



Universidad de las Ciencias Informáticas

Facultad 6

Título: Componente para la extracción de subtítulos
presentes en fotogramas de videos

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias
Informáticas

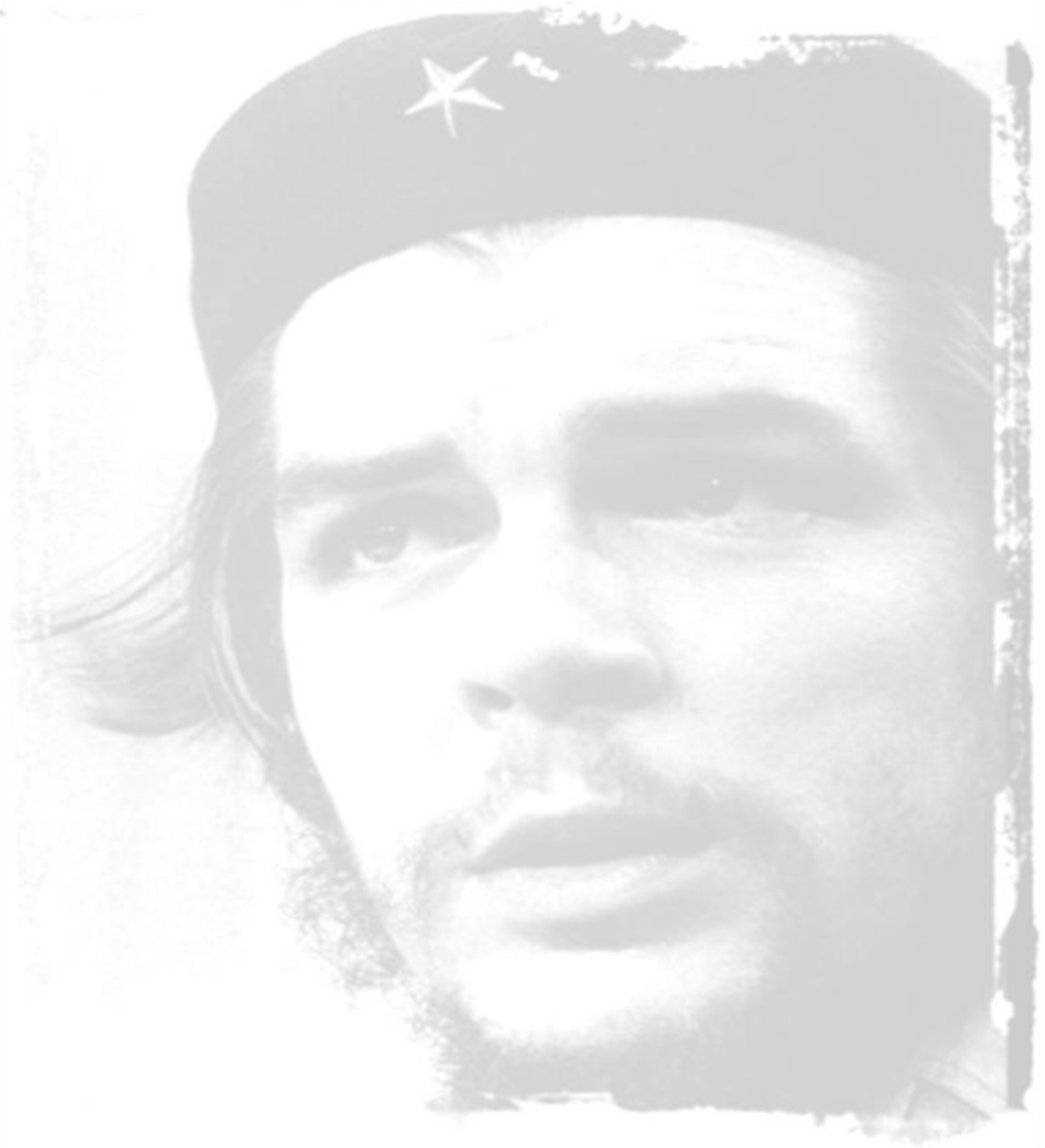
Autor: Daniel González Ibargüen

Tutor: Msc. Rafael L. Cardero Álvarez

Co-tutor: Ing. Jean M. Suárez Pérez

La Habana, Junio del 2012

“Año 54 de la Revolución”



" El néctar de la victoria se saborea en la copa del sacrificio.. "

de

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro GeySed de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Daniel González Ibargüen (autor)

Jean M. Suárez Pérez (tutor)

Datos de contacto:

Tutor: Msc. Rafael L. Cardero Álvarez.

Formación Académica: Ingeniero en Ciencias Informáticas (Julio/2008).

Máster en Ciencias Informáticas (Julio/2011).

Centro Laboral: Universidad de las Ciencias Informáticas (UCI).

Correo Electrónico: rlcardero@uci.cu

Otros Datos de Interés:

- ✓ Diploma de Oro.
- ✓ Premio Mella.
- ✓ Vinculado al Proyecto Productivo UCITeVe y PREMICIA durante sus años de estudios universitarios.
- ✓ Profesor del Departamento de Señales Digitales, Facultad 6.

Co-tutor: Ing. Jean M. Suárez Pérez.

Formación Académica: Ingeniero en Ciencias Informáticas (Julio/2009).

Centro Laboral: Universidad de las Ciencias Informáticas (UCI).

Correo Electrónico: jmsuarez@uci.cu

Otros Datos de Interés:

- ✓ Diploma de Oro.
- ✓ Premio Mella.
- ✓ Vinculado al Proyecto Productivo UCITeVe y CCM durante sus años de estudios universitarios.
- ✓ Líder de Proyecto CCM (Captura y Catalogación de Medias).
- ✓ Profesor del Departamento de Señales Digitales, Facultad 6.

Agradecimientos

A mi madre y a mi padre por ser la bujía inspiradora en mi carrera y la mayor fuente de apoyo y confianza a lo largo de toda mi vida, gracias por su cariño, su existencia y por hacerme el hombre que soy.

A mis dos abuelas que no pudieron disfrutar conmigo el placer de ser ingeniero y de sentirse importante en la sociedad, como me dijo una de ellas. Muy especialmente a mi abuela materna que fue una madre y una amiga en todo el tiempo que estuvo a mi lado y que una vez dijo que no me vería cuando fuera ingeniero, pero si estaba segura que lo lograría.

A mi tío Rigo por su apoyo, sus sabios consejos en los momentos difíciles y su ayuda a lo largo de estos años; así como a mis tíos Ricardo, Raúl y Noel y a mi tía Melba por la ayuda brindada no solo a mi sino a mis padres, para que yo pudiera realizar el tan anhelado sueño de ser ingeniero.

A mis primos David y Ricardo que han sabido ser los hermanos que no me dio la vida, así como al resto de mi familia que supo estar presente en todo momento importante de mi vida.

A mi novia Lise por su amor y su cariño; la que ha estado siempre ahí presente cuando más falta me ha hecho y que tanta ayuda me ha brindado con el trabajo de diploma, así como a su familia que me acogió como uno más de los suyos en muy poco tiempo .

A Yudiel, Alexey, Erliem y Emilio que fueron mis grandes compañeros de batalla en este tiempo de universidad y que hicieron de estos años tan especiales; así como a mis compañeros de cuarto.

A mis amigas Kely, Diana, Ivis, Lisandra, Dayani y Malena por su amistad incondicional y por su ayuda en todo este tiempo.

Al resto de mis amigos durante mi estancia en la UCI, los que han sido mi mayor orgullo en estos años de universidad, ya que me brindaron su confianza y su ayuda incondicional

desde el principio y suplieron la distancia de la familia. Muy especialmente a mi piquete de las canchas, los que fueron más que una familia para mí en todo este tiempo, de los cuales no diré nombre porque son muchos, pero si quiero darle un merecido agradecimiento.

A las amistades de mi tierra y a mis vecinos, por su apoyo en todo momento y por su constante preocupación por mí, por mi familia y por mi desempeño.

A Jean por suprimir el importante papel de tutor y ayudarme en los momentos más nebulosos de la tesis; así como a todos los profesores que han aportado un granito de arena en mi formación desde que era un niño y no pensaba convertirme en un profesional.

Al tribunal de mi tesis y a mi oponente por los consejos oportunos para lograr un mejor trabajo de diploma.

A todo un gran número de personas que se me quedan, pero que también cuentan con mi sincero agradecimiento.

A todos... Muchas Gracias por ayudarme e inspirarme a llegar hasta aquí.

Daniel González Ibargüen.

Dedicatoria

A mi madre y a mi padre por confiar en mí y a mi familia en general por su apoyo; y muy especialmente a mis dos lindas abuelas que no están físicamente a mi lado, pero si muy cerca de mi corazón.

Yo

Resumen

El surgimiento y desarrollo de las telecomunicaciones ha propiciado el uso de los contenidos multimedia, debido a los beneficios que traen en el desarrollo social. Los videos digitales son parte importante de este tipo de contenidos y producto del gran cúmulo de información que poseen, existe una necesidad actual de realizarles búsquedas por su contenido. Partiendo de ello, la presente investigación centra su atención en el subtítulo presente en forma de texto el cual contiene casi en su totalidad la información de dichos videos. El trabajo tiene como objetivo desarrollar un componente para extraer y manipular el texto en los fotogramas de un video de forma automática, para usarlo posteriormente en la búsqueda en videos por contenido. Dejando de esta forma atrás los métodos tradicionales de extracción de texto manual, los cuales en su mayoría ocasionan errores humanos y altos costos económicos, debido al gran tiempo que se emplea para ello; trayendo consigo que la eficiencia de las búsquedas y manipulación de estos textos no sea la deseada. El componente desarrollado como parte de la investigación ofrece mejores resultados ante la presencia de textos horizontales, que posean un contraste relativamente alto con el fondo de la imagen y que no presenten gran distorsión. El proceso de reconocimiento y extracción de texto se realizó con la ayuda del Software de Reconocimiento Óptico de Caracteres GOCR. Finalmente se le hicieron pruebas a la aplicación implementada, que muestran que la eficiencia de los resultados depende en gran medida del OCR empleado.

Palabras Claves:

Extracción de Texto, Fotograma, OCR.

Índice

Agradecimientos	I
Dedicatoria	III
Resumen.....	IV
Introducción.....	1
Capítulo 1: “Fundamentación Teórica del Componente para la extracción de subtítulos presente en fotogramas de videos”	6
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.2.1 Subtítulo	6
1.2.2 Fotograma.....	6
1.2.3 Metadato	7
1.2.4 Imagen Digital	8
1.3 Objeto de Estudio.....	9
1.3.1 Descripción General	9
1.3.2 Situación del dominio actual del problema.....	14
1.3.3 Situación Problemática.....	14
1.4 Análisis de soluciones existentes.....	15
1.5 Conclusiones Parciales.....	18
Capítulo 2: “Propuesta de solución del Componente para la extracción de subtítulos presente en fotogramas de videos”	19
2.1 Introducción.....	19
2.2 Lenguaje de desarrollo.....	19
2.2.1 C++.....	19
2.2.2 ¿Por qué utilizar C++?.....	20

2.3	Entorno de Desarrollo Integrado (IDE)	21
2.3.1	QT Creator	21
2.3.2	¿Por qué utilizar QT Creator?	22
2.4	Comparación entre los OCR a utilizar en el desarrollo del componente	22
3.4.1	Selección del OCR a utilizar	27
2.5	Modelo del Componente	27
2.6	Conclusiones Parciales.....	30
Capítulo 3: “Resultados obtenidos por el Componente para la extracción de subtítulos presente en fotogramas de videos”		31
3.1	Introducción.....	31
3.2	Requisitos para la integración con la Plataforma de Codificación e Indexación	31
3.3	Modelo de Diseño	32
3.3.1	Clases del diseño.....	32
3.3.2	Diagrama de clases del diseño	32
3.4	Modelo de Implementación.....	33
3.4.1	Diagrama de Componentes.....	34
3.5	Pruebas	34
3.6	Conclusiones parciales	37
Conclusiones.....		38
Recomendaciones		39
Trabajos Citados.....		40
Bibliografía		43

Introducción

Desde sus inicios, el hombre tuvo la necesidad de comunicarse para expresar sus necesidades y sentimientos, lo cual lo llevó a evolucionar en este sentido y a crear nuevos medios para ello. Esta constante búsqueda de vías de comunicación y desarrollo posibilitó el surgimiento de las Tecnologías de la Información y las Comunicaciones (TIC). Las TIC pueden definirse como el estudio, diseño, mantenimiento, desarrollo y administración de la información por medios digitales, tales como: teléfonos celulares, televisión, radio, periódicos digitales y computadoras como el medio más versátil (1).

Con el surgimiento y desarrollo de las TIC, todas las esferas: sociales, económicas y políticas, se han visto transformadas, impulsado principalmente por las innumerables ventajas que estas tecnologías brindan, tales como el acceso rápido, preciso e ininterrumpido a la información, y la automatización de casi todos los procesos. Soportados todos estos sobre redes de ordenadores para su comunicación, y teniendo como premisa el aprovechamiento de Internet.

Este convulso avance de la Informática, ha traído aparejado un inmenso desarrollo de los contenidos multimedia, entiéndase por ello: video, imagen y audio; los cuales han aumentado en la última década, motivado principalmente por el gradual crecimiento del mercado de la propaganda y la publicidad, así como por la acelerada evolución de los dispositivos físicos electrónicos (2). En Internet, y a nuestro alrededor, circula un inmenso flujo de información de este tipo, lo cual ha llevado a plantearse que puede ser muy provechoso su manejo y almacenamiento de forma eficaz.

El manejo eficiente de los metadatos¹ presentes en las multimedias, especialmente en los videos digitales, proporciona un alto número de información de gran valor y fácil de aprovechar, además, a diario su uso se va incrementando en todo el mundo. Los textos presentes en los fotogramas o secuencias de imágenes de un video digital, específicamente en forma de subtítulo, poseen casi toda la información del contenido del mismo, por lo cual se torna sumamente provechoso realizar búsquedas sobre estos. Dichas búsquedas permiten de forma clara y precisa, realizar acciones tales como: consultar citas verbales, frases habladas o nombres, brindándose así una herramienta sencilla y fácil de explotar en todos los campos de la vida humana, y que pueda adaptarse más fácil a la búsqueda de los usuarios según sus necesidades.

¹Metadatos, datos de otros datos.

A nivel mundial cadenas de televisión como CNN², BBC³ y grandes Base de Datos (BD) de medias audiovisuales como el caso de YouTube⁴, vienen planteándose desde hace un tiempo que el uso de indexadores de subtítulos y textos de cualquier tipo presentes en los distintos videos, constituye una fuente poderosa e importante de información, la cual al ser explotada de forma correcta, puede arrojar grandes beneficios a los usuarios en comodidad y rapidez. Por citar un ejemplo, el diario deportivo AS, utiliza los provechos de la extracción de texto de los videos, para guardar y buscar resultados de distintos deportes, como partidos de futbol o de tenis, así como para almacenar toda una serie de noticias que aparecen en forma de cortinilla⁵ en estos partidos, las cuales brindan información de todo tipo y de muy útil provecho (3).

Cuba, a pesar de contar con pocos recursos y poca experiencia en el manejo de los metadatos presentes en las medias audiovisuales, no está ajena a este desarrollo tecnológico. Ya se han comenzado a dar los primeros pasos para aprovechar los beneficios que trae el manejo y uso eficiente de estos datos, un ejemplo claro de ello lo constituyen las investigaciones y puestas en prácticas realizadas en CENATAV⁶, en campos como la extracción de caracteres presentes en las matrículas de los carros, tomados desde una cámara de vigilancia, así como la extracción, selección y clasificación de rasgos acústicos para el reconocimiento del locutor (4).

En la Universidad de las Ciencias Informáticas (UCI), se encuentra el Centro de Desarrollo de Geoinformática y Señales Digitales (GEySED), al cual pertenece el Departamento de Señales Digitales. Dicho departamento es el encargado de desarrollar productos genéricos, servicios y soluciones informáticas para el procesamiento de las Señales Digitales, en lo que está implícito el manejo de los archivos de medias y los metadatos presentes en estos, siendo esto de gran interés y utilidad para aquellas entidades que requieran aplicaciones de esta índole.

El proyecto Captura y Catalogación de Media (CCM), perteneciente al Departamento de Señales Digitales, se ha dado la tarea de crear un componente para almacenar y aprovechar el metadato en

² **CNN**, Cable News Network (Cadena de Noticias por Cable). Cadena de televisión estadounidense, fundada en 1980.

³ **BBC**, British Broadcasting Corporation (Corporación Británica de Radiodifusión). Cadena de televisión Británica, fundada en 1922.

⁴ **YouTube**, sitio en Internet donde subir y compartir videos de todo tipo.

⁵ **Cortinilla**, noticias que aparecen en estáticas o en movimiento en la parte inferior de la pantalla.

⁶ **CENATAV**, Centro de Aplicaciones de Tecnologías de Avanzada.

forma de texto presente en los subtítulos de los videos digitales, de forma más sencilla a través de la extracción y comparación de los mismos, ya que actualmente este proceso en casi todo el mundo se realiza de forma manual, acarreado con ello varios inconvenientes: es costoso, tedioso, necesita mucho tiempo de trabajo, además de estar sujeto a ambigüedades y errores humanos; mientras que por otra parte no se tiene un buen aprovechamiento de la información presente en estos metadatos, debido a la ausencia de una herramienta que se encargue del manejo y manipulación eficiente de estos. Por lo cual, a partir del análisis de la problemática expuesta, surge como **problema de la investigación**: ¿Cómo reconocer y comparar el texto de los subtítulos presentes en los fotogramas de video?

Dado el problema investigativo planteado, se tiene como **objeto de estudio**, los procesos de reconocimiento y extracción de los subtítulos contenidos en los fotogramas de un video digital. Delimitando de esta forma el **campo de acción** como la informatización de la extracción de los subtítulos contenidos en los fotogramas de un video digital para los productos del Departamento de Señales Digitales que lo requieran. Por lo tanto, queda definido como **objetivo general**: desarrollar un componente que reconozca y compare el texto de los subtítulos presentes en los fotogramas de los videos.

A partir de los elementos teóricos definidos anteriormente, se establece como **idea a defender** que con el desarrollo de un componente que reconozca y compare el texto de los subtítulos presentes en los fotogramas de los videos, se podrá extraer y contar con la información presente en éstos; información que constituye un metadato relevante y valioso, sobre el cual se podrán realizar diversos tipos de búsquedas.

Para dar cumplimiento al objetivo de la investigación y resolver la problemática planteada, fueron establecidas las siguientes **tareas de la investigación**:

- ✓ Analizar la bibliografía y aplicaciones actuales existentes que permitan detectar y extraer el texto contenido en videos.
- ✓ Describir el funcionamiento de los sistemas para la extracción de texto en video.
- ✓ Caracterizar las herramientas y tecnologías informáticas a usar en el desarrollo del componente propuesto como solución de la investigación.
- ✓ Seleccionar un OCR⁷ para su utilización en el componente.

⁷OCR, siglas en inglés de Optical Character Recognition (Reconocimiento Óptico de Caracteres).

- ✓ Elaborar un modelo para extraer los textos y compararlos con los ya existentes.
- ✓ Elaborar el diseño perteneciente al modelo elaborado para la implementación del componente.
- ✓ Implementar el modelo elaborado.
- ✓ Definir los requisitos para la integración con la Plataforma de Codificación e Indexación.
- ✓ Implementar la estructura de negocio asociada a las pautas identificadas en la plataforma.
- ✓ Validar la propuesta con la utilización de bases de datos heterogéneas.

Como apoyo a la investigación, se emplean los siguientes **métodos científicos**:

Métodos Teóricos:

Análisis Histórico – Lógico: Empleado en el estudio del estado del arte de los sistemas de detección y extracción de textos en videos digitales, así como de las técnicas y procesos más usados en el mundo, asociados a este tipo de herramientas.

Analítico – Sintético: Utilizado para analizar y comprender los procesos de detección y extracción de textos de videos digitales a través de las teorías y documentos que ocupan el objetivo de la investigación, extrayendo los aspectos más importantes relacionados con el objeto de estudio. Ya que el análisis de la bibliografía referente al tema, permitirá realizar una síntesis para arribar a conclusiones que puedan ser posibles soluciones del problema o que ayuden a solucionarlo.

Modelación: Usado en la creación de modelos y diagramas que reflejen la lógica del componente en su diseño e implementación. Aplicando la modelación teórica, ya que permite representar las características y relaciones fundamentales del fenómeno real, llevado al sistema.

Métodos Empíricos:

Observación: Para ver como se desenvuelven todos los fenómenos asociados a la detección y extracción de textos en videos digitales, a través de la observación por medio de registros visuales de lo que ocurre en la situación real.

Como Aporte práctico de la investigación: Se desarrollará un componente de software para la extracción automática del texto presente en forma de subtítulo en los fotogramas de un video digital. Componente el cual podrá ser utilizado por los productos audiovisuales desarrollados en el departamento o la universidad que así lo deseen.

Finalmente, dada la fundamentación teórica de la investigación, la **estructuración del contenido** de la misma para dar cumplimiento a las tareas trazadas, queda conformada de la siguiente forma:

Capítulo 1: “Fundamentación Teórica del Componente para la extracción de subtítulos presente en fotogramas de videos”.

En este capítulo se definen conceptos claves para la fácil comprensión de la investigación y se estudian algunas soluciones existentes para tener un mejor dominio del tipo de componente a desarrollar, propiciando de esta forma el conocimiento necesario para definir la fundamentación teórica de la investigación y la concesión del componente resultado de esta.

Capítulo 2: “Propuesta de solución del Componente para la extracción de subtítulos presente en fotogramas de videos”.

Este capítulo recoge las herramientas a utilizar y el porqué de la selección de estas, así como una breve comparación a raíz de los resultados arrojados por Tesseract⁸ y GOCR⁹, para poder definir que OCR a utilizar en el desarrollo del componente y por último se describe el modelo del sistema a desarrollar.

Capítulo 3: “Resultados obtenidos por el Componente para la extracción de subtítulos presente en fotogramas de videos”.

En este capítulo se definen los principales requisitos de integración del sistema con la Plataforma de Codificación e Indexación, así como se definen los diagramas de Clase del Diseño y el de Componentes del sistema. Por último se muestran las pruebas realizadas al componente para comprobar su desarrollo satisfactorio y el cumplimiento del objetivo de la presente investigación.

⁸Tesseract, Software de reconocimiento óptico de caracteres.

⁹GOCR, Software de reconocimiento óptico de caracteres.

Capítulo 1: “Fundamentación Teórica del Componente para la extracción de subtítulos presente en fotogramas de videos”

1.1 Introducción

Este capítulo aborda el estado del arte de los sistemas de detección y extracción de textos presentes en fotogramas de videos. Así como se definen una serie de conceptos asociados a estos que son de vital importancia para la comprensión y desarrollo de la investigación y la solución que se propone. Todo ello con el objetivo de definir la fundamentación teórica de la investigación.

1.2 Conceptos asociados al dominio del problema.

Para mejor comprensión de los temas tratados en el presente documento y adentrarse en el tema de la detección y extracción de textos en videos digitales, a continuación se definen una serie de conceptos identificados a lo largo de la investigación:

1.2.1 Subtítulo

Un **subtítulo** es un letrero que, al proyectarse un filme, aparece en la parte inferior de la imagen, normalmente con la versión del texto hablado de la película (5).

Un **subtítulo** es un letrero o sobreimpresión que aparece en la parte inferior de la proyección cinematográfica y que corresponde a la transcripción o traducción del texto de la versión original (6).

El **subtítulo** no es más que un conjunto alfanumérico de caracteres, que representan una versión textual del dialogo o narración que se produce en cada una de las escenas de un filme y que la mayoría de las veces aparece en la parte inferior del video y con un alto contraste con respecto al fondo. Este puede ser generado en tiempo real o posterior a la grabación y pueden aparecer superpuestos o incrustados en cada uno de los fotogramas de un video digital.

1.2.2 Fotograma

Un **fotograma** es cada una de las imágenes que se suceden en una película cinematográfica (5).

Un **fotograma** es cada una de las imágenes captadas por una cámara de video y registrada de forma digital o analógica. Al visualizarse dichas imágenes a cierta frecuencia de imágenes por segundo, se

crea la sensación de movimiento percibida por el ojo humano, esta frecuencia debe ser menor de 50 Hz¹⁰ para que el ojo humano observe dicha transacción entre imágenes (7).

La frecuencia que utilizan los sistemas más difundidos en el mundo son:

- ✓ Cine mudo: 16-18 Hz.
- ✓ Cine clásico: 24 Hz.
- ✓ Televisión, de normas Europeas (PAL & SECAN): 25 HZ.
- ✓ Televisión, de norma Estadunidense y utilizada en Cuba (NTSC): 29.97 Hz.

1.2.2.1 Fotograma Clave

Un **fotograma clave** es una marca en la línea de tiempo que, en función del estado del bloque de fotogramas clave, mantiene una pose definida por el artista sin ser modificada por las variaciones de los parámetros en fotogramas adyacentes (8).

Todas las películas con formato MPEG están formadas por recuadros de 16 x 16. Para ahorrar espacio, los recuadros que son casi idénticos a aquellos recuadros del siguiente fotograma de la película son descartados (9).

Resumiendo, un **fotograma clave** no es más que una imagen base o fija, partiendo de la cual se construyen y realizan los cambios en las demás imágenes siguientes a dicho fotograma. Basándose para ello en la similitud entre la imagen base o fotograma clave y las siguientes imágenes, descartando de esta forma los rasgos o similitudes entre una imagen y otra y tomando solo aquello que cambia.

1.2.3 Metadato

Los **metadatos** son información estructurada que describe, explica y localiza, cierta información; o hace más fácil de recuperar, utilizar o administrar una fuente de información. Metadatos se llama a menudo a datos sobre los datos o información acerca de la información (10).

Desde un punto de vista informático los **metadatos** se consideran un conjunto de reglas incluidas en las aplicaciones de manejo de información geográfica que describen la estructura interna de los esquemas de datos (11).

¹⁰Hz, Hercios: unidad de medida para la frecuencia.

En otras palabras, un **metadato** describe “datos de otros datos”. El término se refiere a cualquier elemento descriptivo de los datos, como un título en un archivo de video o un grupo de palabras claves en un artículo escrito en los campos ocultos en una página Web (12). Por ejemplo, las cámaras digitales almacenan metadatos en las imágenes que describen la configuración de la cámara para la foto, tales como fecha y hora en la que se toma una imagen y marca de la cámara. También los archivos de música digital contienen metadatos como el título de la canción, el género, nombre del artista y el nombre del disco. En la presente investigación el metadato con el cual se desea trabajar, es el presente en los subtítulos de los videos, en forma de texto, los cuales brinda casi toda la información de cada una de las escenas.

1.2.3.1 Extracción y almacenamiento de metadatos en forma de subtítulo.

La **extracción y almacenamiento de metadatos en forma de subtítulo**, es aquella donde se identifica el texto plano presente en forma de subtítulo, en cada una de las imágenes de un video digital y se extrae haciendo uso de un componente de reconocimiento óptico de caracteres (15). Dicho componente identifica automáticamente a partir de una imagen, caracteres o símbolos que pertenecen a un determinado alfabeto alfanumérico, con los que se puede interactuar mediante un programa de edición de texto o similar para luego poder ser analizado semánticamente e indexado en un lugar específico como: una base de datos. Los subtítulos presentes en los videos digitales pueden ser almacenados en un fichero final como texto plano o asociándose con la secuencia de la imagen donde aparecen (16), posibilitando de esta forma distintos criterios para su almacenamiento y futura manipulación, como pueden ser: momento (hora, minuto, segundo) en que aparece el texto en el video, cantidad de ocurrencias de ciertos diálogos, palabras, nombres o cifras, entre otros.

1.2.4 Imagen Digital

Una **Imagen Digital**¹¹ es cualquier imagen fija o en movimiento, que se capture en un medio electrónico y que se represente como un archivo de información leído como una serie de pulsos eléctricos (13).

Una **Imagen Digital** es un arreglo bidimensional de píxeles¹². El valor de cada píxel se representa mediante una función f , donde $f(x,y)$ representa el nivel de brillantez, color o intensidad de la imagen en

¹¹ **Digital**, es toda información representada por una serie de pulsos eléctricos basados en un sistema binario (ceros y unos).

¹² **Pixel**, es la abreviatura de la expresión inglesa Picture Element (Elemento de Imagen). Es la menor unidad que compone una imagen en una computadora o cualquier medio electrónico, ésta a su vez, está compuesta por mapas de bits.

tales coordenadas. Por lo tanto, una imagen en blanco y negro puede ser representada por una matriz de dimensión $m \times n$, donde $f(x,y) \in \{0,1\}$. Además de la representación en blanco y negro, otras opciones son las imágenes en tonos grises, para las cuales $f(x,y)$ representa un nivel de intensidad típicamente asociando a un entero entre 0 y 255 y otras más sirven para describir imágenes en color (14).

En general, una **Imagen Digital** no es más que una representación bidimensional ($M \times N$), de la captura de una imagen fija o en movimiento, a través de un medio electrónico, donde ésta se representa como una serie de pulsos eléctricos.

1.3 Objeto de Estudio

Para el buen desempeño de la investigación y el desarrollo de la solución informática es necesario estudiar los procesos de detección y extracción de los textos contenidos en los fotogramas de un video digital.

1.3.1 Descripción General

Debido al constante uso de las medias audiovisuales en la actualidad, se evidencia cada día más el aumento y desarrollo de las mismas en todas las esferas, por lo cual es necesario emplear diversas técnicas y tecnologías que permitan realizar búsquedas eficientes sobre la información que estas poseen. Se puede apreciar que una de las formas más fáciles de realizar dicha búsqueda, es a través de la extracción de los textos en forma de subtítulo que aparecen en las imágenes de un video digital.

En los subtítulos se puede encontrar, casi en su totalidad, la información del contenido del video que se muestra, ya sea cualquier ente que se esté proyectando: un diálogo, una narración o una descripción escrita de algo en específico. Con esto se evidencia la necesidad de utilizar técnicas que permitan detectar y extraer la información que está contenida en ellos, para su posterior utilización.

Los métodos para la detección y extracción del subtítulo contenido en las distintas imágenes de un video digital, tienen como estrategia una serie de pasos, los cuales parten de resaltar las regiones más relevantes de la imagen, haciendo más sencilla la localización del texto en la misma, para luego realizar una binarización de éstas como parte del pre-procesamiento del OCR seleccionado. Una vez obtenida una imagen clara y limpia, es posible detectar la información contenida en ella, y haciendo uso de un OCR, tomar el texto presente en los subtítulos para su posterior manejo en procesos como el análisis semántico y sintáctico de las oraciones que se encuentran en ellos. Luego de realizada la secuencia de procesos antes descritos, el OCR devuelve el texto extraído en ASCII o Unicode, para así poderlo llevar a texto plano(17).

1.3.1.1 Sistemas OCR

Los sistemas OCR son software de reconocimiento automático de texto. Esta tecnología trata de emular el funcionamiento del ojo humano, para extraer de una imagen, el texto presente en éstas, de forma tal que sea comprensible para el ordenador (23). Los mismos trabajan sobre un alfabeto alfanumérico, que identifica automáticamente caracteres o símbolos pertenecientes a dicho alfabeto y lo convierte en cadenas de caracteres (ya sea en ASCII o Unicode) para luego poder interactuar con este por medio de un editor de texto, ya sea Adobe Acróbata, Word, Notepad (18).



Figura 1. Funcionamiento de un OCR (23).

La mayoría de los OCR, el procedimiento que utilizan es el siguiente:

El procedimiento usado en los inicios de estos sistemas consistía en que el OCR aislaba la imagen correspondiente a un carácter y la comparaba con una base de caracteres para determinar su correspondiente código ASCII o bien Unicode. Una vez reconocido dicho carácter este era transformado y trasladado al documento OCR resultante. Dicho proceso tiene el inconveniente de que trabajaba con un número limitado de fuentes, por lo que el resultado obtenido no siempre es el deseado. Además, en textos muy largos y con inclusión de frases en cursiva, este proceso se torna sumamente lento.

Los sistemas actuales de OCR están basados en el análisis de características de los caracteres, en lugar de la coincidencia de las matrices de estos, lo que permite una mayor velocidad en el proceso y el no tener que depender de una limitada base de fuentes. Dando la posibilidad de esta forma que el programa puede formular distintas hipótesis y cotejarlas con los diccionarios contenidos por el mismo (actualmente los programas contienen diccionarios en distintos idiomas), para formar palabras y textos completos.

Vale destacar que estos sistemas no tienen 100% de efectividad, está demostrado tanto en software privativos como OmniPage, Abbyy Fine Reader o READiris; así como en los libres o de código abierto como Tesseract, GOCR y Ocropus. De acuerdo con algunos estudios de la exactitud de los productos de OCR comerciales, estas varían desde setenta por ciento hasta noventa y ocho por ciento y una exactitud total sólo puede lograrse con la ayuda de la revisión humana (22).

Todo sistema OCR sigue una serie de etapas para realizar la detección de caracteres, en cualquier imagen, dentro de las cuales se pueden distinguir al menos cuatro etapas fundamentales, aplicadas en su mayoría en dichos algoritmos, una a continuación de otra, las mismas son:

➤ **Pre-proceso:**

En esta fase de pre-proceso (o adecuación de la imagen) el objetivo que se persigue es eliminar de la imagen cualquier tipo de ruido o imperfección que no pertenezca al carácter, así como normalizar el tamaño del mismo.

La normalización de la imagen también puede implicar, en la mayoría de los casos, un binarizado de la misma, el cual consiste en la conversión a blanco y negro de las imágenes que están en escala de grises o a color. Este proceso se realiza con la ayuda de un método umbral, para decidir qué píxeles conforman los objetos que buscamos (texto) y convertirlos a píxeles blancos y qué píxeles son sólo el entorno de estos objetos (fondo o resto de la imagen) y convertirlos a píxeles negros. Después de realizada la binarización se podrá obtener una imagen en blanco y negro, donde se identifique fácilmente los contornos de los caracteres y símbolos presentes en dicha imagen, posibilitando de esta forma, extraer la parte de la imagen que posee texto (18).

Para la eliminación del ruido que puede aparecer en una imagen digital, bien provocado por manchas reales, gráficas imperfectas, o por defectos técnicos en la adquisición o binarizado de la imagen, se utilizan diversos algoritmos, dentro de los cuales vale destacar los siguientes (20):

- **Etiquetado:** para la división de la imagen en regiones de componentes conectadas.
- **Erosión o expansión:** para la eliminación de pequeños grupos de píxeles.
- **Umbralizado de histograma:** para eliminar o seleccionar los objetos más brillantes o más oscuros de la imagen.

➤ **Segmentación de la imagen:**

El objetivo de la segmentación es simplificar y/o cambiar la representación de la imagen en otra más significativa y más fácil de analizar (18). La segmentación se usa tanto para localizar objetos, como para encontrar los límites de estos dentro de una imagen. Más exactamente, la segmentación de la imagen se realiza a través de la asignación de una etiqueta a cada píxel de la misma, de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares. Aquí las técnicas más usadas son: el agrupamiento por clustering (a través del método de las K-medias), los métodos basados en histogramas y por último, los métodos basados en bordes (19).

➤ **Adelgazamiento de la imagen:**

Este proceso se hace principalmente a través de los modelos de estructuras vecinas y complejos celulares. El mismo consiste en ir borrando sucesivamente los puntos de los contornos de cada componente, de forma que se conserve su tipología. Se debe borrar sucesivamente los píxeles no significativos de la imagen (esta operación se realizará mientras que se elimine al menos un píxel) siempre que esto no provoque una distorsión o pérdida de la proporción de la imagen original, para que al final quede el esqueleto o eje medio de los píxeles significativos (16).

➤ **Comparación con patrones:**

Las aplicaciones de reconocimiento de patrones en forma general requieren una medida que permita discriminar entre dos conjuntos de características. La capacidad de discriminación de esta medida es directamente proporcional a la eficiencia o capacidad de reconocimiento de estas aplicaciones. Algunas de las medidas más conocidas para realizar esta discriminación incluyen la correlación, el error cuadrático medio, la distancia euclidiana, medidas estadísticas tal como: la media, la varianza o la desviación estándar; así como también se utilizan las técnicas de inteligencia artificial, como es el caso de las redes neuronales y los sistemas basados en lógica difusa. Los patrones se obtienen a partir de los procesos de segmentación, extracción de características y descripción donde cada objeto queda representado por una colección de descriptores. El sistema de reconocimiento debe asignar a cada objeto su categoría o clase (conjunto de entidades que comparten alguna característica que las diferencia del resto). Para poder reconocer los patrones se siguen los siguientes procesos: adquisición de datos, extracción de características y toma de decisiones. Este proceso se hace con la finalidad de comparar los caracteres a reconocer con el alfabeto previamente definido, hasta encontrar la máxima coincidencia posible (21).

Ventajas de los OCR (23):

➤ **Búsqueda y recuperación de documentos**

1. La aplicación del OCR permite realizar búsquedas de texto libre sobre la totalidad del documento, posibilitando realizar acciones tales como: búsquedas por palabras claves y citas textuales.
2. En el proceso de creación de los metadatos, el OCR se puede utilizar para generar índices de palabras clave del texto reconocido de forma automática.

➤ **Explotación de los documentos**

3. El OCR permite convertir el texto presente en las imágenes digitales, a texto en formatos editables, permitiendo así poder editar y utilizar el texto a conveniencia del usuario.
4. Aunque el OCR no es una herramienta para hacer los documentos accesibles para personas con discapacidades visuales, su aplicación combinada con otras tecnologías permite que el texto resultante se sintetice en líneas de braille o archivos de audio.

➤ **Perspectiva económica**

5. Ahorro de tiempo respecto a la inserción manual de datos (el OCR puede alcanzar una velocidad de lectura de hasta 1.200 caracteres por segundo) y dejando atrás el costoso y tedioso proceso de extracción manual.
6. El almacenamiento en formato de texto puede suponer un ahorro de espacio respecto del almacenamiento como imagen (el archivo de texto necesita aproximadamente 1/3 del espacio que ocupa la imagen).

Desventajas de los OCR:

1. Carencia de conocimiento y expertos en las instituciones, debido a que es un campo de la informática poco utilizado y del que existe un gran desconocimiento a nivel mundial.
2. Elevado costo de generar texto electrónico (no confundir con imagen digital) este proceso puede realizarse tecleando el texto (forma manual) o a través de OCR (forma manual) y posterior revisión y corrección del texto. Acciones para las cuales se utilizan un cierto número de recursos informáticos, tanto físicos como de software.
3. Nivel de efectividad insatisfactorio del OCR en el reconocimiento de texto en imágenes borrosas, deterioradas, solapadas o con ruido, aunque este proceso puede mejorarse asta cierta medida con un pre-procesamiento de las imágenes.

En la concesión de la herramienta resultado de la presente investigación, luego de realizados todos los anteriores procesos descritos en este epígrafe, se procederá a realizar el almacenamiento del texto

contenido en el subtítulo presente en los fotogramas, cada uno de estos será almacenado con su tiempo de aparición inicial en el video, con el fin de facilitar la búsqueda a los usuarios dentro del fichero del texto extraído, posibilitando así tener una idea de donde se comienza a hablar de un tema específico.

1.3.2 Situación del dominio actual del problema.

En la concesión del proyecto CCM, se planteó que la extracción de audio de los videos podría ser un puntal clave para la comodidad y beneficio de los usuarios a la hora de realizar búsquedas. En la actualidad, dicha funcionalidad no está a disposición y teniendo en cuenta la similitud de los metadatos, se concibió la idea de la creación de un componente para indexar los subtítulos presentes en los fotogramas de los videos, para aprovechar de esta forma dicha información. Aportando ello una vía más para enriquecer las distintas búsquedas que se realizan actualmente en el proyecto, como es el caso de búsquedas de palabras claves, nombres o citas textuales sobre texto plano. Enriqueciéndose de esta manera las vías y medios a disposición de los usuarios y del personal del departamento, para la búsqueda de información.

Debido a que las búsquedas sobre el contenido de los videos, no se realizan empleando los métodos más idóneos para facilitar y agilizar el proceso, el proyecto se planteó la tarea de crear un Componente para la extracción de los subtítulos presentes en los fotogramas de videos.

1.3.3 Situación Problemática.

El rápido desarrollo en el campo de la informática, ha impulsado a buscar nuevas alternativas para el tratamiento de la información, que esta va dando lugar. Como ya se planteaba, el manejo de los textos presentes en los subtítulos, se presenta como un fuerte candidato para la manipulación del gran flujo de información que brindan los videos digitales. Pero el trabajo con estos textos acarrea un amplio número de problemas que de no ser resueltos, imposibilita en gran medida el aprovechamiento eficiente del contenido de ellos, para realizar procesos tales como las búsquedas sobre estos.

La extracción del texto presente en los subtítulos, en casi todo el mundo, se realiza de forma manual, lo cual trae aparejado un gran número de deficiencias, las que se esperan, sean eliminadas con la automatización de este proceso. Entre las deficiencias más comunes se encuentran que:

- ✓ Se cometen errores humanos a la hora de interpretar y redactar el texto extraído de los subtítulos.

- ✓ Se necesita mucho tiempo de trabajo, debido a la cantidad de información que se necesita extraer del video, para poderla pasar a texto plano.
- ✓ El proceso se vuelve sumamente tedioso, por el tiempo a emplear para ello y la cantidad de información a procesar.
- ✓ Este proceso es costoso, por la cantidad de información que se debe obtener, por el tiempo a emplear para ello y por la cantidad de recursos humanos y electrónicos utilizados para extraer los subtítulos.
- ✓ Puede estar sujeto a ambigüedades propias del análisis de las escenas, la traducción o la redacción.

Por otra parte no se puede explotar en su totalidad la valiosa información que proporcionan los videos, debido a que no existe en el departamento una herramienta que se encargue de extraer tanto el texto, como el audio presente en estos; debido a que CCM en un primer momento concibió el trabajo con el metadato audio como una de las líneas de investigación, partiendo de la idea de que la extracción y manipulación de este, como información almacenada, traería altos beneficios en comodidad y rapidez a los usuarios, pero dicha información es muy difícil de manipular y se abortó el trabajo con esta . Lo cual da cabida al mal aprovechamiento de los recursos existentes, refiriéndonos a la mala manipulación de la información presente en los videos, especialmente a la que está en forma de audio y subtítulo. Trayendo con ello que la búsqueda de información se torna a veces sumamente lenta y tediosa, por no contar con un buscador o una referencia concreta hacia la información almacenada en las medias e imposibilitando así obtener en su totalidad la información que se desea. Propiciando todo lo anterior la creación de un componente para extraer los subtítulos presente en los fotogramas de video, debido a que los mismos poseen casi en su totalidad la información presente en dichas medias.

Por todo lo anterior expuesto, y partiendo de la idea de que este tipo de herramientas constituye un mecanismo eficaz de búsqueda, análisis y control de acontecimientos: políticos, sociales, históricos y deportivos, se tomó la decisión de llevar a cabo su desarrollo.

1.4 Análisis de soluciones existentes.

En la actualidad se pueden encontrar un gran número de soluciones para la extracción de texto, las cuales extraen con un alto por ciento de certeza, el texto presente tanto en imágenes, como en documentos, pero la mayoría presenta como limitante que no permite el trabajo con secuencias de

imágenes, y menos aún el indexado de esta información por criterios definidos. Por otra parte, generalmente estos software son propietarios, o sea, a veces se requiere pagar altos costos de licencias para su uso y en su mayoría no brindan acceso a su código. Este problema, imposibilita que estas aplicaciones se puedan estudiar, mejorar o cambiar, ya sea para agregarle otras funcionalidades que sean necesarias, o para modificarle alguna que ya tenga, sin mencionar la implicación económica que representa para el país, lo cual no se ajusta a las políticas actuales del mismo, donde es necesario lograr obtener soluciones de calidad con el mayor ahorro posible de recursos.

Como objetivo de la presente investigación, se persigue desarrollar una herramienta la cual utilice software libre para su concesión, por lo que las soluciones que se estudiaron son solo de este tipo. A continuación se muestran algunos de los OCR de código abierto que brindan respuesta de manera parcial o total al problema científico planteado:

➤ **Ocrad**

Es un software OCR desarrollado como parte del proyecto GNU Linux, que al igual que todo el software desarrollado para GNU es de código abierto y consta con una licencia GNU GPL. Este lee imágenes en pbm (mapa de bits), los formatos PGM (escala de grises) o ppm (color) y produce un texto en byte (8-bit) o UTF-8. También incluye un analizador de diseño capaz de separar las columnas o bloques de texto que normalmente se encuentran en las páginas impresas. Ocrad también se puede utilizar como una aplicación de consola independiente, o como un servidor para otros programas (back- end)¹³(24).

➤ **OCRopus**

El mismo fue llevado a la luz en diciembre del 2007 bajo licencia Apache 2.0, esto significa que usted puede usar OCRopus en un producto comercial sin liberar el código fuente; y tiene como su principal desarrollador a Thomas Breuel, programador del Centro de Investigación Alemán para la Inteligencia Artificial. Este OCR está patrocinado por Google y actualmente solamente está disponible para Linux, su principal punto a explotar es la conversión en grandes cantidades de documentos. OCRopus trabaja con la inclusión de Tesseract como motor OCR y realiza todo el pre-proceso de la imagen y el análisis de la misma, así como trabaja encima del documento explorado antes de pasarlo a Tesseract para el

¹³**Back-End**, término en inglés que se utiliza para este tipo de sistemas, los cuales permiten la iteración como servidor a otros sistemas.

reconocimiento línea por línea o de carácter-por-carácter. Actualmente se puede utilizar solamente desde la interfaz de líneas de comando o a través de gscan2 (25).

➤ **Tesseract**

Este es uno de los software libre de reconocimiento de caracteres más precisos y que goza de mejor confianza en la actualidad. El mismo fue creado por Hewlett-Packard y fue distribuido como software privativo de esta empresa entre 1985y 1995, pero debido al poco avance que tuvo en la década siguiente fue lanzado al mercado como software de código abierto en 2005, obteniendo así en 2006 un patrocinio importante de Google quien actualmente es el principal desarrollador (26).Tesseract en combinación con la librería de procesamiento de imágenes Letonica¹⁴puede leer una amplia variedad de formatos de imagen y reconocer el texto presente en estas en más de 40 idiomas, gracias al alto número de diccionarios con que cuenta este OCR. En la actualidad se distribuye bajo la licencia Apache 2.0, esto significa que usted puede usar Tesseract en un producto comercial sin liberar el código fuente. Este OCR puede ser usado en Linux, en Windows (usando Visual C++ Express 2008/2010) y en Mac OS X, pero solo es probado rigurosamente por los desarrolladores para los sistemas operativos Ubuntu y Windows. El mismo no cuenta con una interfaz de usuario, la interacción con este se realiza a través de la interfaz de líneas de comando, aunque existen varios software que permiten la interacción con este a través de una interfaz gráfica como lo son: OCRFeeder, FreeOCR, Tesseract GUI y TextRipper (27).

➤ **GOCR**

GOCR fue creado por Joerg Schulenburgel, el cual dirige en la actualidad el equipo encargado del desarrollo de este software, el mismo fue concebido para el proyecto GNU Linux y tiene una licencia GNU GPL. Este fue lanzado en mayo de 1997 y permite el trabajo con distintos formatos de imágenes como PNM, PBM, PGM, JPG, y BMP, el mismo posibilita la extracción del texto presente en las imágenes con estos formatos y convertirlos en archivos de texto. La interacción con el mismo puede realizarse a través de la interfaz de comando o de la interfaz gráficagocr.tcl, así como puede ser utilizado como Back-End a otros programas y como motor de OCR en distintas aplicaciones como: FreeOCR y OCRFeeder (28).

¹⁴**Leptonica**, librería de código abierto que es ampliamente útil para el procesamiento de imágenes y aplicaciones de análisis de imagen.

1.5 Conclusiones Parciales

Al termino del presente capítulo se concluye que es necesario realizar la selección de uno de los OCR anteriormente descrito, con el fin de utilizarlo para la extracción del texto presente en los subtítulos de los fotogramas de videos, lo cual constituiría una notable ayuda para el aprovechamiento eficiente del gran cúmulo de información que está presente en los subtítulos de los videos digitales. Teniendo siempre en cuenta que los OCR que existen actualmente a nivel mundial, en su gran mayoría, no cuentan con funcionalidades propias para el trabajo con texto presente en fotogramas de videos, lo cual propicia una insuficiente eficiencia para la extracción de textos presentes en imágenes de video, lo cual limita el proceso de búsqueda y almacenado de la información presente en estos. Por lo cual luego de seleccionado el OCR a utilizar como parte de la estructura del componente resultado de la presente investigación, se debe definir un pos-procesamiento del texto extraído por este, con el fin de mejorar la calidad del mismo.

Capítulo 2: “Propuesta de solución del Componente para la extracción de subtítulos presente en fotogramas de videos”

2.1 Introducción

Este capítulo recoge lo referido a las tecnologías y herramientas a utilizar en el desarrollo del componente para la extracción de subtítulos presentes en fotogramas de videos. También se realizan una serie de pruebas a los OCRs, para definir de acuerdo al resultado arrojado a partir de un análisis previo, cuál de ellos utilizar en la implementación del componente. Por último se define el modelo del sistema el cual constituirá la base de la implementación, y se describe cada uno de los pasos por los cuales debe pasar la imagen y posteriormente el texto extraído hasta su almacenamiento final.

2.2 Lenguaje de desarrollo

Un lenguaje de desarrollo o programación es aquel elemento dentro de la Informática, que nos permite crear programas usando instrucciones y operadores que se rigen por reglas. (29) O sea, es el lenguaje que emplean los desarrolladores para que la computadora realice las acciones que se desean. Actualmente existen diversos tipos de lenguajes de desarrollo que pueden clasificarse tanto en Alto o Bajo nivel, como en Estructurado u Orientado a Objetos, o con fines Web o Desktop. Algunos de estos lenguajes son PHP, C++, Java y C#

2.2.1 C++

C++ fue creado en la década de 1980 por Bjarne Stroustrup en los Laboratorios Bell en Murray Hill, New Jersey. Inicialmente recibió la denominación de C con clase, hasta que en 1983 se le dio el nombre de C++. Desde 1980 C++ ha sufrido cuatro importantes revisiones, una en 1985, otra en 1990, en 1998 y la última que se tiene constancia en 2006.

Es un lenguaje híbrido basado en C, que se puede compilar y su principal característica es el soporte para programación orientada a objetos (abstracción, encapsulación y polimorfismo), pero añade otras propiedades como el soporte de plantillas o programación genérica (templates), la posibilidad de redefinir operadores y la identificación de tipos en tiempo de ejecución. Una de las particularidades del C++ es la sobrecarga de operadores. La sintaxis de clases y objetos permite manipular diversas estructuras de datos y operaciones; las excepciones permiten procesar de un modo claro los casos de errores, elevando de esta forma su expresividad (30).

C++ es un lenguaje popular, utilizado actualmente para resolver diferentes tipos de problemas desde los más simples hasta los más complejos, su código puede ser compilado en varias plataformas. Incorpora una nueva característica que permite aumentar la eficiencia en cuanto a la llamada de funciones, dicha característica es la posibilidad de escribir funciones en línea (inline); inline o en línea es una petición al compilador para sustituir la llamada a la función directamente por su código, es decir, genera un código en línea. Este es el lenguaje de programación utilizado para la creación de la mayoría de los plugin¹⁵ que presenta el Sistema de Captura y Catalogación de Media, por lo cual se hace necesario su utilización para la integración del componente para extracción de subtítulos presentes en fotogramas de videos, con dicho sistema.

2.2.2 ¿Por qué utilizar C++?

Algunas de las características que presenta C++ que se tuvieron en cuenta para su utilización en el desarrollo del componente son las siguientes:

- ✓ El mismo sigue la línea de desarrollo del proyecto, ya que en su mayoría los componentes desarrollados en este, están realizados mediante la utilización de este lenguaje.
- ✓ C++ posee un gran número de librerías que evitan la re-implementación de operaciones comunes.
- ✓ La posibilidad que brinda C++ de orientar la programación a objetos, característica que le permite al programador diseñar aplicaciones desde un punto de vista más cercano a la vida real; permitiendo también la reutilización del código de una forma más lógica y productiva.
- ✓ Permite desarrollar herramientas distribuibles sin la necesidad de pagar altos costos de licencias.
- ✓ Herramientas: Existe una gran cantidad de compiladores para C++ en los distintos sistemas operativos y en todos los casos existe más de un ejemplar con licencia libre.
- ✓ Portabilidad: Es un lenguaje multiplataforma sus aplicaciones solamente tienen que ser recompiladas nuevamente para adaptarse al sistema operativo, a diferencia de otros lenguajes como Java que necesita de una máquina virtual, lo que hace el proceso mucho más lento. El

¹⁵ **Plugin**, es un complemento o una aplicación que se relaciona con otra para aportarle una característica o un servicio específico.

lenguaje está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.

- ✓ Eficiencia: Es uno de los lenguajes más rápidos en cuanto a ejecución.

2.3 Entorno de Desarrollo Integrado (IDE)

Un IDE es un software que sirve de base para la implementación de una aplicación de forma visual, sencilla y práctica. Los IDEs por lo general contienen un editor de código, un compilador, un depurador y un constructor de interfaces gráficas. Los más modernos IDEs incluyen un navegador de clases, un inspector de objetos y se integran con sistemas de control de versiones. Los mismos están hechos para soportar uno o más lenguajes de programación. (30) En otras palabras, es un programa compuesto por un conjunto de herramientas que facilitan y agilizan al programador el desarrollo de un software.

2.3.1 QT Creator

Es un IDE para la realización de aplicaciones con las bibliotecas Qt, creado por la compañía Noruega Trolltech (actualmente QT Software). En marzo del 2009 liberó su primera versión estable, teniendo gran aceptación, esta versión posee la licencia GPL permitiendo el desarrollo de aplicaciones con fines comerciales. Puede ser ejecutado en los sistemas operativos desde Windows 98 hasta Windows XP y Seven, requiriendo el compilador MinGW incluido en la instalación del IDE, Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado, además hay una versión para Linux con gcc¹⁶ 3.3, Solaris, HP-UX, Mac OS X 10.4 o superior, requiriendo Qt 4.x; puede ser soportado por otras versiones de Unix con X11(X- sistemas de ventanas para dotar a los sistemas Unix de interfaz gráfica, 11-versión 11)(31).

Entre las posibilidades que brinda Qt Creator están la del trabajo con hilos independientemente del sistema operativo, así como la posibilidad de trabajar con un modulo creado para la manipulación de los protocolos de red. Este también posee soporte para aplicaciones orientadas a componentes y permite el trabajo con los gestores de bases de datos más conocidos como MySQL y PostgreSQL, además de permitir implementar soporte para otros. Estas características hacen de Qt un IDE altamente aplicable a cualquier aplicación de escritorio sin tener que utilizar librerías externas, lo cual constituye una de las principales características por las cuales se decidió su utilización en la implementación del componente resultado de la investigación. Hasta el momento ha sido liberado bajo

¹⁶ **Gcc**, es un comando usado para ejecutar el compilador de C.

dos licencias, la LGPL y la comercial; para el desarrollo de la presente solución se ha decidido trabajar con la versión LGPL10 de Qt puesto que provee un mayor intercambio con la comunidad de clientes y desarrolladores además de que permite su utilización sin tener que pagar licencia de desarrollo.

2.3.2 ¿Por qué utilizar QT Creator?

Se decidió utilizar Qt Creator para el desarrollo del sistema ya que da soporte al lenguaje de programación C++ (seleccionado como lenguaje de desarrollo del componente) y presenta un alto rendimiento con este; y por qué es necesaria su utilización para la integración del componente con el Sistema de Captura y Catalogación de Media, debido a que es el IDE utilizado por la mayoría de las aplicaciones y plugin que presenta dicho sistema. El mismo ofrece gran compatibilidad con varios sistemas operativos, así como posee riquezas en sus funciones y librerías. Este también tiene la característica de presentar el código fuente disponible, así como cuenta con gran documentación y una alta calidad en cuanto a soporte técnico; además posee una comunidad de desarrollo que le da mantenimiento y proporciona una gran bibliografía y ayuda.

2.4 Comparación entre los OCR a utilizar en el desarrollo del componente

Luego de investigar y analizar los sistemas con los cuales se podría dar respuesta al problema planteado en la investigación y centrado principalmente en los OCR no privativos, se realizó un estudio de los OCRs que cumplieran con las métricas definidas para su utilización en el desarrollo del componente, las cuales fueron:

- ✓ Utilizar para el desarrollo del componente un software no propietario.
- ✓ Utilizar un OCR que tuviese una eficiencia que superara el 80 % de las letras y los espacios encontrados respecto al texto presente en la imagen.
- ✓ Que no fuese muy notable el cambio de la eficiencia del OCR ante imágenes con problemas, como letras opacas, fondo muy similar al color de la letra o ruido en la imagen.

Al final, el estudio dio como resultado que los dos software que mejor cumplieran los parámetros definidos en las métricas, y que mejores resultados arrojaron, fueron GOCR y Tesseract, siempre teniendo en cuenta que su eficiencia es muy variable ante distintos casos, como letras muy pequeñas o con muy poco espacio entre ellas, además, en la mayoría de los casos, su eficiencia estuvo por debajo del 80% definido en las métricas para su selección. Los restantes OCR estudiados, los cuales fueron OCRopus y Ocrad, tuvieron resultados muy alejados de la realidad, o en la mayoría de los casos no extraían el texto presente en las imágenes, por lo que se tomó la decisión de estudiar y

probar más a fondo a Tesseract y GOCR, ya que resultaron los OCR más eficientes dentro de los cuatro estudiados.

Las Tablas 2 y 3 recogen los resultados devueltos por estos dos OCR, o sea, el texto plano extraído de las imágenes, después del procesamiento de diez de las mismas, pudiéndose apreciar claramente que la eficiencia de los dos OCR, no es la esperada. También se realizó un estudio del procesamiento de dichos OCR con imágenes de distintos tipos, y con dichas imágenes en estado natural (Tabla 1), o sea, sin ningún procesamiento previo, y las mismas imágenes con un pre-procesamiento (Tabla 2) de binarización, o llevado de la imagen natural a una imagen en escala de grises, con el objetivo de buscar una mejor eficiencia en el procesamiento de dichos OCR. En la Tabla 1 se ejemplifican diez de las imágenes a las cuales se le realizó el proceso de extracción de texto y de pre-procesamiento de las mismas:

No. Imagen	Imagen Natural	Imagen Binaria	Imagen en Escala de Grises
Imagen 1			
Imagen 2			
Imagen 3			
Imagen 4			
Imagen 5			

Imagen 6			
Imagen 7			
Imagen 8			
Imagen 9			
Imagen 10			

Tabla 1. Representación de algunas de las imágenes con las cuales se probó el trabajo de los OCR.

IMAGENES	Resultado de TESSERACT para imagenes a Color	Resultado de GOCR para imagen a Color
Imagen 1	para hacerles pensar y mirar en l`zN: irección equivocada	0 _ _ _____ _____B_0_0_
Imagen 2	Llevas rézdn .deberiatirar lxs dados	Lleva_g_r_zõn 7deber_a t_rar los dados
Imagen 3	sasxm\$§§ le yas a uespenar. qs vas a mm.	Si saltas- no te vas a desperta_. 'Te vas a mo1ir.
Imagen 4	- VoyF>1t`e:51r una M. P. - Espera, gque es una MP.?feet	- voy Ta t_emmer una m. _. - Espera, ique_ es una M.P.?feet
Imagen 5	rmesppqmxcgiiygi	_____ __?__O?_?_?_? _____u_____0_____J

Imagen 6	,P\$rezc0 gelatina. Mi@g1i gran pellejería.	_ ___? ___/___ _a_o__d__ol___ _O___??_
Imagen 7	Tenemcs que enfrznlar al ejércim del infizmc	Tenemo5 que enfrenlar al ejército del in_e_no
Imagen 8	'ǻ'uǻ• amen! un. Gaunnyskum L1. rm mu	0_0 __ _u_0 __ _0_0_0__ ___0i_0_0_8__0
Imagen 9	.:1.:{ .4 .,45*.ib llevihaunmqprde mis. Deiéinslgl	_____e0 ____' _____ _r.____. _____ 0_____'0_ _ _'i..
Imagen 10	guedocqgercua]quiersiib Ei; name 10 were? \$1.	Pued0 coger cualquier sitio Si nadie la quiere? Si.

Tabla 2. Resultados arrojados por Tesseract y GOOCR con las imágenes a color o en estado natural.

IMAGENES	Resultado de TESSERACT con Imagen Binarizada	Resultado de GOOCR con Imagen Binarizada	Resultado de TESSERACT para Imagen en Escala de Grises	Resultado de GOOCR para Imagen en Escala de Grises
Imagen 1	pafaha€er]c=: pénsar y mirar en' adireckzién — lequiv0ca'da'	Par_h_cerles pensar y mirar en la direcciòn €quivocada	para hacerles pensar y m rarcn ?7. recc On equivocada	_____ _____ _B_0__0_
Imagen 2	Llsvasrézondebsriatn rar los dados	Llevas razon deber_at_rar los dados	Llsvasrézbn .deberia tirar los dadosy	Lleva__razo. n .,, e eria tirar os a os

Imagen 3	T2 vas a rnurxr	S_ sartas no te vas a despertar. Te vas a m0rir.	Si sallaghcle vas a dæ\$!9v\ar. {6 vas a mcrir.	Si saltas- no te vas a desperta_ _e vas a mo1ir.
Imagen 4	- VoyF1t`é:51er una M. P. - Espera, cqug as una M.P.'? feet	- voy'Tat_emmer una m. _. - Espera, ique_ es una M.P.?feet	- VoyFæ1t`éE1er una M. P. - Espera, gqué as una MP.?feet	- voy Ta t_emmer una m. _. - Espera, ique_ es una M.P.?feet
Imagen 5	No mm mmm V! Es un cmcoan una burbuia	Naes_nmonstr_a. _' Es un chico en una burbujJ.	N° &\$É" !'!P.'!\$'!!!9:	E_ _n chic0 en un_burbuja.
Imagen 6	4PéP€zcU;gQel5tina. MYéfnJifgranpéllejeria.	Parezco 'g_ela' tina. Mira mi gran pellejeria.	4PZirezcm gelatina. l 'Mnra¥mn gran pellejena.	__? _ , _____o __ __0__ ____0, , _____ _____
Imagen 7	Tenemos que enfrentar al ejército de\infiernc	Tenemo5 que enfrent_r al ejércitodel infierno	Tenemcs que enfrnlar al ejérciw del infizmc	Tenemo5 que enfrenlar al ejército del in_erno
Imagen 8	NO RECONOSE NADA	Ouc_ Ou_crcsunn gntlcrn7 fucra dc m_v_sta	E `æFu"æ.r.æ'd\$Wisu	0_0_?_u_0n___0 _0_0__ __0_0_000_8_0
Imagen 9	Ouizn cl upcfque dm cl uv:,u\$ llevaba una capa de mz\$ Deieiucsla	a_Jil___cl r)_u_uR _i__cl _Jvr_u lleu._ha_rn_coiJ__e n_a_. DejAmoilo.	1:.{ .4 .,.45*.ib llevihaunmqprde mis. Deiéinsgl	_____ __e0____' _____ _____.____. ____0__ ____'0____'.i..

Imagen 10	puedecmgcrmqmrs gticsw wadweWo were? sw	c_UCdCCO?_r CU3_4J_er S'_T_CS_ 1adie)_qJiere? si	;Pued0 cqgarcuaiquiar sitio si name 10 aware? S1.	Pued0 coger cualquier sitio 5i nadie la quie.re? Si.
------------------	---	---	---	--

Tabla 3. Resultados arrojados por Tesseract y GOCR con un pre-procesamiento de las imágenes.

2.4.1 Selección del OCR a utilizar

Del conjunto de programas de tipo OCR investigados y analizados con anterioridad, y basándose en los resultados emitidos de la toma de doscientas regiones de subtítulos, de treinta películas distintas, de los cuales se pone a disposición una muestra en las tablas anteriores, el OCR que mejores resultados obtuvo en la extracción de dichos subtítulos, y por tanto el que se decide seleccionar, es GOCR. Teniendo en cuenta que sus resultados son óptimos cuando las imágenes a procesar son previamente binarizadas, alcanzando de esta forma una eficiencia alrededor del sesenta y tres por ciento, mientras que el resultado arrojado por Tesseract está sobre el cincuenta y dos por ciento del total de caracteres y espacios encontrados. También es importante anotar que GOCR es un software con licencia GNU GPL, con lo cual, cumple con la línea de desarrollo del proyecto. Además la utilización de este OCR es muy sencilla a través de la consola de comando, unido a que posee una serie de métodos pre-definidos para el trabajo con las imágenes y el texto, de lo cual carece Tesseract. Otra característica importante de GOCR es que los caracteres que no reconocen, los representa con un guion bajo (“_”), de forma que en la etapa de pos-procesamiento es más fácil la corrección ortográfica con palabras que presentan guion bajo, que no con caracteres extraños; esto se debe a que muchas veces es una palabra que sólo le falta alguna letra, pero representa una palabra con sentido, no ocurriendo así con Tesseract.

2.5 Modelo del Componente

El modelo del Componente está basado en el trabajo con el OCR GOCR, ya que apoyándose en las pruebas realizadas, fue el que mejores resultados brindó, prefijando que su eficiencia depende en gran medida del pre-procesamiento por el cual pase la imagen y de la calidad que posea la misma. Las imágenes que son binarizadas antes de entrar al sistema, emiten mejores resultados que las demás (ver Tabla 3). Sin embargo, en la mayoría de los casos, el resultado final es totalmente dependiente de la calidad del diccionario con el cual se realice el proceso de pos-procesamiento. El Modelo del Sistema se muestra en la Figura 2.

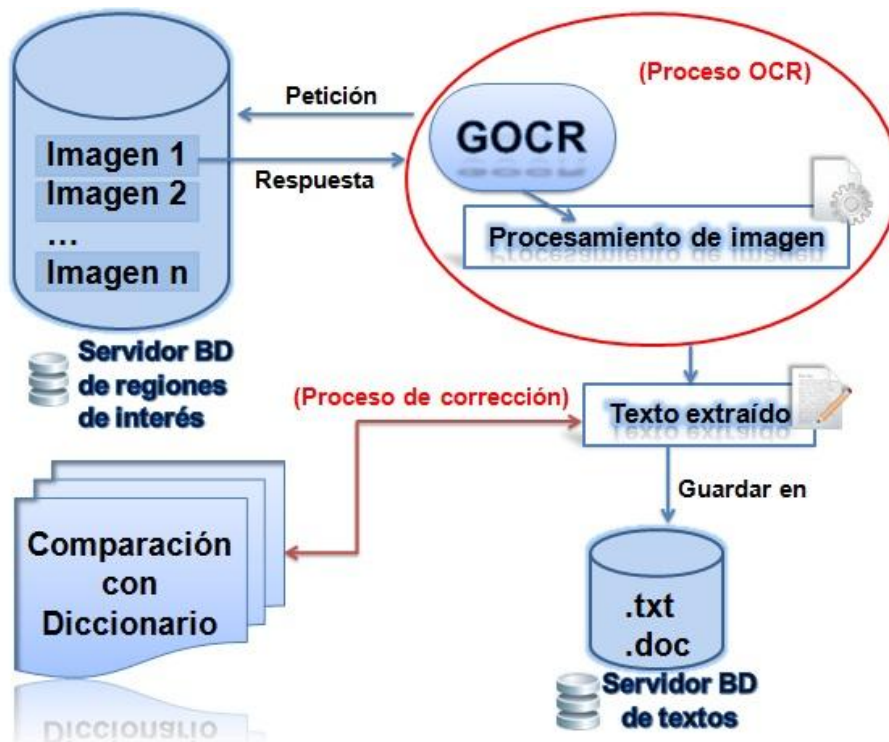


Figura 2. Modelo del Componente.

➤ **1ro- Obtención de las regiones de interés.**

Se debe entrar una imagen de formato .bmp o.jpg (preferentemente), debido a que estas son dos de varias extensiones, con las cuales el OCR mejor trabaja, o sea mejor reconoce el texto presente en la imagen y de acuerdo con las pruebas realizadas, el tipo de imagen que mejor reconoce el GOCR son las entradas con un pre-procesamiento binario (*imagen en blanco y negro*) de la misma. Según las pruebas realizadas, algunos resultados obtenidos de la investigación que se deben considerar para que el OCR realice el proceso de extracción de texto de forma más eficiente, son los siguientes:

- ✓ Las regiones de texto deben tener un alto contraste con el fondo.
- ✓ El texto presente en las imágenes debe estar alineado horizontalmente.
- ✓ El texto debe tener un fuerte borde vertical con respecto al fondo.
- ✓ Las imágenes deben tener no más de 3 líneas de texto.
- ✓ Las regiones de texto deben tener una dimensión entre 40 y 100 pixeles de altura y entre 200 y 350 de ancho y con un peso no superior a 40 Kb. ¹⁷

¹⁷KB, kilo byte, unidad de medida digital.

➤ **2do- Extracción del texto presente en las imágenes.**

Luego de realizada la petición de una nueva imagen a procesar y obteniéndose como respuesta dicha imagen, se realiza el proceso de extracción del texto presente en ésta, o sea, del subtítulo que aparece en la misma, este proceso se realizará con la ayuda del software GOCR. Dicho proceso de extracción de texto será llevado a cabo a través de la consola de comando con la siguiente instrucción:

gocr imagen_x.jpg -o salida.

La anterior instrucción se puede describir como el llamado al proceso del OCR seleccionado, en este caso **gocr**, la imagen a la cual se le realiza la extracción del texto (**imagen_x.jpg**), y el fichero **-o salida**, el cual posee el texto que se extrajo de dicha imagen y lo guarda en un fichero de texto sencillo (text/plain) por defecto, para que luego este texto extraído pueda ser manejado como texto plano y pueda ser editado.

➤ **3ro- Pos-procesamiento del texto extraído.**

Al texto almacenado se le realiza un proceso de revisión, lo cual consiste en comprobar mediante reglas XML (previamente definidas y cargadas) y diccionarios, las faltas de ortografía presentes en el texto extraído, por ejemplo: **revolucion** con **revolución**, además de los errores sencillos de cambio de caracteres en las palabras como **nonte** por **monte**, la aproximación de palabras ilógicas con palabras reales, ejemplo: **som_s** por **somos**, unido a los signos de puntuación, y en el mejor de los casos, si lo permite el diccionario, la gramática y la sintaxis del texto. Aunque vale destacar que este no es un proceso cien por ciento eficiente sin la revisión y la corrección de una persona.

➤ **4to- Almacenamiento del texto.**

El texto plano presente en el fichero de salida del GOCR será extraído y almacenado en un nuevo fichero previamente definido, este fichero puede ser un .doc, .txt, o cualquier otro de edición de texto que se desee, pero antes de ser guardado, se le realizará un proceso de comprobación del texto de salida del OCR al fichero, con el último texto almacenado. El objetivo de dicha comparación es que no se guarde dos veces el mismo texto, ya que las regiones de interés pueden estar repetidas y poseer los mismos textos por disímiles factores, como son: cambio de fondo de la imagen, o subtítulo presente en demasiados fotogramas del video. Si el texto obtenido como salida del OCR, no es igual al último texto almacenado, será guardado, pero si ocurre lo contrario no se almacenará, para evitar la repetición de contenido. Es válido aclarar que puede aparecer el mismo subtítulo dentro del video, pero

no de forma que aparezca uno detrás del otro, o sea, que aparezcan en momentos de tiempo diferentes.

2.6 Conclusiones Parciales

Teniendo en cuenta que la investigación en este capítulo fue orientada hacia software no privativo, se puede concluir que se obtendrán mayores beneficios, debido a que se podrá contar con una herramienta libre de costo que sirva de apoyo en el proceso de extracción del texto presente en el subtítulo de las imágenes de video, además de que ello responde a las necesidades del proyecto. Los elementos abordados en el capítulo permiten obtener resultados favorables para la investigación, ya que posibilitan definir las bases para implementar la solución del problema de la investigación, además permite entender mejor cómo y con qué trabajará el Componente a desarrollar. Por tanto, ya puede iniciarse la etapa de construcción del Componente, a partir de la selección de GOCR como herramienta de apoyo principal para la extracción del texto, y el modelo del componente anteriormente descrito.

Capítulo 3: “Resultados obtenidos por el Componente para la extracción de subtítulos presente en fotogramas de videos”

3.1 Introducción

En este capítulo se abordan los resultados obtenidos después de haber aplicado la propuesta así como las ventajas que brinda este modelo. El enfoque principal consiste en asegurar a través de los experimentos realizados, que el modelo propuesto es adecuado para la detección de subtítulos presentes en fotogramas de video. Se presenta una descripción del modelo activo y a través de los resultados obtenidos se detallan los pasos necesarios para el reconocimiento y extracción de los subtítulos presentes en fotogramas de videos.

3.2 Requisitos para la integración con la Plataforma de Codificación e Indexación

Los requisitos de integración definidos para la integración con la plataforma son los siguientes:

➤ **Cancelar Proceso:**

El proceso podrá ser cancelado en cualquier momento en que el usuario lo desee, posibilitando de esta forma que se borren automáticamente todos los registros creados en la máquina por dicho proceso, de forma tal que no quede ninguna evidencia del mismo; de manera que si el usuario desea repetirlo, éste empiece desde el principio nuevamente.

➤ **Porcentaje de realización del Proceso:**

Este requisito tiene como objetivo permitirle al usuario apreciar el porcentaje de cumplimiento de la realización del proceso en general, o sea, el estado en que se encuentra el mismo. Le da una idea a los usuarios de cuánto se ha procesado y cuánto falta por procesar aún.

➤ **Nombre del Proceso:**

Nombre que identifica al componente de extracción y comparación de subtítulos presentes en fotogramas de videos.

➤ **Envío de errores encontrados durante el proceso:**

Este se realizará con el fin de permitirle al usuario visualizar los distintos errores que pueden ir ocurriendo antes o durante la realización del proceso de extracción y comparación de subtítulos. El

mismo debe permitir detectarlos, corregirlos en la medida de lo posible y mostrarlos al usuario. Se pueden encontrar errores, tales como: no se encuentra instalado el GOCR, o fue cambiada la dirección de las imágenes a las cuales se le realizará el proceso de extracción de subtítulo.

3.3 Modelo de Diseño

Un modelo de diseño no es más que un conjunto de diagramas que describen el diseño lógico de una aplicación o Sistema, dentro de éste se encuentran los Diagramas de Clases del Diseño.

3.3.1 Clases del diseño

Una clase de diseño es una abstracción de una clase o construcción similar en la implementación del sistema y tiene operaciones, parámetros, atributos y tipos; necesarios para su implementación en el lenguaje de programación elegido.

3.3.2 Diagrama de clases del diseño

Un diagrama de clases del diseño muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Este tipo de diagramas se utiliza para modelar la vista de diseño de un sistema, el cual incluye modelar el vocabulario del sistema, o modelar esquemas, que permiten al desarrollador tener una mejor visión de lo que va a implementar.

Los diagramas de clases del diseño también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Estos diagramas de diseño son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

En la Figura 3 se muestra el Diagrama de Clases del Diseño del Componente para la extracción de subtítulos presente en fotogramas de videos, donde se presentan las clases empleadas en la solución del problema, de conjunto con sus atributos, operaciones y relaciones establecidas entre ellas.

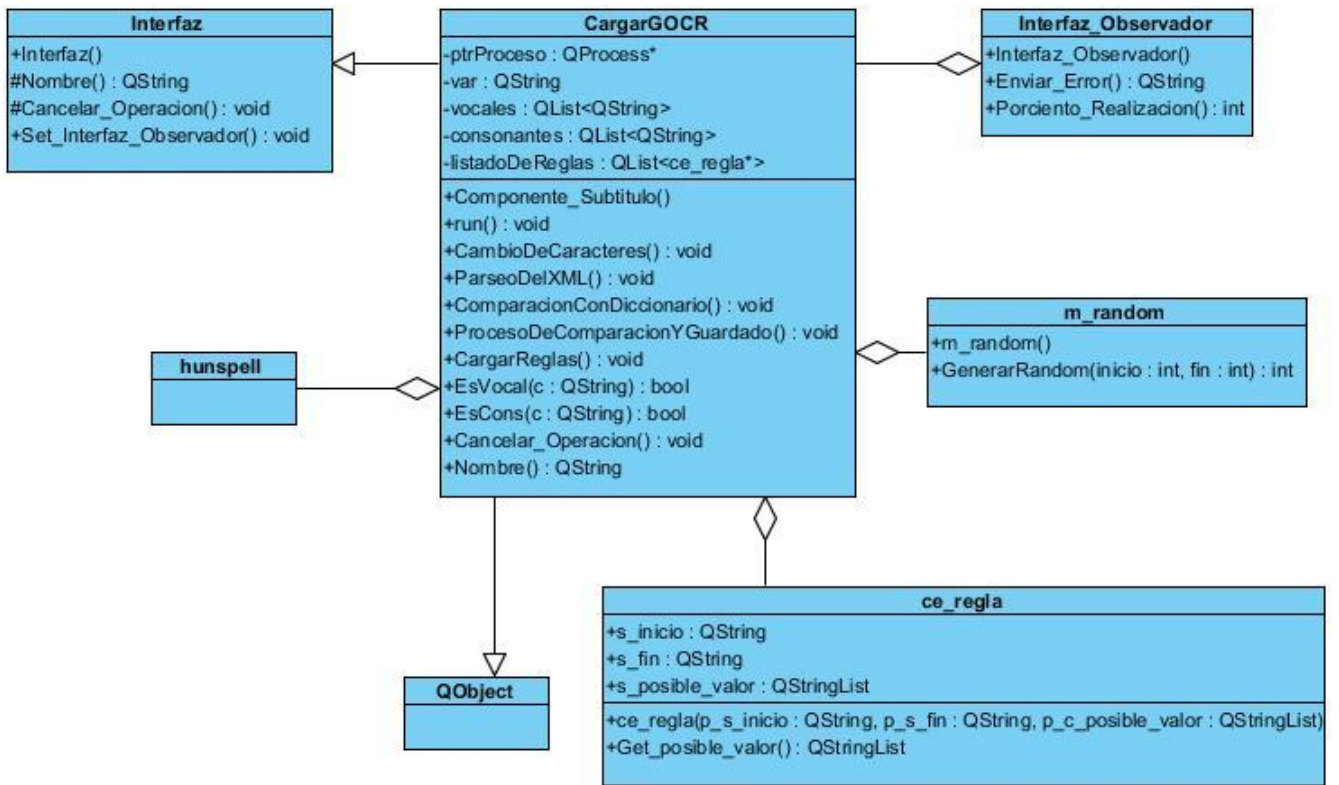


Figura 3. Diagrama de Clases del Diseño.

En el Diagrama de Clases del Diseño del Componente para la extracción de subtítulos presentes en fotogramas de videos, se muestran las clases que componen dicha aplicación, entre estas se encuentra la clase *Componente_Subtitulo*, que es la más importante, ya que la misma abarca los principales métodos para la realización de las funcionalidades que dan solución al problema de la investigación, tales como *run()* y *CambioDeCaracteres()*. También están presentes las clases *m_random* y *ce_regla*, las cuales son utilizadas en el método *ParseoDelXML()* de la clase *Componente_Subtitulo*. Esta última es heredera de la clase *QObject*, la cual es una librería propia de Qt Creator, usada para dar solución a problemas específicos, tales como la ejecución del software GOCR, así como de *Hunspell* que es una librería que permite la corrección del texto extraído a través de diccionarios previamente cargados.

3.4 Modelo de Implementación

El modelo de implementación está compuesto por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe

la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (32).

3.4.1 Diagrama de Componentes

Uno de los artefactos que se generan como parte del modelo de implementación es el Diagrama de Componentes, el cual muestra un conjunto de elementos del modelo, tales como subsistemas de implementación y sus relaciones. Se utiliza para modelar la vista estática de un sistema, además de mostrar la organización y las dependencias lógicas entre un conjunto de componentes del software.

En el Diagrama que aparece a continuación se muestra la relación de dependencia que existe entre el paquete de clases *Componente de extracción de subtítulos* y la *Librería Hunspell*, así como la dependencia del primero con el programa *GOCR*. El *Componente de extracción de subtítulos* contiene todas las clases de las cuales depende para realizar sus funcionalidades. Mientras que el *GOCR* es el software utilizado para la extracción del texto de los fotogramas y *Hunspell* es una librería que nos permite la corrección del texto extraído.

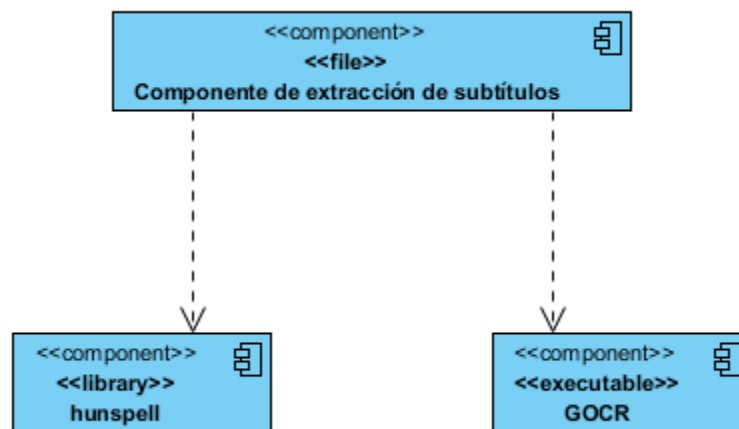


Figura 4. Diagrama de Componentes.

3.5 Pruebas

Las pruebas son un elemento crítico para la garantía de la calidad del software, éstas no mejoran ni aseguran la calidad del mismo, pero sí lo hace indirectamente previendo un grupo de debilidades asociadas a la organización y sus riesgos. Estas pruebas representan una revisión final de las especificaciones, del diseño y de la codificación del software; para su ejecución existen diferentes

estrategias, tipos y métodos de pruebas. Los tipos de pruebas son varios: pueden ser de funcionalidad, usabilidad, fiabilidad, rendimiento y soporte (33).

Teniendo en cuenta que la prueba es un proceso de ejecución con la intención de descubrir errores y que un buen caso de prueba es aquel que tiene una probabilidad muy alta de descubrir un nuevo error, se realizan pruebas de validación. Estas pruebas se sustentan en la necesidad de minimizar los cambios de caracteres o los errores ortográficos en las palabras extraídas del subtítulo. Para medir el correcto funcionamiento del componente se utilizarán técnicas de validación empleando como métrica el cálculo del Recall (R) y Precision (P) del proceso final de la extracción del texto. Para los algoritmos con materiales audiovisuales, estas variables quedan definidas de la siguiente forma (34):

$$R = \frac{\text{correctos}}{\text{correctos} + \text{falsos negativos (Fn)}}$$

$$P = \frac{\text{correctos}}{\text{correctos} + \text{falsos positivos (Fp)}}$$

Estas pruebas se realizaron comprobando visualmente el resultado obtenido luego de la ejecución de los métodos utilizados en el Componente para la extracción de subtítulos presente en fotogramas de videos, contra el texto extraído de una muestra de cien imágenes, las cuales son las mismas utilizadas en las pruebas realizadas para la selección del OCR a utilizar en el componente. Para ello se documentó: la cantidad de palabras extraídas con respecto al texto original, la cantidad de faltas de ortografía que presentaban estas palabras, la cantidad de faltas de ortografía corregidas y la cantidad de caracteres no pertenecientes al texto, calculando para cada uno de ellos los falsos positivos, falsos negativos, Recall y Precision.

A continuación los resultados obtenidos del análisis del texto extraído de los subtítulos y del texto real presente en las imágenes con las cuales se realizó la prueba:

Cantidad de palabras extraídas con respecto al texto original					
Cantidad total de palabras	Incorrectas	Fp	Fn	Recall	Precision
809	327	31	296	52.4	91.3

Tabla 4. Cantidad de palabras extraídas.

Cantidad de faltas de ortografía					
Cantidad total de palabras	Incorrectas	Fp	Fn	Recall	Precision
809	301	12	289	54.9	96.6

Tabla 5. Cantidad de faltas de ortografía.

Cantidad de caracteres extraídos					
Cantidad total de caracteres	Incorrectas	Fp	Fn	Recall	Precision
1131	589	281	308	65.6	67.7

Tabla 6. Cantidad de caracteres extraídos

La Tabla 4 muestra la cantidad total de palabras presentes en 100 imágenes a las cuales se le realizaron las pruebas, éstas devolvieron como resultado la detección de un total de 809 palabras de las cuales se tomaron como incorrectas 327, de ellas se extrajeron 31 palabras que no existían (Fp), las que en su mayoría eran artículos y 296 que no se extrajeron que si existían (Fn). En la Tabla 5 se muestran los resultados de las 301 palabras que resultaron ser incorrectas, de las cuales 12 se corrigieron mal (Fp) y 289 no se reconocieron como faltas de ortografía (Fn).

En la Tabla 6 se muestra la prueba realizada a la extracción del texto de veinte de las imágenes, donde se contó la cantidad de caracteres totales extraídos por el componente (1131), de los que resultaron ser incorrectos un total de 589, de estos se tomaron como Falsos positivos los que se representaron como texto en donde no lo había (281) y la cantidad de caracteres no extraídos por la herramienta, como Falsos negativos (308).

Analizando lo anterior, se puede observar que el algoritmo funciona con un *recall* cerca de 60, esto se debe a que existe un gran número de falsos negativos, los cuales son directamente proporcional a la cantidad de reglas XML con que se trabaje en el componente, la calidad del diccionario y principalmente el buen funcionamiento del OCR seleccionado;

En el caso del *precision* del algoritmo, muestra números superiores a 90 para las dos primeras pruebas y superior a 67 para la tercera. Si se analiza de forma general, es mayor que su *recall*, esto se debe a su alta sensibilidad, lo que proporciona la detección de un número mayor de falsos positivos, siendo ello un problema, el cual también depende de la cantidad de reglas XML con que se trabaje en el

componente, la calidad del diccionario seleccionado y principalmente en las características del OCR seleccionado y su funcionamiento para imágenes con texto.

3.6 Conclusiones parciales

En este capítulo se mostraron los resultados obtenidos en la etapa de diseño, implementación y prueba. Se construyó el diagrama de Clases del Diseño, el cual garantizó el funcionamiento adecuado de los datos del sistema y sirvió de base para la fase de implementación. Como resultado final se obtuvo el código fuente del Componente para la extracción de subtítulos presentes en fotogramas de videos, posibilitando la extracción, almacenamiento y manipulación de dichos textos. El desarrollo de la aplicación concluyó con la realización de las pruebas, las que determinaron que existen algunos errores en el reconocimiento del texto presente en los subtítulos de videos, lo cual se debe fundamentalmente al OCR empleado en su desarrollo, ya que actualmente no existe un OCR de software libre lo suficientemente eficiente para dicho reconocimiento de caracteres en los textos presentes en imágenes de fotogramas de videos; a pesar de ello se implementó un parseo del texto extraído por el OCR, que compara el resultado con reglas de escritura del lenguaje español, para mejorarlo, además de realizar una corrección ortográfica con un diccionario en español, sin embargo, las pruebas realizadas indican que ello no es suficiente para obtener un resultado óptimo con la calidad requerida.

Conclusiones

Con el desarrollo de la presente investigación se obtuvo como resultado el Componente para la extracción de subtítulos presentes en fotogramas de videos, para los productos del Departamento de Señales Digitales de la Facultad 6 de la UCI, el cual tiene implementado las funcionalidades que fueron requeridas para dar solución al problema científico planteado, cumpliendo así con el objetivo trazado.

Durante todo el proceso de desarrollo se solucionaron las limitaciones en cuanto a la obtención y extracción del texto presente en los fotogramas de video, así como el almacenamiento de estos para su futuro uso y manipulación. Actualmente el Componente se encuentra disponible en el servidor del proyecto Captura y Catalogación de Media de la facultad 6.

A pesar de que el software que se desarrolló no tiene toda la eficiencia esperada, ya que actualmente no existe un OCR de software libre lo suficientemente eficiente para dicho reconocimiento de caracteres en los textos presentes en imágenes de fotogramas de videos; sus características sí posibilitarán realizar un gran número de operaciones con la información presente en ellos de forma automática, permitiendo al usuario un acceso rápido y seguro a dicha información. De este modo, el proceso de investigación culminó con el aporte de una importante herramienta para todos los sistemas del departamento que requieran su utilización, lo cual supera la ineficiente extracción manual del texto, empleada con anterioridad.

Recomendaciones

Para la obtención de mejores resultados en el futuro y para la puesta en marcha del Componente de extracción de subtítulos presentes en fotogramas de videos, se recomienda:

- ✓ Poner en explotación el Componente para la extracción del texto presente en los subtítulos de los videos, con el fin de aprovechar el gran cúmulo de información que en este se encuentra.
- ✓ Estudiar los OCR privativos, en aras de que se obtenga una mejor extracción del texto presente en los fotogramas de video y aumente la eficiencia del sistema.
- ✓ Agregar nuevas reglas XML para la corrección ortográfica del texto extraído.
- ✓ Emplear un diccionario más eficiente para la corrección ortográfica del texto extraído.

Trabajos Citados

1. **Echeverría, Javier.** *Impacto Cultural, Social y Lingüístico de las Nuevas Tecnologías de la Información y las Comunicaciones (TIC).* Biblioteca: Universidad de las Ciencias Informáticas. [En línea] [Citado el: 27 de Octubre de 2011.]
http://bibliodoc.uci.cu/html/impacto_ntic.htm .
2. Enciclopedia Cubana en la Red (ECURED). [En línea] [Citado el: 28 de octubre de 2011.]
http://www.ecured.cu/index.php/EcuRed:Enciclopedia_cubana
3. Gestión Inteligente de Activos Multimedia. [En línea] [Citado el: 28 de octubre de 2011]
http://www.isid.es/Spanish/product/producto_v01.htm
4. Centro de Aplicaciones de Tecnologías de Avanzada. [En línea] [Citado el: 1 de noviembre de 2011.] <http://www.cenatav.co.cu/es/publicaciones.html>
5. Diccionario de la Real Academia Española. [En línea] [Citado el: 4 de noviembre de 2011.]
<http://buscon.rae.es/drae1>
6. Diccionario de la Lengua Española. [En línea] [Citado el: 11 de noviembre de 2011.]
<http://www.wordreference.com/definicion/subtitulo>
7. **Universidad de las Ciencias Informáticas.** 2009. Conferencia: "Características del video. Conceptos", La Habana, Cuba, 2009.
8. Taller Synfig. [En línea] [Citado el: 4 de noviembre de 2011.]
<http://tallersynfig.wordpress.com/contenido/fotogramas-clave/>
9. Digital Digest. [En línea] [Citado el: 11 de noviembre de 2011.]
http://www.digital-digest.com/nickyguides_translated/keyframes_spa.htm
10. **NISO Prensa.** *Understanding Metadata.* Bethesda: s.n., 2006. ISBN: 1-880124-62-9.
11. **Korth y Silberschatz.** *Data Mining.* Bruselas: s.n., 2010.
12. **Rivera Suarez, Yoel y Echemendia Tourt, Fernando.** *Sistema para la Detección y Extracción de Texto en Imágenes Digitales.* UCI Ciudad de la Habana. Cuba. 2008. Trabajo de Diploma.
13. Web del Profesor. [En línea] ,2010. [Citado el: 13 de noviembre de 2011.]
<http://webdelprofesor.ula.ve/humanidades/raymond/computacion2/archivos/ImagDig.pdf>
14. *Sobre la Historia y la Estética de la Imagen Digital.pdf* [En línea] [Citado el: 14 de noviembre de 2011].
<http://www.pucsp.br/~gb/texts/SOBRELAHISTORIAYLAESTETICADELAIMAGENDIGITAL.pdf>
15. **Matsumoto, Yoshihide, Uemiya, Tadashi, Shishibory, Masami y Kita, Kenji.** *A Method for Detecting Subtitle Regions in Videos Using Video Text Candidate Images and Color Segmentation*

- Images*. Facultad de Ingeniería. Universidad de Tokushima. Japón. 2010. Revista internacional de Inteligencia Avanzada. Volumen 2. No.1.
16. **Wu, Víctor, Manmatha, Robert, Riseman, Edward.** *Finding Text in Images*. Departamento de Ciencias de la Computación. Grupo de Indexación y recuperación de Multimedia. Universidad de Massachusetts. EEUU. 2006. Publicación Científica.
 17. **Pastor y Gadea, Moisés.** *Aportaciones al Reconocimiento Automático de Texto Manuscrito*. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. España. 2008. Tesis Doctoral.
 18. **Vijayakumar, V. y Nedunchezian, R.** *A Novel Method for Super Imposed Text Extraction in a Sport video*. Departamento de Aplicaciones de la Informática. Universidad de Coimbatore. La India. 2011. Revista Internacional de Aplicaciones de la Informática. Volumen 15. No.1.
 19. **Jean Serra, J. Angulo.** *Segmentación de Imágenes en Color Utilizando Histogramas Bi-Variables en Espacios de Color Polar, Luminancia, Saturación y Matiz*. Centro de Morfología Matemática. Escuela de Minas de Paris. Francia. 2005. Revista Computación y Sistemas. Volumen 8. No.4.
 20. **Cano, Javier y Pérez, Juan C.** OCR (Optical Character Recognition). Instituto tecnológico de Informática. Valencia. España. 2011 [En línea] [citado el: 25 de enero de 2011].
<http://www.iti.es/media/about/docs/tic/02/2003-11-ocr.pdf>
 21. **Hua, Xian-Sheng, Chen, Xian-Rong, Wenyin, Liu y Zhang, Hong-Jiang.** *Automatic Location of Text in Video Frames*. Centro de Investigación de Microsoft en China. Beijing. China.
 22. Freewaregenius. [En línea] [Citado el: 19 de diciembre de 2011.]
<http://www.freewaregenius.com/2011/11/01/how-to-extract-text-from-images-a-comparison-of-free-ocr-tools/>
 23. Kultura 2.0 [En línea] [Citado el: 19 de enero de 2011]
<http://www.kulturklik.euskadi.net/wp-content/uploads/2011/04/pildora-OCR-2.pdf>
 24. Página Oficial de GNU Operating System. [En línea] [Citado el: 23 de enero de 2011]
<http://www.gnu.org/software/ocrad/>
 25. Página Oficial de OCRopus. [En línea] [Citado el: 23 de enero de 2011]
<https://code.google.com/p/ocropus/>
 26. Página Oficial de Tesseract. [En línea] [Citado: 27 de enero de 2011]
<http://code.google.com/p/tesseract-ocr/>
 27. RepairFAQ. [En línea] [Citado: 27 de enero de 2011]
<http://tesseract-ocr.repairfaq.org/>
 28. Página Oficial de GOCR. [En línea] [Citado: 29 febrero de 2011]

<http://jocr.sourceforge.net>

29. **Arias Marin, Marvin David.** *Definición de lenguaje de programación. Tipos. Ejemplos. Catedraprogramacion.*[En línea] [Citado el: 6 de febrero de 2011.]
<http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
30. **Schildt, Herbert.** *C++.* *Guía de autoenseñansa.* Madrid. España. ISBN 84-481-3203-3.
31. Lenguajes de Programación. [En línea][Citado el: 9 de febrero de 2011.]
<http://www.frt.utn.edu.ar/sistemas/paradigmas/lenguajes.htm>.
32. **MeRinde.** *Modelo de Implementación.* [En línea] [Citado el: 18 de mayo de 2011.]
http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=495&Itemid=291
33. **Pressman, Roger S.** *Ingeniería de software, un enfoque practico, chapter Técnicas de prueba del software.* 2001.
34. **Pupo Rodríguez, Luis Angel.** *Implementación de un componente para la separación automática de segmentos de videos integrado al Sistema de Captura y Catalogación de Medias.* UCI. Ciudad de la Habana. Cuba. 2011. Trabajo de Diploma.

Bibliografía

1. **Rivera Suarez, Yoel y Echemendia Tourt, Fernando.** *Sistema para la Detección y Extracción de Texto en Imágenes Digitales*. UCI. Ciudad de la Habana. Cuba. 2008. Trabajo de Diploma.
2. **NISO Prensa.** *Understanding Metadata*. Bethesda: s.n., 2006. ISBN: 1-880124-62-9.
3. Sobre la Historia y la Estética de la Imagen Digital.pdf. [En línea] [Citado el: 14 de noviembre de 2011].
<http://www.pucsp.br/~gb/texts/SOBRELAHISTORIAYLAESTETICADELAIMAGENDIGITAL.pdf>
4. **Wu, Víctor, Manmatha, Robert, Riseman, Edward.** *Finding Text in Images*. Departamento de Ciencias de la Computación. Grupo de Indexación y recuperación de Multimedia. Universidad de Massachusetts. 2006. Publicación Científica.
5. **Pastor y Gadea, Moisés.** *Aportaciones al Reconocimiento Automático de Texto Manuscrito*. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. 2008. Tesis Doctoral.
6. **Wolf, Christian y Jolian, Jean-Michel.** *Model Base Text Detection in Images and Video: a Learning Approach*. LIRIS INSA. Lyon. Francia. 2004. Reporte Técnico.
7. **Arriagada García, Cristian y Aracena-Pizarro, Diego.** *Detección y Reconocimiento de Señales de Tránsito Utilizando Matching de Chamfer*. Escuela Universitaria de Ingeniería Industrial, Informática y de Sistemas. Universidad de Tarapacá. Arica. Chile. 2007. Ingeniare. Revista chilena de ingeniería, vol. 15 N° 2, 2007, pp. 174-184.
8. **Hua, Xian-Sheng, Chen, Xian-Rong, Wenyin, Liu y Zhang, Hong-Jiang.** *Automatic Location of Text in Video Frames*. Centro de Investigación de Microsoft en China. Beijing. República Popular de China.
9. **Korth y Silberschatz.** *Data Mining*. Bruselas: s.n., 2010.
10. *Automatic Text Segmentation and Text Recognition for Video Indexing*. **Lienhart, Rainer y Effelsberg, Wolfgang.** Universidad de Mannheim. Mannheim. Alemania. 1998.
11. **Cesaire Velázquez, Yanet y Fonseca Avila, Ana J.** *de Algoritmos para la Detección y Localización de texto en Imágenes*. UCI. Ciudad de la Habana. Cuba. 2010. Trabajo de Diploma.
12. **Chen, Datong, Odobez, Jean-Marc y Boulard, Hervé.** *Text detection and recognition in images and video frames*. Instituto de Inteligencia Perceptual y Artificial de Swisa. Martigny. Swisa. 2003
13. **Sonsoles Herrero, Martín.** *Análisis comparativo de técnicas de segmentación de secuencia de video basado en el modelado del fondo*. Universidad Autónoma de Madrid. España. 2009. Proyecto fin de Carrera.

14. **Matsumoto, Yoshihide, Uemiya, Tadashi, Shishibory, Masami y Kita, Kenji.** *A Method for Detecting Subtitle Regions in Videos Using Video Text Candidate Images and Color Segmentation Images.* Facultad de Ingeniería. Universidad de Tokushima. Japón. 2010. Revista internacional de Inteligencia Avanzada. Volumen 2. No.1.
15. **Vijayakumar, V. y Nedunchezian, R.** *A Novel Method for Super Imposed Text Extraction in a Sport video.* Departamento de Aplicaciones de la Informática. Universidad de Coimbatore. La India. 2011. Revista Internacional de Aplicaciones de la Informática. Volumen 15. No.1.
16. Centro de Aplicaciones de Tecnologías de Avanzada. [En línea] [Citado el: 1 de noviembre de 2011.] <http://www.cenatav.co.cu/es/publicaciones.html>
17. **Bertini, M, Colombo, C y Del Bimbo, A.** *Automatic Caption Localization in Videos Using Salient Points.* Departamento de Sistemas e Informatica. Universidad de Firenze. Italia. 2001.
18. **Caimi, Annamaria.** *Audiovisual Translation and Languages Learning: The Promotion of Intralingual Subtitle.* Universidad de Pavia. Italia. 2006. Revista de Traducción Especializada.
19. **Navarro, Manuel.** *Byte. Movilidad: las TIC del siglo XXI.* 135, Madrid : Publicaciones Informáticas MKM, 2007.
20. **Jean Serra, J. Angulo.** *Segmentación de Imágenes en Color Utilizando Histogramas Bi-Variables en Espacios de Color Polar, Luminancia, Saturación y Matiz.* Centro de Morfología Matemática. Escuela de Minas de Paris. Francia. 2005. Revista Computación y Sistemas. Volumen 8. No.4.
21. **De Giusti, Marisa, Vila, María, Lujan Villarreal, Gonzalo.** *Digitalización y Reconocimiento de Documentos Manuscritos para la Preservación del Patrimonio Cultural.* Universidad Nacional de la Plata. Argentina. 2008.
22. **Quintero Pérez, Reinier.** *Propuesta de una técnica para la detección de imágenes por bordes o contornos en secuencias de video.* UCI. Ciudad de la Habana. Cuba. 2011. Trabajo de Diploma.
23. **Krishni, Bal, Pustakalaya, Puraskar y Rupakheti, Prajwal.** *Final Report on Nepali Optical Character Recognition*
24. **Arias Marin, Marvin David.** *Definición de lenguaje de programación. Tipos. Ejemplos. Catedraprogramacion.* [En línea] [Citado el: 6 de febrero de 2011.] <http://catedraprogramacion.foroactivo.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
25. **Schildt, Herbert.** *C++.* *Guía de autoenseñansa.* Madrid. España. ISBN 84-481-3203-3.
26. **MeRinde.** *Modelo de Implementación.* [En línea] [Citado el: 18 de mayo de 2011.] http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=495&Itemid=291

27. **Pressman, Roger S.** *Ingeniería de software, un enfoque practico, chapter Técnicas de prueba del software, pages 281–299.* 2001.
28. **Pupo Rodríguez, Luis Angel.** *Implementación de un componente para la separación automática de segmentos de videos integrado al Sistema de Captura y Catalogación de Medias.* UCI. Ciudad de la Habana. Cuba. 2011. Trabajo de Diploma.