las diferentes distorsiones de caracteres que pueden encontrarse en un documento.

Tiene soporte para UTF-8 y puede reconocer más de 100 idiomas, admite varios formatos de salida: texto plano, pdf, tsv y html.

Antes de seleccionar esta herramienta se realizaron diferentes pruebas en las cuales se comprobó la eficiencia del intérprete al procesar varias imágenes.

Obtiene mejores resultados cuando la calidad de la imagen es bastante buena se recomienda que sea de 300dpi, es por ello que en la mayoría de desarrollos la herramienta va complementada con procesamiento de imágenes; realizando operaciones de re escalamiento de imagen, binarización, eliminación de ruido, rotación de imagen y eliminación de bordes.

Esta basado en reconocimiento de caracteres mediante un clasificador adaptativo, el cual mediante el número de coincidencias encontradas con el training set de la herramienta genera una interpretación del texto en la

imagen. Para su funcionamiento es necesario descargar las librerías que contienen el motor OCR junto al tessdata desde su repositorio en github, existen múltiples archivos tess para múltiples propósitos en lo que se refiere a lenguaje, eficiencia y tiempo de ejecución.

Para el proyecto desarrollado se configuró al motor OCR de Tesseract con los lenguajes de español e inglés con un archivo tessdata que prioriza el tiempo de ejecución, y en modo de reconocimiento por defecto que tiene un nivel de aciertos bueno con un tiempo de ejecución bastante corto, los otros modos del motor son más efectivos pero toman mayor tiempo y están más dirigidos a análisis de documentos completos en el caso del sistema a implementar no es necesario ya que se envía a procesar segmentos de imágenes solo con texto.

CAPÍTULO 3

3. SOLUCIÓN

En este capítulo se detalla el diseño e implementación del proyecto.

3.1 Diseño e implementación de módulos

El proyecto acorde a las necesidades planteadas en el capítulo anterior fue dividido en 6 módulos ver Figura 3.1, que serán explicados a detalle en esta sección.

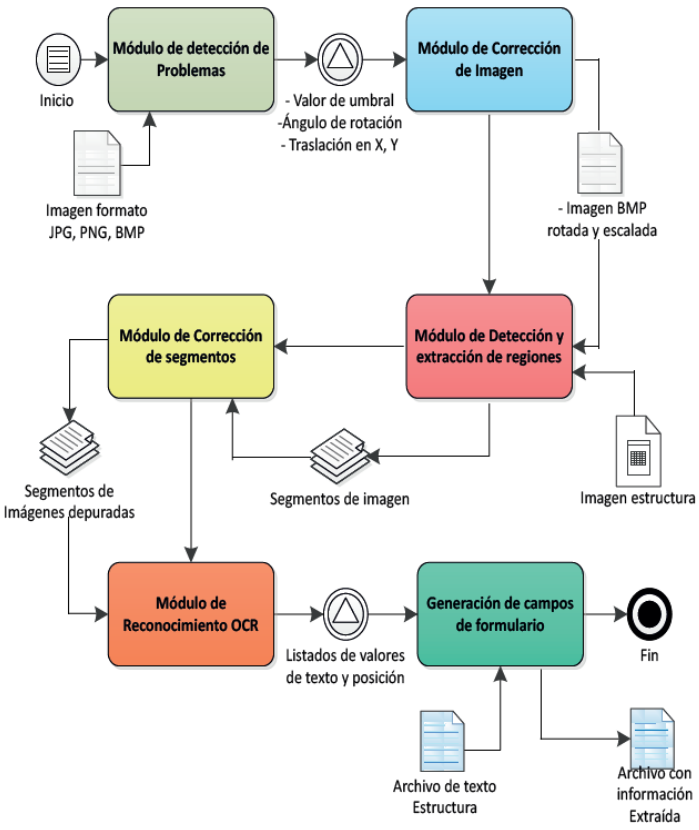

Figura 3.1: Diseño de aplicación

Debido a que la mayoría de los módulos tienen como objetivo el mejorar la calidad de imagen y realizar correcciones necesarias, es importante indicar que se utilizó funciones de procesamiento proporcionadas por las librerías de EMGU CV. Las imágenes son representadas mediante estructuras

basadas en matrices que provee EMGU CV. El programa desarrollado admite cualquier tipo de imagen las cuales generalmente se encuentran definidas en modo RGB (Red Green Blue).

Cada canal de la imagen es una matriz M de dimensión n x m x c; n definido por la altura de la imagen expresado en el número de filas de la matriz, m determinado por el ancho de la imagen expresado en su número de columnas y c dado por el número de canales de la imagen. Con esta representación de la imagen podemos acceder al valor de cada pixel representado como M[i,j,k] siendo i, el número de fila de la imagen, j es el número de columna y k que hace referencia a su canal de color de la imagen que representa el eje z de la imagen. El valor obtenido de M[i,j,k] está expresado por valores del 0 al 255, donde cada valor representa la intensidad del canal de ese punto de la imagen.

Basados en esta representación a lo largo de este capítulo se irá explicando a detalle el procesamiento realizado en cada uno de los módulos a los diferentes tipos de imágenes, mediante algoritmos correctivos que manejan las diversas características de la imagen a nivel de pixel.

Para el funcionamiento de los módulos 3 y 6 ver Figura 3.1, es necesario disponer de dos archivos definidos previamente por el usuario:

- **Archivo Layout:** Es una imagen que contiene los campos de interés a extraer del documento representado por una matriz de valores binarios mediante formas rectangulares que representan las regiones que contienen información a extraer, como se muestra en Figura 3.2.
- **Archivo de estructura:** Es un archivo de texto que contiene la estructura de datos de los nombres de los campos a extraer acompañado con el valor de posición de los ejes X, Y de la imagen (Figura 3.2).

Figura 3.2: Layout y campos del documento

3.1.1 Módulo de detección de problemas

En este módulo se detectan los problemas relacionados a la calidad de la imagen mediante las siguientes operaciones:

- **Cálculo de promedio de umbral:** Se implementó la función “valorUmbral” que calcula sobre la imagen en escala de grises el valor promedio de la intensidad de color en los pixeles de la imagen. Este valor es asignado a una variable global definida como “umbral”.
- **Detección de ángulo de rotación de la imagen**: Mediante la función “getSkewAngle” de la librería “Aforge" se calcula el ángulo de inclinación basado en la función de Hough la cual detecta las líneas en la imagen y determina su ángulo de inclinación obteniendo el promedio de los ángulos calculados de cada una de las líneas encontradas.
- **Detección de desplazamiento de imagen:** Se implementó un algoritmo que recorre la imagen de izquierda a derecha y de arriba hacia abajo, hasta encontrar el primer y último conjunto de pixeles consecutivos cuyo valor de intensidad sea menor al valor de la variable “umbral” encontrado previamente. Una vez encontrada la posición del grupo de pixeles se obtiene los puntos máximos y mínimos de estos pixeles consecutivos, lo cual define las esquinas superior izquierda e inferior derecha del área de interés de la imagen y permite extraer únicamente la sección de interés del documento.

Con los valores calculados en los puntos 2 y 3 se puede detectar los problemas que presenta la imagen y dar aviso al usuario:

- Si el ángulo es diferente de cero se mostrará mensaje de retroalimentación mostrando el ángulo de rotación, permitiendo corregir su inclinación en el siguiente módulo.
- Mediante el punto mínimo calculado se detecta si existe traslación en la imagen, notificando mediante mensaje al usuario.

3.1.2 Módulo de corrección y mejora de calidad de imagen

El módulo de corrección permite mejorar la calidad de la imagen para obtener mejores resultados en el reconocimiento de texto utilizando el motor OCR de Tesseract, mediante los siguientes pasos:

- Rotación de imagen con ángulo detectado mediante algoritmo de Houghlines Probabilístico.
- Recortar área de interés de la imagen con coordenadas mínima y máxima.
- Escalamiento de imagen a dimensiones de layout definido por documento.

En caso de realizar correcciones sobre la imagen procesada se mostrará un mensaje en la pantalla como se puede observar en Figura 3.3.

Es importante resaltar que en las pruebas realizadas con imágenes modificadas de manera intencional con problemas de rotación, traslación y diferentes dimensiones producía interpretaciones con una tasa de acierto por debajo del 10%, el módulo de corrección mejorará un 70%.

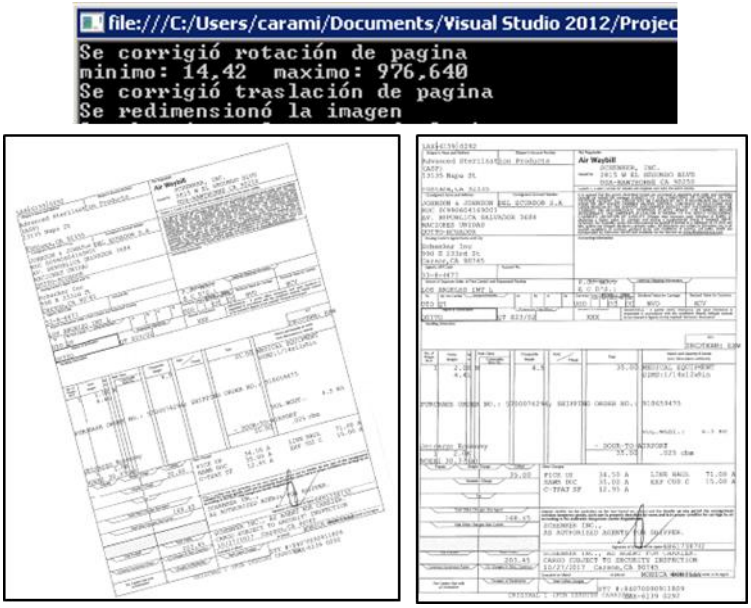

Figura 3.3: Corrección de rotación de imagen

3.1.3 Módulo de detección y extracción de regiones de texto

El módulo de detección de regiones se trabaja con el archivo de imagen que contiene la estructura del documento. En la Figura 3.4 se observa que los pixeles blancos representan el área de interés a extraer y los pixeles negros las áreas a excluir de la imagen.

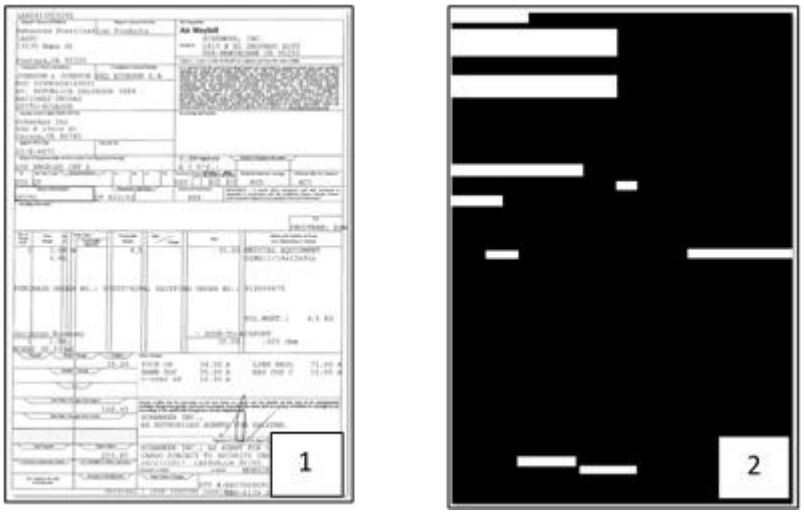

Figura 3.4: 1) Imagen original 2) Layout de documento

Con el archivo a procesar y el archivo de estructura se procede a realizar una operación AND (conjunción) entre los valores de intensidad de color de ambas imágenes representados en dos matrices. El resultado obtenido es una imagen filtrada la cual contiene solo las regiones a extraer. Posteriormente se procede a realizar una dilatación con un elemento estructural horizontal con la finalidad de marcar las líneas de texto del documento.

La imagen con líneas horizontales que representan la posición del texto en el documento, permite calcular la altura aproximada de las letras de la imagen, mediante el cálculo del valor promedio de cada una de las líneas de texto marcadas. Este valor será denominado “tamañoLetra”, el cual permitirá calcular el tamaño de varios elementos estructurales que serán utilizados para realizar operaciones de procesamiento de imágenes en los siguientes módulos.

Con el valor “umbral” encontrado en el módulo 1, se procede a realizar las siguientes operaciones a la imagen, con la finalidad de detectar la ubicación de los campos que se va a extraer.

i. Aplicación de Threshold Otsu acotado por valor de umbral.
ii. Inversión de colores de imagen.
iii. Detección y eliminación de líneas utilizando algoritmo de Hough (Figura 3.5).
iv. Operación de erosión y dilatación de texto basado en tamaño de letra aplicada una vez sobre la imagen obtenida en punto 3.
v. Operación de closing con estructura basada en tamaño de letra aplicada una vez sobre la imagen obtenida en punto 4.
vi. Operación de erosión y dilatación aplicada una vez sobre la imagen obtenida en punto 5 (Figura 3.6).
vii. Aproximación de contornos a forma poligonal generada.
viii. Obtención de lista de rectángulos que contienen texto y su posición/orden correspondiente.
ix. Generación de segmentos de imágenes basadas en áreas rectangulares detectadas, mostrado en Figura 3.7.

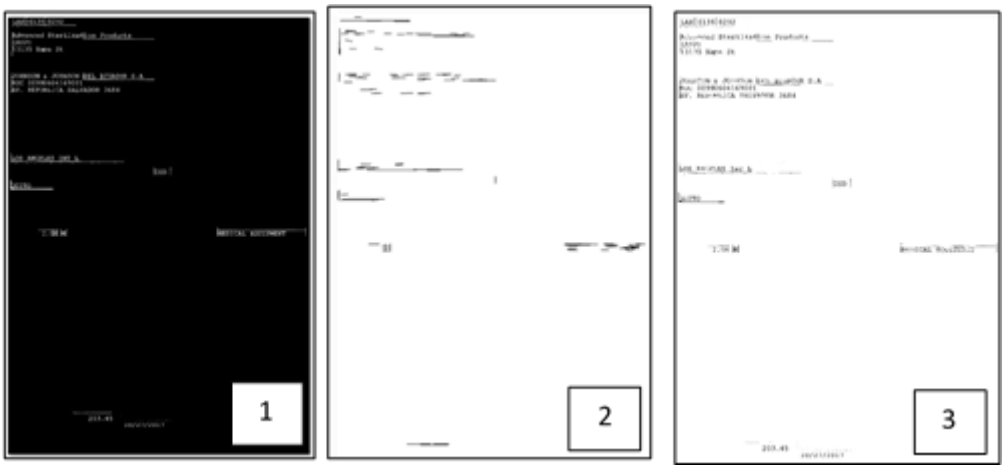

Figura 3.5: 1) Aplicación de máscara 2) Detección de líneas 3) Inversión y eliminación de líneas.

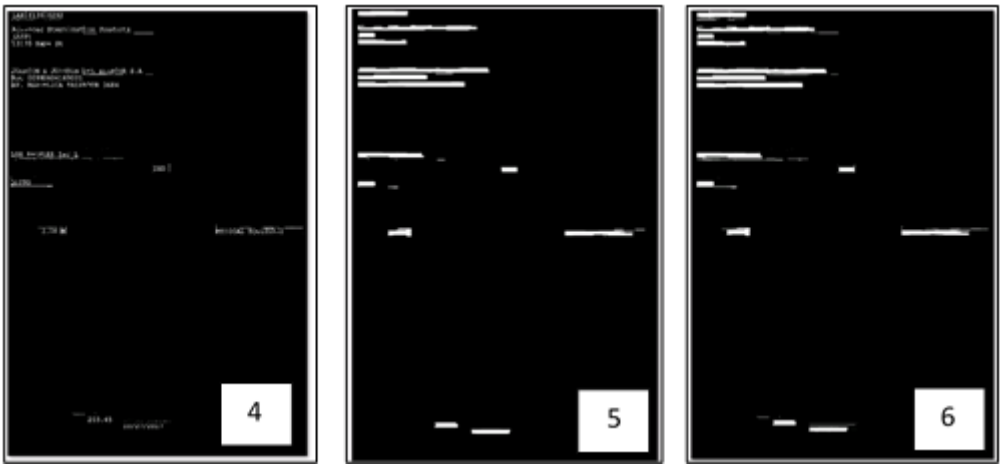

Figura 3.6: 4) Erosión y dilatación 5) Closing 6) Erosión y Dilatación

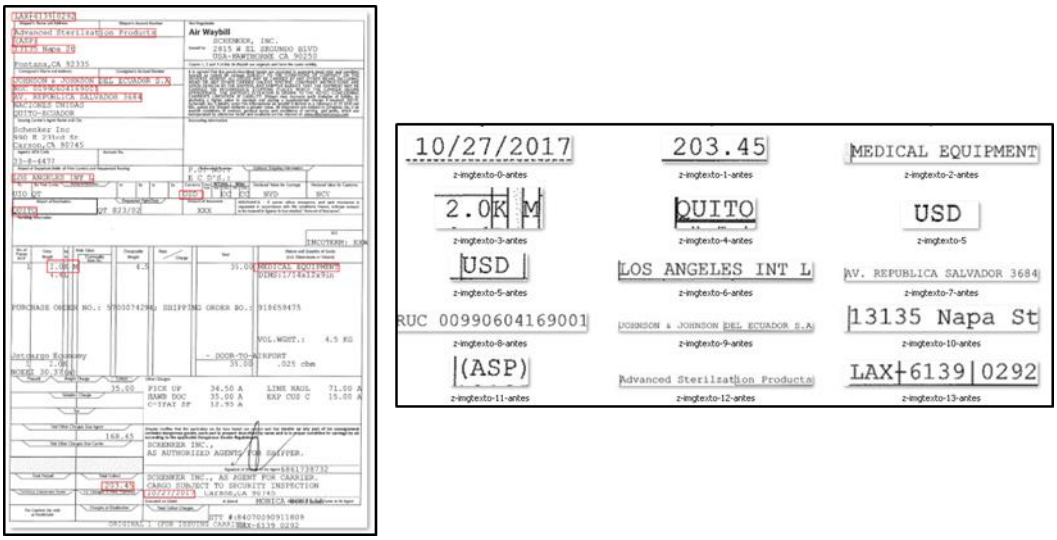

Figura 3.7: Detección y generación de campos en segmentos de imágenes

Una vez realizadas las operaciones listadas anteriormente se obtiene una lista que contiene cada uno de los segmentos de imagen con los campos que se desea a extraer con la siguiente información (Tabla 3.1):

SegmentoImagen	
orden	Representa el orden de extracción del segmento
posicionX	Número de columna en la imagen
posicionY	Número de fila en la imagen
altura	Número de filas que contiene el segmento
ancho	Número de columnas que contiene el segmento

Tabla 3.1: Representación de segmento de imagen.

Como resultado de este módulo el sistema ha generado k segmentos de imagen de las secciones del documento que contienen los campos de interés definidos por el usuario, dividiendo la imagen en n imágenes acorde al número de campos definidos en la estructura inicial.

Si el sistema no llega a encontrar el número de campos listados en el archivo de estructura, no afectará su comportamiento, podrá detectar y generar los demás campos encontrados.

Este escenario puede presentarse por dos razones si en el documento analizado no existe texto en el área definida por el usuario y por una mala aplicación de las operaciones de procesamiento de este módulo.

3.1.4 Módulo de corrección de errores de morfología

Los segmentos de imágenes obtenidos en el módulo anterior son tomados de la imagen original, por esta razón contienen puntos y segmentos de líneas verticales u horizontales cercanas al texto, lo cual podría generar una interpretación errónea del motor de reconocimiento.

Este inconveniente se genera en los documentos con una estructura determinada, los cuales generalmente contienen cuadriculas alrededor del texto impreso, en ciertas ocasiones estas líneas pueden llegar a afectar la morfología de las letras distorsionando su forma.

En las pruebas realizadas en la sección 2.1 de la herramienta OCR al analizar segmentos de imagen con este tipo de formas adicionales

alrededor del texto generaba en su interpretación caracteres adicionales como: comas, guion bajo, puntos, letras I, L, Y signos de admiración donde realmente no existen.

Este módulo tiene como objetivo el identificar líneas que no forman parte de las palabras para así aislarlas y eliminarlas, basándose en su dimensión y tamaño de letra encontrado anteriormente. Es importante tener en cuenta que al eliminar este tipo de formas no propia de los caracteres impresos podría producir pérdida de información, eliminación de letras o parte de letras al eliminar líneas por error las cuales sean parte del texto a interpretar.

Los segmentos de imágenes extraídos tienen una dimensión mayor al texto que contienen definidas de esta manera en el módulo anterior, con la finalidad de poder aplicar operaciones de procesamiento para depurar los segmentos de imagen.

Se evalúa si un conjunto de píxeles consecutivos dentro de la imagen son parte de una letra, caso contrario si se trata de una línea que ocupa toda una columna o parte de ella, ya que una letra pocas veces podría ocupar la totalidad de la altura del segmento extraído.

Figura 3.8 : Separación de fondo utilizando threshold

El proceso de detección y eliminación consta de los siguientes pasos:

i. Cálculo de valor promedio de intensidad de color por cada segmento.
ii. Aplicación de threshold otsu a cada segmento acotado por el valor obtenido en el punto 1. Lo cual permite resaltar el texto sobre el demás contenido y corregir los problemas de brillo de la imagen, Ver Figura 3.8.
iii. Encontrar porcentaje de pixeles diferentes al color blanco por columna en base a altura de segmento de imagen y altura de letra.

iv. Se realiza un recorrido columna a columna donde se suma el número de pixeles pintados en cada columna, de esta forma se obtiene una lista del número de pixeles con valor de intensidad menor al umbral por cada columna de la imagen. Luego se procede a evaluar el número de pixeles de cada columna comparándolo con el porcentaje de pixeles calculado anteriormente que debería tener el segmento de imagen acorde a su altura, mediante la proporción entre númeroPixelesCi/alturaSegmento*100 se obtiene el porcentaje de pixeles pintados, como se puede visualizar en Tabla 3.2, si este valor se encuentra por encima del porcentaje que debería tener la imagen conforme al tamaño de la letra, los pixeles de la columna toman el valor de 255 es decir son blanqueados. De esta forma se elimina las líneas verticales de la imagen, encontradas en las columnas que tienen una mayor concentración de pixeles mayor la altura definida previamente de las letras (Figura 3.9).

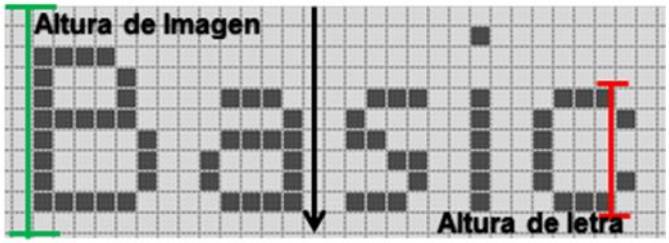

Figura 3.9: Palabra con altura y recorrido por columna.

Columna	1	2	3	4	5	6	7	8	9	10
Numero de pixeles	0	8	3	3	3	4	3	0	0	2
% de pixeles	0%	73%	27%	27%	27%	36%	27%	0%	0%	18%

Tabla 3.2: Resultados obtenidos del análisis de pixeles por columna

v. Se realiza un procedimiento similar al punto anterior, mediante un recorrido fila por fila se obtiene el número de pixeles diferentes de blanco por cada fila, luego evalúa la cantidad de pixeles en cada fila con la proporción númeroPixelesFj/anchoSegmento*100 como se puede observar en Tabla 3.3. Si el porcentaje encontrado

se encuentra encima del porcentaje estimado acorde al tamaño del segmento se procede a eliminar la fila. Es importante destacar que el porcentaje de pixeles por fila es mucho menor que el de columnas ya que generalmente las letras suelen ir separadas por espacios y esto ocasiona que el porcentaje de pixeles se encuentre debajo del 50%, como se observa en la Figura 3.10,3.11.

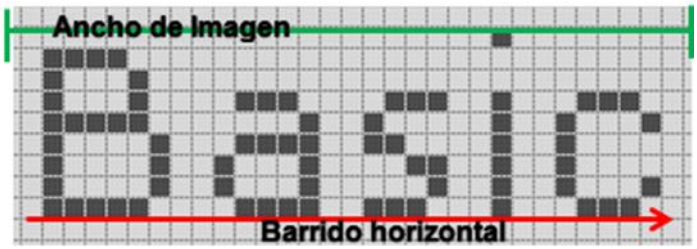

Figura 3.10 : Palabra con ancho y recorrido por fila.

Fila	1	2	3	4	5	6	7	8	9	10
Numero de pixeles	0	1	4	12	10	10	8	8	8	17
% de pixeles	0%	3%	11%	34%	29%	29%	23%	23%	23%	49%

Tabla 3.3: Resultados obtenidos del análisis de pixeles por fila.

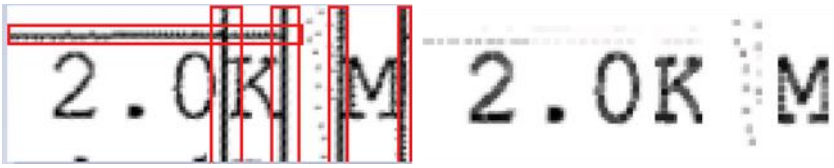

Figura 3.11: Detección y eliminación de líneas en segmentos.

vi. Una vez eliminadas líneas verticales y horizontales se procede a identificar los pixeles que no forman parte del texto a procesar. Se delimita la imagen verticalmente, para esto es necesario encontrar posición mínima del pixel que tenga un valor de intensidad de color por debajo del umbral de la imagen columna a columna por cada segmento. Se recorre de forma horizontal cada una de las columnas de la imagen como se puede observar en Figura 3.12, se toma la posición mínima del pixel que cumple

la condición detener un valor menor que el umbral cuya posición se encuentra en la fila más alta cercana al eje y=0 de cada una de las columnas. Para posteriormente proceder a calcular la media de las posiciones mínimas encontradas en cada columna, así se obtiene un valor mínimo por imagen, ver tabla 3.4.

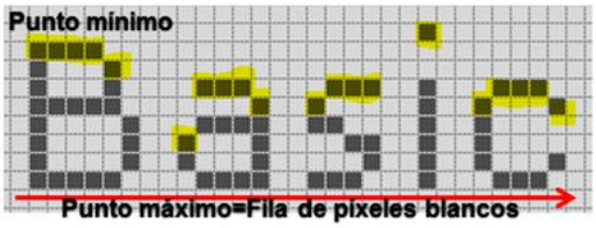

Figura 3.12: Detección de píxeles mínimos y detección de fila delimitadora máxima.

Columna	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Valor punto mínimo	3	3	3	3	4	5	5	5	5	5	5	6	6	2	5	5	5	6

Valores ordenados:	2	3	3	3	3	4	5	5	5	5	5	5	5	5	5	6	6	6
Media:	5																	

Tabla 3.4: Resultados de cálculo de posición mínimo.

vii. Con el valor mínimo encontrado se calcula el valor máximo de la imagen de manera diferente al punto anterior. Al tener un segmento de imagen de altura = n filas, se procede a buscar las filas que tengan una cantidad de pixeles blancos que abarque la mayor parte de su ancho, es decir las filas vacías ubicadas bajo la mitad de la imagen. De esta forma se puede identificar en qué posición termina el área que ocupa el texto de la imagen. Con el valor de la fila mínima se encuentra la fila máxima. Mediante un barrido fila por fila desde n/2 a n, se obtiene la cantidad de pixeles blancos, si tiene un 95% de pixeles blancos en relación al ancho de la imagen, la posición Y encontrada será el valor máximo.

viii. Una vez encontrados lo valores máximos y mínimos como se observa en figura 3.13 se procede a acotar la imagen, al blanquear las filas que se encuentren por debajo del mínimo y por encima del máximo. Es importante tener en cuenta que muchas

palabras suelen tener letras que tienen un tamaño por encima o debajo de la posición de las demás letras por ejemplo las letras l,t,p,q,y,g; por lo cual el acotamiento considera aumentar el rango encontrado. Para el valor mínimo se disminuye un quinto del tamaño de la letra calculado y al valor máximo se aumenta un décimo del tamaño de la letra de esta manera se obtendrá mejores resultados sin eliminar parte de letras que produzcan perdida de información en el texto (Figura 3.14) (Tabla 3.5).

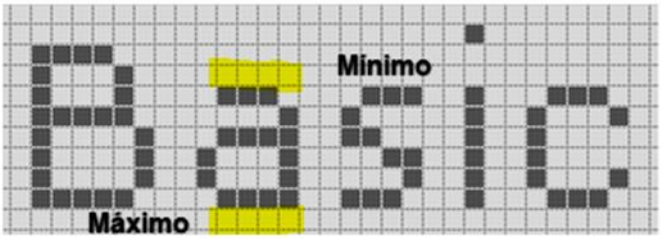

Figura 3.13: Delimitador mínimo y máximo marcado en la palabra.

Altura calculada letra X:	8
Adicional mínimo X/5:	2
Adicional máximo X/10:	1

	Inicial	**Final**
Valor mínimo	5	3
Valor máximo	10	11

Tabla 3.5: Cálculo de nuevo valor máximo y mínimo.

ix. Una vez realizado el acotamiento y eliminación de líneas se procede a aplicar erosión para eliminar píxeles sueltos y dilatación para resaltar el contorno de las letras de la imagen.

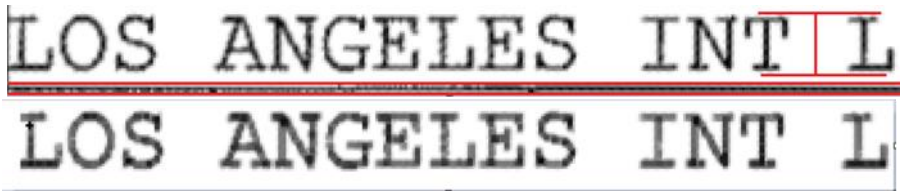

Figura 3.14: Eliminación de áreas fuera del rango min-max.

3.1.5 Módulo de reconocimiento OCR

El modulo OCR según las necesidades del usuario fue implementado como un módulo independiente capaz de procesar varias imágenes al mismo tiempo.

El motor OCR toma el listado de segmentos de imágenes depuradas en el módulo anterior procesándolos en paralelo, 3 segmentos de imágenes a la vez, fue configurado de tal manera que se tenga una mejor calidad en la interpretación en el menor tiempo posible. Al tener la ventaja de procesar imágenes pequeñas que contienen únicamente texto se obtiene mejores resultados con un modo de configuración por defecto del motor OCR que realiza un análisis plano sobre líneas de texto y no sobre un documento completo en el cual requeriría mayor procesamiento para reconocer e interpretar las secciones de texto.

3.1.6 Módulo de generación de campos de formulario

El módulo de generación de campos utiliza el archivo de estructura que contiene la posición y nombre de los campos del documento. Para poder identificar cada uno de los segmentos de imagen que fueron depurados y se dispone de ellos en una lista que contiene la imagen, orden de extracción, posición X, posición Y, alto y ancho. Se complementa la información del archivo de estructura con la información que contiene la lista de segmentos se realiza un recorrido de la lista en donde se compara la ubicación del segmento con cada una de las posiciones del archivo de estructura tanto para X y Y.

Para la posición Y_1 (número de fila donde se encuentra ubicado el segmento de imagen), se procede a restar el Y de cada uno de los campos del archivo con la posición Y del segmento de imagen, se calcula el valor absoluto del resultado y si el valor hallado se encuentra dentro del rango $\mathrm{Y} + \mathrm{tamañoTexto} * 0.5$,$\mathrm{Y} - \mathrm{tamañoTexto} * 0.5$ se cumple la primera condición para identificar a que campo pertenece el segmento analizado.

De igual manera se procede a comparar los valores X del archivo con la posición X_1 del segmento analizado. Se calcula el valor absoluto de la

diferencia entre ambos valores, el cual debe encontrarse dentro del rango definido por $Y - \text{tamañoTexto} * 2$, $X + \text{tamañoTexto} * 2$ como se muestra en Figura 3.15. Si el valor de posición X está dentro del rango se cumple la segunda condición.

Una vez encontrada la posición del archivo de estructura que cumpla ambas condiciones, el segmento analizado se asigna al campo que encontró coincidencia en su posición.

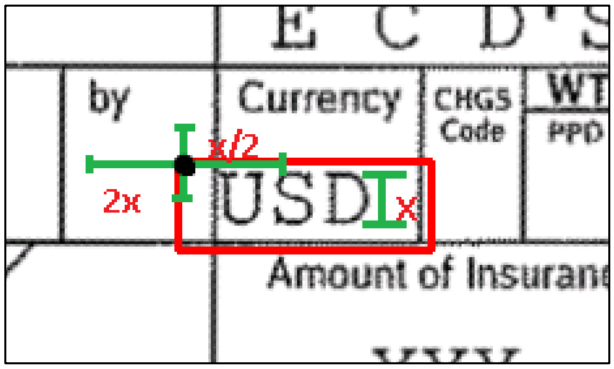

Figura 3.15: Identificación de campo por posición de punto dentro de intervalo.

Una vez asignado el campo a la imagen se genera un archivo estructurado, recorriendo el listado de imágenes con el texto interpretado en el módulo anterior y el nombre del campo identificado de la siguiente manera (Figura 3.16):

```
NombreCampo1:TextoCampo1
NombreCampo2:TextoCampo2
.
.
.
NombreCampoN:TextoCampoN
```

Figura 3.16: Estructura de archivo generado

Se genera un archivo de texto plano con una estructura que puede ser utilizable por cualquier otro programa permitiendo darle usos múltiples, ya que acorde a las especificaciones del usuario, este sistema fue implementado como un programa ejecutable que puede ser integrado a cualquier otra aplicación y tomará los archivos a procesar de un

directorio definido por el usuario del computador para procesar y generar el archivo de salida.

Para ejecutar el programa desarrollado se requiere tener:

- Directorio con archivos tessdata.
- Programa compilado con librerías de Emgu CV y tesseract.
- Imagen con Layout.
- Archivo de texto con estructura de posición/campo.

El sistema OCR de extracción de información de documentos, puede procesar documentos escaneados con diferentes formatos siempre y cuando se defina los dos archivos de estructura que permiten realizar la extracción e identificación de los campos.

Al iniciar el programa ejecutable debe enviar los siguientes parámetros a través de línea de comandos mostrados en Tabla 3.6.

Parámetros	**Módulos**					
	1	**2**	**3**	**4**	**5**	**6**
Unidad de disco.	X	X	X	X	X	X
Nombre de archivo.		X				
Nombre de imagen con estructura			X			
Nombre de archivo txt con estructura						X

Tabla 3.6: Parámetros utilizados por módulo.

CAPÍTULO 4

4. RESULTADOS

En este capítulo se muestra los resultados del proyecto desarrollado, obtenidos mediante pruebas que permitirán medir la efectividad de dos aspectos importantes del programa, análisis similares en [16].

- Identificación de áreas a extraer del documento.
- Interpretación del texto de los segmentos de imagen.

Para lo cual se realizó pruebas sobre 10 documentos escaneados, con la finalidad de evaluar el rendimiento del programa desarrollado. Los documentos con los que se realizó las pruebas tenían las siguientes características:

- Formularios de Importación.
- Buena calidad de imagen.
- Texto impreso claro.
- Documento sin presencia de manchas ni tachones.
- Documento sin letra manuscrita.

Estas características fueron definidas luego de realizar pruebas sobre diferentes documentos de importación, entre los cuales existían imágenes escaneadas con problemas no manejables por la herramienta desarrollada.

- Texto impreso con poca calidad.
- Ubicación de texto en diferente posición de la definida en la estructura.
- Presencia de marcas de agua, manchas o formas ajenas a la estructura definida inicialmente.

Los resultados obtenidos en este tipo de documentos tuvieron un porcentaje de aciertos por debajo del 10%, ocasionado principalmente por una mala detección de las áreas que contienen los campos del documento. Al tener texto fuera de posición la máscara aplicada con el archivo de estructura ocasiona un corte en la imagen erróneo, Ver Figura 4.1.

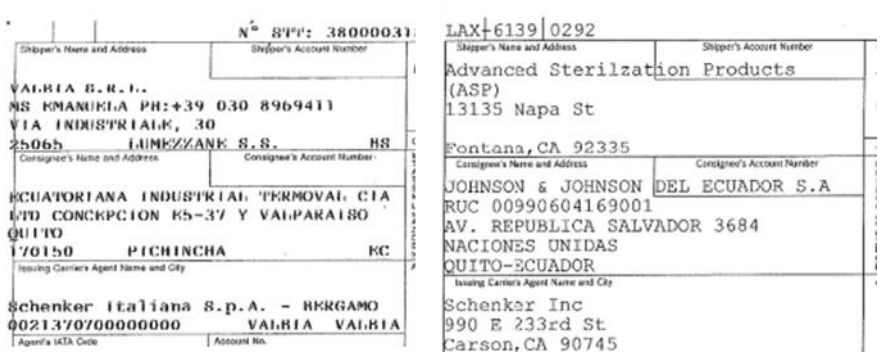

Figura 4.1: Imagen con texto en otra posición

De esta misma manera la presencia de formas no identificadas en la estructura inicial origina un comportamiento similar, al provocar un escalamiento de una región de diferentes proporciones a la de la estructura. El área detectada es diferente a la que debería detectar al encontrar los pixeles de esta forma en los mínimos y máximos encontrados de la imagen. Esto origina una mala aplicación de la macara, Ver Figura 4.2.

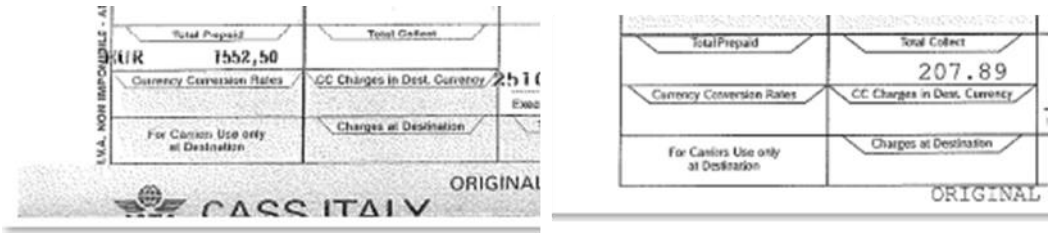

Figura 4.2 : Imagen con logo en parte inferior

El texto con poca nitidez causa pérdida de información en su impresión lo cual provoca encontrar pixeles faltantes en la forma de las letras, Ver Figura 4.3. Esto puede generar una mala lectura del OCR al interpretar al texto ya que un carácter cortado puede ser identificado como dos caracteres o por otro carácter diferente al original.

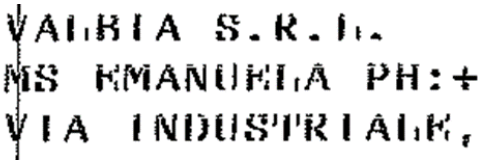

Figura 4.3: Imagen con texto mal impreso

4.1 Prueba de Identificación de áreas a extraer

La primera prueba realizada mide la identificación de las áreas de texto del documento en el módulo 3, según la estructura definida por el usuario. Como se mencionó en la sección 3.1 en el módulo de detección y extracción de segmentos, se realiza diversas operaciones de procesamiento sobre la imagen con la finalidad de segmentar las secciones de texto y tomar únicamente las áreas del documento que contienen los campos de interés. Pero la detección y reconocimiento de estas áreas puede presentar errores al momento de realizar el procesamiento del documento por diversos factores como lo son:

- Presencia de texto que no es parte de la estructura usual del documento.
- Campo a extraer muy cercano a otro campo de interés.
- Texto impreso en diferente tamaño de letra.
- Marcas o gráficos en el documento que no forman parte del documento.
- Impresión de campos desplazados vertical u horizontalmente de su posición establecida.
- Presencia de muchas líneas alrededor del texto a extraer.

Todos estos factores pueden influir en la detección del área de interés del documento, la cual debe coincidir con el tamaño del documento de estructura layout que se crea inicialmente, cualquier variación de contenido podría afectar el tamaño del área detectada. Se modifica la proporción del documento al escalar al tamaño del layout.

Esto provoca que el comportamiento de la herramienta al realizar la máscara sobre el documento, por lo cual recorta partes del documento que contienen otros campos diferentes a los que se desea extraer.

Mientras que la impresión de campos muy cercanos a otros puede generar la unión de ambos campos o la inclusión de otras palabras en un campo, Ver Figura 4.4. Este mismo efecto es provocado si un campo contiene palabras muy separadas esto puede generar la exclusión de parte del campo.

Es por ello que a través de una matriz de confusión [17], se midió el comportamiento de la detección de campos del sistema, el cual fue modelado como un clasificador el cual analiza el documento y conforme la posición del campo respecto al layout es tomada o excluida como área de interés a extraer del formulario.

Los valores obtenidos en la matriz de confusión fueron medidos por número de palabras, se realizó un conteo de las palabras que contiene el documento cuantificando las evaluadas correctamente e incorrectamente según el procesamiento realizado por el sistema.

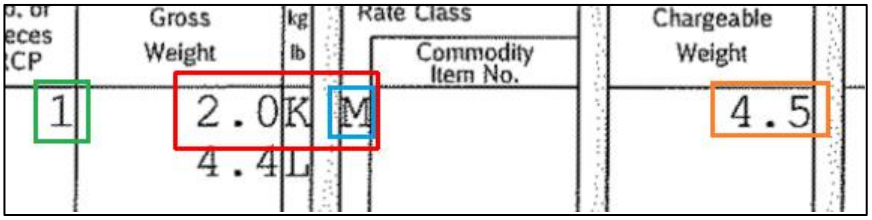

Figura 4.4: Representación de casos detectados por sistema

Los resultados obtenidos se muestran en la Tabla 4.1:

		Clasificación		
		A	**B**	**Total**
Real	**A**	307	8	315
	B	16	423	439
	Total	323	431	754

Tabla 4.1: Matriz de confusión

Caso	Descripción de caso
A	Palabras que pertenecen a área definida en estructura
B	Palabras que no pertenecen a área definida en estructura

96,82%	Palabras clasificadas correctamente

Tabla 4.2: Casos evaluados y porcentaje de casos correctos.

En la cual se evalúa los dos posibles casos al momento de reconocer las áreas de texto del documento como se puede ver en tabla 4.2.

Se obtuvo 307 palabras que formaban parte de los campos de interés del documento identificadas como parte de un área a extraer de la imagen según el layout definido. Mientras que 8 palabras que formaban parte de los campos de interés del documento no fueron identificados como parte de un área a extraer de la imagen, quedando excluidas de los segmentos de imagen generados por el sistema.

Otras 423 palabras que no formaban parte de los campos de interés del documento no fueron identificadas como parte de las áreas a extraer de la imagen excluyéndolas de manera correcta de los segmentos generados por el sistema. Mientras que 16 palabras que no formaban parte de los campos de interés fueron identificadas como parte de un área extraer de la imagen, incluyéndolas de manera incorrecta dentro de los segmentos de imagen generados por el sistema.

4.2 Prueba de Identificación de áreas a extraer

La segunda prueba fue realizada sobre el módulo de interpretación OCR se utilizó los segmentos de imágenes obtenidos de la prueba anterior. Sobre las 323 palabras que se extrajeron se midió la precisión de reconocimiento de texto a partir de los segmentos de imágenes analizadas por el Sistema OCR.

Cabe destacar que en las imágenes generadas por el sistema hay un 97% de palabras correctamente tomadas del documento, mientras que el otro 3% son palabras que no pertenecían al campo extraídas de más o palabras no identificadas que formaban parte del campo a extraer.

En este módulo no se evalúa si el campo generado es correcto por lo cual se considerará el 100% de las palabras que identificó el sistema para extraer del documento. Para evaluar su interpretación como texto a partir de la imagen analizada.

Los resultados generados por esta prueba pueden ser visualizados en la Tabla 4.3:

	Interpretación OCR		
	Correcta	**Incorrecta**	**Total**
Palabras en áreas de texto extraídas	302	21	323

Porcentaje de precisión de Reconocimiento:	93%

Tabla 4.3: Resultados y cálculo de precisión de interpretación.

De 323 palabras analizadas se obtuvo: 302 palabras correctamente interpretadas. Tuvo un 93% de precisión de interpretación del Sistema OCR.

Mientras que 21 palabras fueron interpretadas erróneamente. Con un 7% de error de interpretación de palabras. En estas palabras el OCR interpretaba alrededor de uno a dos caracteres de manera incorrecta, es decir con 5% de error en reconocimiento.

Con la evaluación de los módulos 3 y 5 en cada una de las pruebas se pudo obtener un cuantificador general del Sistema OCR obteniendo el producto entre el porcentaje de palabras extraídas correctamente y el porcentaje de precisión de interpretación. Se tuvo un 90.5% de eficiencia en los resultados generados por el programa implementado. Este porcentaje fue calculado a partir del producto de los porcentajes encontrados en las dos pruebas realizadas en el módulo de detección e interpretación respectivamente con 96.82% y 93%.

En las pruebas realizadas el motor de reconocimiento fue ejecutado en 3 núcleos a la vez y se obtuvo un tiempo promedio de 7 segundos de procesamiento en documentos con 14 campos a extraer.

CONCLUSIONES Y RECOMENDACIONES

A partir de los resultados obtenidos en las pruebas realizadas en las diferentes secciones del documento. Se pudo medir el comportamiento y efectividad del programa desarrollado con lo cual se puede concluir que se cumplieron los objetivos planteados con los siguientes puntos a considerar.

Conclusiones

El programa desarrollado es capaz de reemplazar el trabajo manual, al tener una efectividad por encima del 90% en las interpretaciones generadas en un tiempo de 7 segundos por documento.

El porcentaje de efectividad de interpretación del texto en las imágenes está relacionado directamente a la calidad de la imagen. No obstante es importante destacar que las medidas correctivas implementadas en la herramienta permiten obtener resultados más confiables y tener mayor capacidad de respuesta al poder solventar los problemas más comunes en los documentos de importación.

El diseño de la herramienta permite integrar los resultados generados a cualquier tipo de plataforma tecnológica, por lo cual puede ser utilizado y adaptado según las necesidades del usuario.

Recomendaciones

Como se detalló en el capítulo 4 existen limitaciones de la herramienta desarrollada relacionadas a la calidad del texto impreso y la posición de texto fuera de la estructura definida, las cuales pueden ser consideradas como mejoras a futuro.

Es recomendable trabajar con documentos estructurados con una resolución mínima de 300 dpi.

El tamaño de letra del documento debe ser uniforme, diferentes tamaños podrían ocasionar un mal cálculo al momento de estimar el tamaño de letra de esta manera podría provocar un mal procesamiento de la imagen.

Se recomienda analizar documentos con texto impreso. Si la imagen escaneada contiene texto manuscrito este no podrá ser interpretado, ya que requeriría entrenar el motor de reconocimiento.

BIBLIOGRAFÍA

[1] Lucas S. and Amiri A., "Statistical syntactic methods for high-performance OCR," *IEE Proceedings on Vision, Image and Signal Processing,* vol. 143, pp. 23-30, 1996.

[2] R. Holley, "How Good Can It Get? Analysing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs," *D-Lib Magazine,* vol. 15, pp. 1082-9873, 2009.

[3] V. Wu, R. Manmatha and E. Riseman, "TextFinder: An Automatic System to Detect and Recognize Text In Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 21, pp. 1224-1229, 1999.

[4] G. Yuddy, "MINTECON," 22 Abril 2015. [Online]. Available: http://www.eoi.es/blogs/mintecon/2015/04/22/el-error-humano/. [Accessed Diciembre 2017].

[5] V. Ruben, "SOFTZONE," 14 02 2017. [Online]. Available: https://www.softzone.es/2017/02/14/5-aplicaciones-ocr-pasar-documentos-escaneados-texto/. [Accessed Diciembre 2017].

[6] M. I. De Arquer, "Fiabilidad humana: métodos de cuantificación, juicio de expertos," 1995. [Online]. Available: http://www.oect.es/inshtweb/contenidos/documentacion/fichastecnicas/ntp/ficheros/401a500/ntp_401.pdf. [Accessed Diciembre 2017].

[7] A. Antonacopoulos and A. Brough, "Methodology for Flexible and Efficient Analysis of the Performance of Page Segmentation Algorithms," *Proceedings of the 5th International Conference on Document Analysis and Recognition,* pp. 451-454, 1999.

[8] T. Hochgurtel, "Stackoverflow," 26 Mayo 2017. [Online]. Available: https://stackoverflow.com/questions/9480013/image-processing-to-improve-tesseract-ocr-accuracy. [Accessed Diciembre 2017].

[9] A. Antonacopoulos and D. Bridson, "Performance Analysis Framework for Layout Analysis Methods," *7th International Conference on Document Analysis and Recognition,* pp. 1258-1262, 2007.

[10] A. Antonacopoulos and A. Coenen, "Region Description and Comparative Analysis Using a Tesseral Representation," *Proceedings of the 5th International Conference on Document Analysis,* pp. 193-196, 1999.

[11] R. Cattoni, T. Coianiz, S. Messelodi and C. Modena, "Geometric Layout Analysis Techniques for Document Image Understanding: a Review," IRST, Italy, 1998.

[12] J. Nilsson, "Quora," 21 Agosto 2017. [Online]. Available: https://www.quora.com/Which-one-is-better-for-OCR-Tesseract-or-Vuforia-SDK. [Accessed Diciembre 2017].

[13] S. Mao, A. Rosenfeld and T. Kanungo, "Document Structure Analysis Algorithms: A Literature Survey," *SPIE,* vol. 5010, pp. 197-207, 2003.

[14] H. Jianying, K. Ramanujan and W. Gordon, "Document Image Layout Comparison and Classification," *Proc. 5th Int'l Conf. Doc. Anal. Rec.,* 1999.

[15] "EMGU," 04 Noviembre 2017. [Online]. Available: http://www.emgu.com/wiki/index.php/Main_Page. [Accessed Diciembre 2017].

[16] "Github," 2018. [Online]. Available: https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality.

[17] K. IBM, "IBM Knowledge Center," 2015. [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SSEPGG_9.5.0/com.ibm.im.visual.doc/c_confusion_matrix_view.html. [Accessed Diciembre 2017].

[18] T. A. Salthouse, "Effects of age and skill in typing.," *Journal of Experimental Psychology: General,* pp. 345-371, 1984.

[19] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys,* vol. 24, pp. 377-439, 1992.

[20] R. J:, "El fin del trabajo. Nuevas tecnologías contra puestos de trabajo: el nacimiento de una nueva era," *Revista Chilena de Derecho informatico,* 1996.