

Universidad de las Ciencias Informáticas

Facultad X



**Título: Sistema detector de piel utilizando Redes Neuronales
Artificiales**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor(es): Cesar Rodelmis Aguilera Valls.

Tutor(es): Ing. Siovel Rodríguez Morales.
Ing. Mailen Proenza Guerra

Ciudad de la Habana
Curso 2008 – 2009

DECLARACIÓN DE AUTORIA

Declaro que soy el único autor de este trabajo y autorizo a 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 _____.

Cesar Rodelmis Aguilera Valls

Autor

Ing. Siovel Rodríguez Morales

Tutor

Ing. Mailen Proenza Guerra

Tutora



*La vida es una obra de teatro
que no permite ensayos...*

*Por eso, canta, ríe, baila,
llora y vive intensamente cada
momento de tu vida antes que
el telón baje y la obra termine
sin aplausos.*

Charles Chaplin

AGRADECIMIENTOS.

Agradezco a mis padres por haberme guiado y apoyado en todos estos años en los momentos buenos y malos. A mis abuelos por formarme y guiarme. A mis hermanos Raquel, Alvaro y Jorge Luis por su cariño. A mi familia por ayudarme a lo largo de mi vida estudiantil. A mis tutores por el apoyo que me han brindado. A mis hermanos de la vocacional a los cuales quiero y estimo por los momentos vividos durante 8 años, por su amistad y consideración. A mis compañeros de grupo, los viejos y los nuevos, por haberme soportado durante tanto tiempo. A mis amigos de Santiago por su consejo y amistad desinteresada. A todos aquellos que de una forma u otra, me han ayudado y han permitido que este sueño se haga realidad.

RESUMEN

En la Universidad de las Ciencias Informáticas se está implementando, desde el año 2005 a petición de la Oficina de Seguridad para las Redes Informáticas, un sistema de filtrado que pretende; regular los contenidos inadecuados que provienen de Internet. Este sistema de filtrado tiene como componente principal una Base de Datos de URLs Categorizadas mediante la cual se toma la decisión de autorizar o denegar el acceso a los recursos solicitados. Dado que es inviable la categorización de las URLs de forma manual se necesita para esto un proceso automático por lo que surge el Motor de Categorización Inteligente de Contenido. Este motor es el encargado de clasificar de forma automática las páginas Web, para poder dar un criterio general de una página Web es necesario la categorización tanto de su texto como de sus imágenes. Las imágenes constituyen un componente esencial en las páginas Web, estas son mayormente usadas para hacer más atractivos los contenidos, sin embargo también están contribuyendo a la existencia de contenidos dañinos, que a su vez pueden ser ilegales, tales como la pornografía y la pedofilia. Para la clasificación de las imágenes se siguen una serie de etapas bien definidas. Dentro de estas etapas la segmentación es el proceso más importante, y a la vez el más crítico de un sistema de tratamiento computacional de imágenes, convirtiéndose en uno de los temas más estudiados y sobre todo constituye el primer paso en la segmentación de imágenes. En este artículo se propone un sistema para la segmentación de imágenes mediante el color de la piel basados en Redes Neuronales Artificiales con vista a su aplicación en el módulo para la categorización de imágenes pornográficas. Su aplicación permitirá mejorar la categorización de imágenes pornográficas.

Palabras claves: Segmentación, Redes Neuronales Artificiales.

ÍNDICE

INTRODUCCIÓN.....	1
1. FUNDAMENTACIÓN TEÓRICA.....	5
1.1 INTRODUCCIÓN	5
1.2 ÁMBITO INTERNACIONAL	5
1.3 ÁMBITO NACIONAL.....	5
1.4 SEGMENTACIÓN DE IMÁGENES	6
1.4.1 Técnicas de segmentación.....	9
1.4.2 Aplicaciones del proceso de segmentación.....	10
1.5 REDES NEURONALES ARTIFICIALES	11
1.5.1 Mecanismo de aprendizaje.....	14
1.5.1.1 Aprendizaje supervisado.....	14
1.5.1.2 Aprendizaje no supervisado.....	14
1.5.2 MLP	15
1.5.3 Mapas auto-organizados (SOM)	16
1.5.4 Learning Vector Quantization	18
1.5.5 Support Vector Machines	18
1.6 TECNOLOGÍA A UTILIZAR.....	19
1.6.1 Lenguaje de programación.....	19
1.6.2 Bibliotecas disponibles.....	21
1.6.2.1 Librería FANN	22
1.6.2.2 Librería OpenCV	22
1.6.3 IDE de desarrollo.....	22
1.6.4 Herramienta Case.....	23
1.6.5 Lenguaje de modelación.....	24
1.6.6 Otras Herramientas.....	24
1.7 METODOLOGÍA DE DESARROLLO.....	25
1.7 CONCLUSIONES	30
2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.....	31
2.1 INTRODUCCIÓN.....	31
2.2 DESCRIPCIÓN DEL PROBLEMA.....	31
2.3 MODELO DE DOMINIO.....	31
2.4 SOLUCIÓN PROPUESTA.....	32
2.5 SEGMENTACIÓN.....	33
2.5.1 Modelado del color de la piel.....	33
2.6 PERCEPTRÓN MULTICAPA.....	34
2.6.1 Arquitectura de la MLP.....	35
2.6.2 Fase de Entrenamiento.....	39
2.6.3 Fase de ejecución.....	42
2.7 LEVANTAMIENTO DE REQUISITOS.....	43
2.8 PLANIFICACIÓN DEL PROYECTO POR ROLES.....	44
2.9 HISTORIAS DE USUARIOS Y TAREAS.....	45
2.10 LISTA DE RIESGO.....	50
2.11 DIAGRAMA DE COMPONENTES.....	52
2.12 PLANIFICACIÓN DE HISTORIAS DE USUARIO.....	53
2.12.1 Plan de Releases.....	54
2.13 CONCLUSIONES.....	55
3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	56
3.1 INTRODUCCIÓN.....	56
3.2 CASOS DE PRUEBAS.....	56
3.2.1 Caso de prueba para la historia de usuario M_DP_0.1.....	56
3.2.2 Caso de prueba para la historia de usuario M_DP_0.2.....	57
3.3 RESULTADOS OBTENIDOS.....	59
3.4 ACERCA DEL TIEMPO DE DESARROLLO.....	59

3.5 CONCLUSIONES.	60
CONCLUSIONES GENERALES	61
REFERENCIAS BIBLIOGRÁFICAS.	62
BIBLIOGRAFÍA.	65
FLUJO DE RECONOCIMIENTO DE IMÁGENES.	67
ESTRUCTURA DE LAS NEURONAS	67
PERCEPTRÓN MULTICAPA.	68
ARQUITECTURA DEL PROYECTO MOCIC	69
GLOSARIO DE TÉRMINOS	71

ÍNDICE DE FIGURAS

Figura. 1.1: Proceso de segmentación.....	8
Figura. 1.2: Fases de XP.....	28
Figura. 2.1 Modelo de dominio del problema.....	32
Figura. 2.2 Componentes de una red neuronal	35
Figura. 2.3 Arquitectura de MLP con modelo RGB normalizado	37
Figura. 2.4 Topología de la MLP con los datos de entrada binarios	38
Figura. 2.5 Diagrama de componente fase entrenamiento.....	52
Figura. 2.6 Diagrama de componente fase Segmentación.	53

ÍNDICE DE TABLAS

ÍNDICE DE TABLAS

TABLA 1 LISTA DE RESERVA DEL PRODUCTO	54
TABLA 2 PLANIFICACIÓN DE ROLES	55
TABLA 3 LISTA DE RIESGO	61
TABLA 4 PLAN DE RELEASES	65

INTRODUCCIÓN

En la Universidad de las Ciencias Informáticas (UCI) se ha desarrollado, a petición de la Oficina de Seguridad para las Redes Informáticas (OSRI), un proyecto llamado Filtrado de Paquetes por Contenido (FILPACON) el cual pretende regular el flujo de materiales inadecuados a nuestra red. El componente fundamental de FILPACON es una Base de Datos de URLs Categorizadas (BDUC). Internet¹ en la actualidad constituye la mayor fuente de información a nivel mundial, su contenido equivale a 3 millones de veces la cantidad de libros escritos en la historia de la humanidad (1). Por tal motivo llenar la BDUC de forma manual es una tarea inviable.

Para mantener actualizada la BDUC se usará el Motor de Categorización Automatizada de Documentos HTML (MCADHTML). Uno de los módulos identificados en este motor es el módulo para la categorización de imágenes pornográficas, el cual es fundamental para la categorización general del documento HTML dado que las imágenes constituyen un componente importante en las páginas Web. Éstas son utilizadas, entre otras cosas, para hacer más atractivos los contenidos o añadir elementos gráficos como flechas de navegación, logotipos de empresas, avatares de usuarios entre otros. Hoy en día es casi inimaginable un sitio Web sin la presencia de imágenes. Sin embargo, estas contribuyen de forma decisiva a la existencia de contenidos dañinos en la red.

Para el reconocimiento de imágenes pornográficas se siguen una serie de pasos bien definidos ya que éste forma parte del reconocimiento de patrones ([ver anexo A](#)). La segmentación de la imagen es el proceso que subdivide una imagen en regiones homogéneas. Cada región constituye una parte u objeto en la escena. En otras palabras, la segmentación de la imagen es definida como la división de regiones que están conectadas, entonces cada píxel en la imagen adquiere una única etiqueta que indica a la región que pertenece.

En el Categorización Automático de Imágenes Pornográficas la segmentación se realiza a partir del color de la piel humana dado que este tipo de imágenes muestran, generalmente, personas desnudas. Estas imágenes muestran mucha piel y debido a esto el color de la piel es una característica básica usada para la categorización de tales imágenes. Actualmente este proceso se desarrolla mediante el modelo propuesto por Jones y Regh.

¹ Internet es un sistema de redes distribuidas que entre sus principales servicios está la www, el correo electrónico, etc.

INTRODUCCIÓN.

El modelo propuesto por Jone y Regh presenta dificultades a la hora de clasificar los píxeles, en piel y no piel, en una gran gama de colores cercanos al rojo, esto conlleva a una baja efectividad de la segmentación la cual actualmente es del 88%. Por esta razón la presente tesis de grado centró su desarrollo en la implementación de un algoritmo que permita clasificar los píxeles en piel o no piel basado en Redes Neuronales Artificiales (RNA).

En la actualidad las Redes Neuronales Artificiales juegan un papel importante como una herramienta efectiva para la solución de distintos problemas como clasificación y reconocimiento de patrones, agrupación, aproximación de funciones, entre otros. Las RNA son un intento de simular el procesamiento de información que realiza el sistema nervioso; pueden ser vistas como un sistema de procesamiento en paralelo y distribuido que consiste en un gran número de unidades (neuronas) conectadas entre sí, donde cada una de ellas es un sistema simple de cálculo que a partir de un número variable de entradas produce una salida

Dada la necesidad de mejorar la efectividad en la detección del color de la piel surge el siguiente **problema científico**: ¿Cómo detectar el color de la piel en imágenes, utilizando Redes Neuronales Artificiales?

Por lo expuesto se defiende la siguiente idea: Si se implementa un modelo de segmentación basado en RNA más efectivo que el existente, mejorará la efectividad del categorizador de imágenes.

Partiendo de la problemática planteada anteriormente se define como **objeto de estudio** el proceso de segmentación de imágenes y como **campo de acción** la segmentación basada en Redes Neuronales Artificiales.

A raíz de lo planteado se trazó como **objetivo general** Implementar un algoritmo detector de piel utilizando RNA, desglosándose éste en los siguientes **objetivos específicos**:

- Aplicar los patrones de diseños que se correspondan con el problema planteado.
- Aplicar el modelo de segmentación basado en RNA.
- Comparar la efectividad de la segmentación por RNA con respecto a la segmentación basada en el modelo propuesto por Jones y Regh.

Para cumplir con los objetivos planteados se definen las siguientes **tareas investigativas**:

1. Estudiar trabajos similares para poner en práctica la reutilización de componentes.
2. Estudiar las herramientas libres que puedan facilitar el desarrollo del presente trabajo.
3. Estudiar los métodos de segmentación basados en las Redes Neuronales Artificiales.
4. Estudiar el modelo de segmentación propuesto por Jones y Regh.
5. Aplicar la segmentación mediante ambos modelos para comprobar la efectividad del modelo basado en RNA con respecto al modelo propuesto por Jones y Regh.

Esta investigación tuvo como soporte procedimientos que permitieron estudiar las características del objeto de la investigación que no son observables directamente y dar cumplimiento a las tareas propuestas con anterioridad. Los métodos científicos y las técnicas empleados en el desarrollo de este estudio se describen a continuación:

Modelación: Se usó mediante UML para reflejar la estructura, relaciones internas y características de la solución a través de diagramas.

Histórico-lógico: Se emplea este método para el estudio del proceso de segmentación, posibilitando estudiar las tendencias, características y desarrollo.

Observación: Se utilizó para comparar la efectividad del modelo de segmentación basado en mapas Auto-organizados con respecto al modelo propuesto por Jones y Regh, mediante los resultados obtenidos en las pruebas.

Analítico-sintético: Se utilizó para comprender el proceso de categorización de imágenes mediante el análisis de los subprocesos y su funcionamiento como un todo en el proceso de categorización.

Estructura del contenido

Para entender de una mejor forma el presente documento es necesario concebir su organización y algunas reglas que se han definido para su comprensión, las cuales son:

- Las palabras en otro idioma se escribirán en letra cursiva.
- Las palabras con letras en mayúsculas representan acrónimos y estarán descritas en el Glosario de términos.
- Los nombres de los componentes de los diagramas presentados estarán en idioma inglés.

- Los títulos de artículos de diarios y revistas, prólogos y secciones de un libro, conferencias o charlas irán entre comillas.

El presente trabajo consta de tres capítulos los cuales en su conjunto abordarán todo el contenido referente al estudio y desarrollo de una solución para hacer de Guano un entorno de escritorio usable.

Capítulo 1: “FUNDAMENTACIÓN TEÓRICA”: El primer capítulo está centrado en la puesta al día con relación al campo de acción. Se analiza la existencia de soluciones a escala nacional e internacional que puedan reutilizarse. Se presenta una breve descripción del objeto de estudio. Además se identifican las tecnologías y herramientas, así como la metodología y técnicas para el desarrollo del software.

Capítulo 2: “ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA”: En el segundo capítulo se profundiza en el problema a resolver a través de su descripción y se muestra el modelo generado. Se presenta una descripción detallada del proceso de clasificación llevado a cabo en este problema específico, que es la segmentación de imágenes basadas en mapas Auto-organizados,. Se realiza una propuesta de solución y se identifican los requisitos funcionales y no funcionales que deben tenerse en cuenta.

Capítulo 3: “VALIDACIÓN DE LA SOLUCIÓN PROPUESTA”: En éste capítulo se detallan las pruebas realizadas al software, como de tipo caja negra, así como el impacto que tendrá el sistema una vez implantado.

1. FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

El procesamiento y análisis de imágenes es una disciplina entre la ingeniería y las matemáticas que está en pleno desarrollo debido a sus numerosas aplicaciones cinéticas e industriales, ayudadas por el gran avance tecnológico producido en los últimos años. La segmentación de imágenes es uno de los temas fundamentales más estudiado y útil en el procesamiento y análisis de imágenes. Esta permite cuantificación y visualización de los objetos de interés y de su calidad depende en gran medida el resultado final de la interpretación automática de una escena. En el presente capítulo se enfocará en la descripción, características y utilidad del proceso de segmentación.

1.2 Ámbito Internacional

A nivel internacional existen diversos tipos de segmentación de imágenes, los cuales son usados en diferentes áreas. Hasta ahora, no existe un método universal de segmentación, sino que está íntimamente ligado con la tarea a resolver y termina cuando satisface los objetivos del observador.

1.3 Ámbito Nacional

En el marco del Programa Intercosmo de los países socialistas en la década de 1980 se dispuso de imágenes satelitales en formato digital, permitiendo su procesamiento e interpretación asistida por las computadoras, incorporando los métodos de clasificación automatizados. En nuestro país el trabajo se enmarcó en la clasificación de imágenes de teledetección en particular en la selección de las muestras de aprendizajes y control y en la inclusión de descriptores de texturas en los vectores de rasgos de los patrones. Los métodos de segmentación usados para el reconocimiento de patrones se basaron en la segmentación por tono y textura para diferenciar las áreas a clasificar utilizando criterios de tono, textura o conjuntamente.

Las principales áreas de aplicación de los métodos de reconocimiento de patrones donde utilizan la segmentación han sido:

- ❖ Detección de islas de calor.
- ❖ Detección de focos contaminantes en ríos y costas.
- ❖ Estudio de la biomasa en pastos naturales y artificiales.
- ❖ Clasificación de cultivos agrícolas.
- ❖ Cartografía de la salinidad superficial del suelo.
- ❖ Inventario forestal de especies y la vigilancia contra incendios forestales.
- ❖ Investigaciones del medio geográfico, ambientales y meteorológicas.
- ❖ Investigaciones oceanográficas y marinas.
- ❖ Investigaciones geológicas e hidrogeológicas.

La asimilación y el desarrollo teórico alcanzado en Cuba en las técnicas de reconocimientos de patrones se ha visto manifestado en la programación de varios sistemas de procesamiento de imágenes, especialmente para las imágenes de teledetección los principales exponentes son: Telemap V2.0 (2), RSI (3), Tx Estudio (4) y Tn Estudio V2.0 (5).

Nuestro país no se ha quedado atrás y como se especificará más adelante en este capítulo el proceso de segmentación se usa en diferentes esferas como la medicina, biología y demás.

1.4 Segmentación de imágenes

La segmentación de la imagen es el proceso que subdivide una imagen en regiones homogéneas. Cada región homogénea constituye una parte o un objeto en la escena. En otras palabras, la segmentación de la imagen se define como la división de regiones que están conectadas, entonces cada píxel en la imagen adquiere una única etiqueta que indica a la región que pertenece (6).

La Segmentación de la imagen es el proceso más importante, y a la vez el más crítico, de un sistema de Tratamiento Computacional de Imágenes. Su función es aislar los objetos presentes en la imagen para proceder, posteriormente, a la etapa de identificación. Estos objetos conforman los descriptores que pasarán a formar parte del vector de características, el cual es pasado a los siguientes subprocesos. Si los resultados de la segmentación no son aceptablemente buenos, separación de un objeto en varias partes o agrupamiento de varios objetos en uno, la etapa siguiente de identificación no tiene ninguna posibilidad de actuar correctamente. No existe, actualmente, una solución general al problema de la segmentación. Por otra parte, si existen soluciones buenas en dominios concretos.

CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

La segmentación se basa en el estudio de tres propiedades sobre las imágenes:

Similitud: Cada uno de los píxeles de un elemento de la imagen tiene valores parecidos para alguna propiedad.

Discontinuidad: Los objetos se destacan del entorno y tienen, por tanto, unos bordes definidos.

Conectividad: Los píxeles pertenecientes al mismo objeto tienen que ser contiguos, es decir, deben estar agrupados.

Estas suposiciones no son fáciles de cumplir pero son fundamentales para una buena segmentación.

La segmentación se clasifica de dos formas:

Segmentación Completa: Obtener regiones disjuntas que correspondan, únicamente, con objetos de la imagen de entrada.

- Suele necesitar cooperación con procesos de alto nivel que usen conocimiento específico del dominio.

Segmentación Parcial: Las regiones que se obtienen no corresponden directamente con los objetos de la imagen (corresponden a zonas homogéneas respecto alguna propiedad).

- La imagen se divide en regiones que son homogéneas respecto a una determinada propiedad como el color, reflectancia, textura, etc.
- En imágenes complejas puede ser necesario obtener regiones homogéneas con solapamiento.
- La segmentación final se realiza con la ayuda de información de alto nivel.

Más que una clasificación, la distinción entre segmentación parcial debe entenderse como una sucesión de objetivos. Todos los sistemas de Tratamiento Computacional de Imágenes tienen como objetivo conseguir una segmentación completa en la cual se obtenga todos los objetos presentes en la imagen y, además cada objeto en una sola región disjunta. Sin embargo, casi siempre hay que conformarse con obtener una aceptable segmentación parcial en la cual se obtiene zonas de imagen que son homogéneas respecto a alguna propiedad. Esas zonas corresponden a objetos de la imagen en un número aceptable de casos, aunque se dan otras circunstancias: un objeto está en dos o más regiones y hay regiones que no se corresponden con nada significativo de la imagen.

El proceso de segmentación se caracteriza generalmente por una vez procesada la imagen, se realiza una segmentación que divide la imagen en regiones. Para clasificar dichas regiones se extrae de ellas unas series de características que permite clasificarlas como se ilustra en la figura 1.

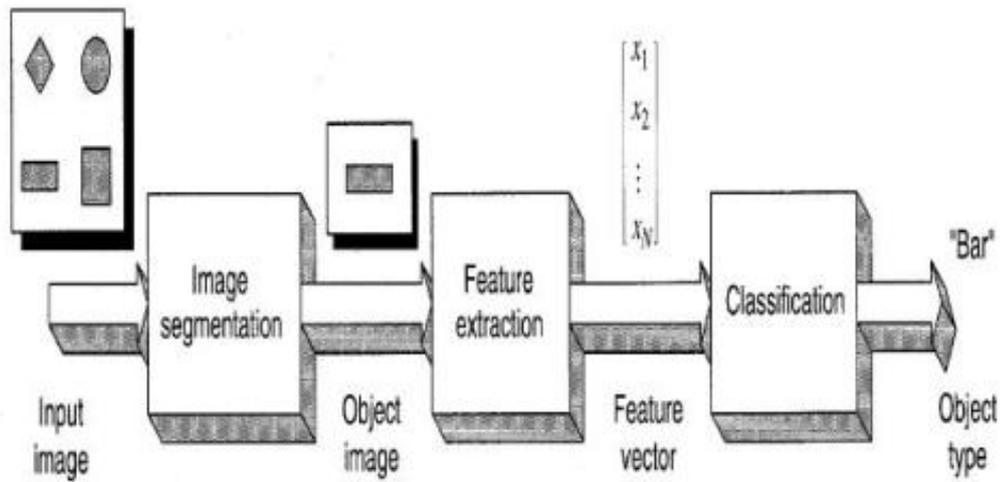


Figura 1.1: Proceso de segmentación.

En cuanto al grado de interacción del usuario en el proceso de segmentación, estos se pueden clasificar en:

Manual: El usuario realiza la segmentación él mismo con la ayuda de una herramienta informática.

Automática: El ordenador realiza todo el proceso de forma automática.

Semi-automática o interactiva: El ordenador realiza el proceso, pero el usuario interviene en determinados momentos sobre el mismo para definir parámetros o corregir resultados. Es el método empleado generalmente.

1.4.1 Técnicas de segmentación.

En función de la propiedad que se utiliza, para realizar la segmentación, se puede establecer una clasificación básica de los métodos de segmentación en:

Agrupamiento de puntos (Métodos de *thresholding*): Se basa en la partición de los elementos de imagen sólo en base a la información sobre sus tonalidades, de acuerdo a que se encuentra entre valores de umbral especificados. En ciertos casos, el análisis de la distribución de intensidades de la imagen permite encontrar de manera óptima los valores característicos que establecen la separación de los objetos de interés, básicamente en zonas correspondiente a valles en el histograma de intensidades de la imagen global o en histogramas locales. Sin embargo, suelen presentarse problemas cuando tales valles no son fácilmente identificables, indicando en este caso que las intensidades de los objetos se superponen. Otro aspecto frecuente es la variación de intensidades incluso dentro de los mismos objetos. En tales casos, un umbral adecuado para un área de imagen puede funcionar pobremente en otros sectores de la misma, por lo cual se han propuesto soluciones mediante umbrales dinámicos.

Métodos basados en regiones: Permiten considerar tanto la similitud de intensidades como la conectividad de los elementos, razón por la cual constituyen una de las alternativas más robustas y a la vez más simples. En un esquema de crecimiento de regiones, los objetos de interés dentro de la imagen se constituyen a partir de puntos iniciales (conocidos como semillas), y evolucionan, mediante la incorporación sucesiva de los puntos próximos y similares a los que ya han sido incluidos en cada región. El proceso de crecimiento continúa de esta manera hasta que no se encuentren más puntos en condiciones de integrarse a las regiones.

Métodos basados en bordes: Realizan la detección de transiciones de tonalidades entre puntos de diferentes componentes. Idealmente, los bordes deberían corresponder a los límites entre objetos homogéneos y superficies de objetos. Si bien se han planteado numerosas variantes, hay dos enfoques principales dentro de esta clase de métodos y son el uso de la primera derivada a través del empleo de máscaras tales como las de Roberts o Sobel (7) o el enfoque propuesto por Canny (8). La generalización isotrópica de la segunda derivada es el Laplaciano (8). Luego de aplicar alguno de los filtros, usualmente prosigue una etapa de conexión de los bordes encontrados para formar curvas o superficies conectadas. Esto es necesario debido a que, excepto en el caso de imágenes simples libres de ruido, la detección de bordes generalmente resulta en un conjunto de elementos de borde fragmentados. La naturaleza derivativa de este enfoque lo hace extremadamente sensible a la presencia de ruido en la imagen

y además los bordes detectados no siempre resultan continuos, dificultando la determinación de las regiones dentro de los límites hallados.

Muchos métodos emplean técnicas combinadas para obtener lo mejor de ambos mundos. Como por ejemplo:

- Técnicas basadas en relajación. Consiste en agrupar píxeles en un conjunto de clases de forma iterativa conforme se relajan las condiciones de pertenencia. Se parte de unas estimaciones de pertenencia iniciales que se van refinando iterativamente de acuerdo a la intensidad del píxel a agrupar y el contexto local (intensidad, textura local. etc.).
- Segmentación en color.
- Segmentación de movimiento.
- Contornos activos.
- Morfología matemática.
- Redes Neuronales Artificiales.
- Clustering

1.4.2 Aplicaciones del proceso de segmentación.

La segmentación de imágenes es uno de los temas fundamentales más estudiados y útiles en el procesamiento y análisis de imágenes, sobre todo porque constituye el paso primero e inevitable para la mayoría de las tareas de análisis cuantitativo de imágenes. Dado al desarrollo de las TIC se puede llevar acabo en diferentes esferas de las ciencias el procesamiento de imágenes. Es valido aclarar que las áreas de aplicación presentadas representan una pequeña parte del total de éstas.

Medicina: La habilidad de detectar estructuras homogéneas dentro de una imagen constituye un aspecto fundamental para la automatización de diverso tipo de aplicaciones en el área de la medicina, permitiendo la inclusión de diferentes criterios de evaluación de similitud de los elementos de la imagen y además posibilita la inclusión de información adicional sobre las características de la imagen y las diferentes estructuras de interés a detectar en la misma. Diversos procedimientos médicos han incorporado estos procedimientos, por ejemplo en el diagnostico de patologías, tratamiento de radioterapia de cirugías. Así como ha permitido el surgimiento de diferentes modalidades como tomografía computada (CT) o imágenes de resonancia magnética (MRI) (9).

Biología: Ha permitido el desarrollo de aplicaciones de gran importancia social. Aumentando el volumen de información y bibliográfico. Se han desarrollado aplicaciones como la evaluación de fotos o imágenes con microscopios ópticos o electrónicos, Observación y análisis de procesos de crecimientos bacterianos, análisis de microestructuras biológicas en varios rangos espectrales.

Filtrado: Mediante la categorización de imágenes que se desea filtrar por ejemplo las imágenes, símbolos, rostros, objetos, en el filtrado de los textos entre otros

Seguridad: Se ha logrado un aumento en la seguridad mediante el reconocimiento de patrones biométricos como son el rostro, las huellas dactilares, el iris del ojo entre otros.

Telefonía: Se han realizado desarrollos orientados a hacer síntesis de voz y de voz visual, con el objetivo de optimizar este tipo de aplicaciones. Se ha desarrollado un sistema capaz de realizar síntesis paramétrica de voz visual en tiempo real (10).

El análisis de textura es útil en muchas aplicaciones de procesamiento de imágenes, como: la visión robótica, la inspección industrial, la interpretación de datos sísmicos y el foto análisis.

Deporte: Se ha aplicado con éxito a imágenes un algoritmo de segmentación, en principio diseñado para imágenes de video, en la limitación de 17 campos de golf construidos en Cataluña en la década de los 90, mediante técnicas de teledetección para establecer sus dimensiones y los usos anteriores del suelo en el que ahora son los campos. (11).

Educación: Actualmente existen sistemas de software diferentes, particularmente atractivos para el procesamiento de imágenes, debido a su rica colección de herramientas, ejemplos de ellos son el Khoros y el Matlab, que constituyen herramientas valiosas para la enseñanza del procesamiento de señales e imágenes digitales en Ingeniería, Biología y Medicina.

1.5 Redes Neuronales Artificiales

Por miles de años la dinámica y comportamiento del sistema nervioso ha sido una incógnita para el ser humano; el estudio del sistema nervioso se remonta a la antigua Grecia, donde Platón y Aristóteles durante el siglo VI (A. de C.), dieron las primeras explicaciones teóricas del funcionamiento del cerebro y el origen de la mente, además de ser el primero en establecer los principios formales del razonamiento deductivo, hasta finales del siglo XIX con el trabajo de Santiago Ramón y Cajal, quien descubriera la unidad funcional y estructural del sistema nervioso:

la neurona(ver Anexo B). Describió prácticamente todos los tipos de neuronas de las distintas regiones del cerebro, cerebelo y la retina de muchas especies animales; pudo demostrar que las neuronas son células individuales, separadas una de otra, y que constituyen la unidad anatómica y funcional del sistema nervioso. Las neuronas están organizadas en redes de intercambio de información permitiendo que el cerebro esté conectado con todos y cada uno de los sentidos y órganos del cuerpo humano.

Las Redes Neuronales Artificiales son redes de elementos simples interconectadas masivamente en paralelo (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico (12). Para el desarrollo se utilizará una red supervisada, concretamente un Perceptrón Multicapa o MLP.

Características de las RNA

Existen cuatro aspectos que caracterizan una red neuronal: su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida, y la forma de representación de estas informaciones (13).

Ventajas de la Redes Neuronales Artificiales

La principal razón de la utilización de las redes neuronales radica en el gran número de aplicaciones con éxitos rotundos. El éxito en las aplicaciones se debe principalmente a las ventajas que las redes neuronales tienen sobre otro tipo de modelos computacionales. A continuación se presentan algunas de estas ventajas :

Aprendizaje Adaptable: La capacidad adaptable es una de las características más atractivas de las redes neuronales. Aprenden a llevar a cabo ciertas tareas mediante un entrenamiento. Como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamiento, no es necesario que se elabore un modelo a priori ni necesitamos especificar funciones de distribución de probabilidad.

Tolerancia a fallos: Las redes neuronales son los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Comparados con los sistemas computacionales tradicionales, los cuales pierden su funcionalidad en cuanto sufren un pequeño error de memoria, en las redes neuronales, si se produce un fallo en un pequeño número de neuronas, aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina. Hay dos aspectos distintos respecto a la tolerancia a fallos: primero, las redes pueden aprender a reconocer los

patrones con ruido, distorsionados o incompletos, esta es una tolerancia a fallos respecto a los datos. Segundo, pueden seguir realizando su función aunque se destruya parte de la red. La razón por la que las redes neuronales son tolerantes a fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento.

Operación en tiempo real: Una de las mayores prioridades es la necesidad de realizar grandes procesos con datos de forma muy rápida. Las redes neuronales se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las redes neuronales puedan operar en un entorno de tiempo real, la necesidad de cambio de los pesos de las conexiones o entrenamiento es mínima. Por tanto, las redes neuronales son una excelente alternativa para el reconocimiento y clasificación de patrones en tiempo real.

Aplicaciones

Las RNA se han aplicado a multitud de campos con excelentes resultados. Las RNA son más que nada una tecnología computacional y por lo tanto puede utilizarse en cualquier campo susceptible de operativizar sus problemas computacionalmente. Dentro del campo del "*Human Computer Interaction*" destaca la creación de interfaces adaptativas, que se modifican en función de las características de los usuarios. Además han conseguido excelentes resultados en generación de reconocedores de voz para interfaces naturales, creación de detectores de movimiento para punteros que sustituyan al ratón, etc.

Algunos de los ejemplos donde se están utilizando las RNA en la actualidad son:

- Reconocimiento de patrones y procesamiento de la información visual.
- Reconocimiento y generación de habla y escritura.
- Control de robots.
- Toma de decisiones y Data Mining.
- Segmentación, compresión y fusión de datos.
- Modelización de datos cerebrales.
- Previsiones meteorológicas.

La lista podría ser interminable. Las RNA se presentan actualmente como una valiosa herramienta capaz de ser aplicada con buenos resultados a cualquier campo, incluidos los Factores Humanos y la Interacción Hombre-ordenador.

1.5.1 Mecanismo de aprendizaje.

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero. Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por tanto se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables o el margen de error es menor o igual al que se ha definido como aceptable.

Un aspecto importante respecto al aprendizaje en las redes neuronales es el conocer cómo se modifican los valores de los pesos; es decir, cuales son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información. Estos criterios determinan lo que se conoce como la regla de aprendizaje de la red. De forma general, se suelen considerar dos tipos de reglas: las que responden a lo que habitualmente se conoce como aprendizaje supervisado, y las correspondientes a un aprendizaje no supervisado. La diferencia fundamental entre ambos tipos de aprendizaje está en la existencia o no de un agente externo (supervisor) que controle el proceso de aprendizaje de la red.

1.5.1.1 Aprendizaje supervisado.

La técnica mayormente utilizada para realizar un aprendizaje supervisado consiste en ajustar los pesos de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, una función de error cometido en la salida. Existen varias formas de calcular el error y luego adaptar los pesos con la corrección correspondiente. Una de las más implementadas utiliza una función que permite cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanto más información se tenga del error cometido, más rápido se puede aprender.

1.5.1.2 Aprendizaje no supervisado.

Las redes con aprendizaje no supervisado no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; por

ello, suele decirse que estas redes son capaces de auto-organizarse. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presentan en su entrada.

En algunos casos, la salida representa el grado de familiaridad o similitud entre los datos que se le están presentando en la entrada y la información que se le ha mostrado hasta entonces (en el pasado). En otro caso podría realizar una agrupación o clasificación de patrones o categorías según su similitud indicando, la salida de la red, a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas.

Finalmente, algunas redes con aprendizaje no supervisado lo que realizan es un mapeo de características, obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada, de tal forma que si se presentan a la red datos similares, siempre sean afectadas neuronas de salida próximas entre sí, en la misma zona del mapa.

1.5.2 MLP

El perceptrón multicapa es una red neuronal artificial formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple). El perceptrón multicapa puede ser totalmente o localmente conectado.

En el primer caso cada salida de una neurona de la capa i es entrada de todas las neuronas de la capa $i + 1$, mientras que el segundo, cada neurona de la capa i es entrada de una serie de neuronas (región) de la capa $i + 1$. Las capas pueden clasificarse en tres tipos:

Capa de entrada: Constituida por aquellas neuronas que introducen los patrones de entrada en la red. En estas neuronas no se produce procesamiento.

Capas ocultas: Formada por aquellas neuronas cuyas entradas provienen de capas anteriores y las salidas pasan a neuronas de capas posteriores.

Capa de salida: Neuronas cuyos valores de salida se corresponden con las salidas de toda la red. La propagación hacia atrás (también conocido como retropropagación del error o regla delta generalizada), es un algoritmo utilizado en el entrenamiento de estas redes, por ello, el perceptrón multicapa también es conocido como red de retropropagación.

Características

El *Multiplayer Perceptrón* se caracteriza por presentar una no-linealidad en la salida, capas de neuronas ocultas y un alto grado de conectividad. Es de entrenamiento supervisado. Utiliza el algoritmo de retropropagación del error, que está basado en la regla de aprendizaje por corrección de error, considerada como una generalización del algoritmo de los cuadrados mínimos (LMS), utilizado en filtrado adaptivo mediante redes lineales simples.

Su operación consta de dos fases, una directa y una inversa o de retroceso. En la fase directa, se ingresa el patrón de actividad en la capa de entrada de la red (vector de entrada), que recorre todas las capas subsiguientes. Se obtiene la respuesta real de la red en la capa de salida. En esta fase, los pesos sinápticos de la red permanecen fijos. En la fase inversa, los pesos sinápticos son ajustados de acuerdo con la regla de corrección del error. Esta regla, conocida como método de Levenberg-Marquardt, minimiza el cuadrado de las diferencias entre la respuesta o salida deseada y la salida real de la red.

Limitaciones

El Perceptrón Multicapa no extrapola bien, es decir, si la red se entrena mal o de manera insuficiente, las salidas pueden ser imprecisas. La existencia de mínimos locales en la función de error dificulta considerablemente el entrenamiento, pues una vez alcanzado un mínimo el entrenamiento se detiene aunque no se haya alcanzado la tasa de convergencia fijada.

Cuando caemos en un mínimo local sin satisfacer el porcentaje de error permitido se puede considerar: cambiar la topología de la red (número de capas y número de neuronas), comenzar el entrenamiento con unos pesos iniciales diferentes, modificar los parámetros de aprendizaje, modificar el conjunto de entrenamiento o presentar los patrones en otro orden.

1.5.3 Mapas auto-organizados (SOM)

Los modelos de Mapas Autoorganizado (SOM) fueron introducidas por T. Kohonen y son un tipo especial de redes neuronales artificiales de aprendizaje no supervisado que ha sido exitosamente aplicado como una herramienta de *Data Mining*.

Las ventajas de los mapas autoorganizado radican en:

- ❖ Que son capaces de preservar la topología del espacio de los datos.
- ❖ Proyectan datos altamente dimensionales a un esquema de representación de baja dimensión

- ❖Tienen la habilidad de encontrar similitudes en los datos.

El éxito de las redes SOM se debe a su propiedad especial de crear de forma efectiva representaciones internas especialmente organizadas de varias características de las señales de entrada y sus abstracciones. Las SOM quantizan el espacio de los datos formado por los datos de entrenamiento y simultáneamente desarrollan una proyección de los datos a una grilla regular de baja dimensión, la que preserva la topología del espacio de entrada. La grilla puede ser utilizada de manera eficiente en visualizaciones. Las SOM implementan un mapa ordenado de dimensionalidad reducida de los datos que respeta la función de densidad de probabilidad que subyace en el comportamiento de los datos.

Las SOM pueden ser descritas formalmente como un mapeamiento no-lineal, ordenado, suave de los datos de entrada altamente dimensionales hacia los elementos de un arreglo regular de baja dimensión.

El algoritmo de las SOM consiste en un procedimiento iterativo capaz de representar la estructura topológica del espacio de entrada (discreto o continuo) por medio de un conjunto discreto de prototipos de *vectores de peso* las que son asociadas a neuronas de la red. Las SOM mapean los patrones de entradas vecinos a neuronas vecinas.

Durante la etapa de entrenamiento, se presenta a la red un conjunto de informaciones de entrada (vectores de entrenamiento) para que ésta establezca en función de la semejanza entre los datos las diferentes categorías (una por neurona de salida), que servirán durante la fase de funcionamiento para realizar clasificaciones de nuevos datos que se presenten a la red. Los valores finales de los pesos de las conexiones entre cada neurona de la capa de salida con las de entrada se corresponderán con los valores de los componentes del vector de aprendizaje que consigue activar la neurona correspondiente. En el caso de existir más patrones de entrenamiento que neuronas de salida, más de uno deberá asociarse con la misma neurona, es decir pertenecerán a la misma clase.

En este modelo el aprendizaje no concluye después de presentarle una vez todos los patrones de entrada, sino que habrá que repetir el proceso varias veces para refinar el mapa topológico de salida, de tal forma que cuantas más veces se presenten los datos, tanto más se reducirán las zonas de neuronas que se deben activar ante entradas parecidas, consiguiendo que la red pueda realizar una clasificación más selectiva.

1.5.4 Learning Vector Quantization

Esta red es un híbrido que emplea tanto aprendizaje no supervisado, como aprendizaje supervisado para clasificación de patrones. El algoritmo LVQ es un método de clasificación basado en el aprendizaje competitivo neuronal, que permite reforzar positivamente (premiando) o negativamente (castigando) los pesos de las conexiones, dependiendo de que la clasificación haya sido realizada correcta o incorrectamente. LVQ usa un aprendizaje supervisado para definir regiones de clases en el espacio de los datos de entrada. Con este propósito, un subconjunto de vectores de similitud etiquetados de forma similar forman una región de una clase.

La arquitectura de la red LVQ presenta dos capas con N neuronas de entrada y M de salida. Cada una de las N neuronas de entrada se conecta a las M de salida hacia delante (feedforward). Entre las neuronas de la capa de salida, existen conexiones laterales de inhibición (peso negativo). El valor que se asigne a los pesos de las conexiones feedforward entre las capas de entrada y salida durante el proceso de aprendizaje de la red, va a depender de esta interacción lateral. El aprendizaje es de tipo competitivo. Las neuronas de la salida compiten por activarse y sólo una de ellas permanece activa ante una determinada información de entrada a la red. Esta neurona se denomina prevalente o vencedora, y en función de ella se ajustan los pesos de las conexiones.

Características

Las características más relevantes de este método son la sencillez de las heurísticas empleadas y la rapidez de cálculo.

1.5.5 Support Vector Machines

Los SVM resultan ser sistemas de aprendizaje universal que, en su forma más simple, son capaces de encontrar una función lineal discriminante que separe el espacio de representación en regiones correspondientes a cada una de las clases consideradas. Como en todo aprendizaje supervisado, la entrada al sistema es un conjunto de ejemplos de entrenamiento. Si el conjunto de entrenamiento es linealmente separable, el SVM encuentra un hiperplano que maximiza la distancia euclídea a los ejemplos de entrenamiento más cercanos. En términos geométricos, SVM puede ser visto como el intento de encontrar una superficie que separe a los ejemplos positivos de los negativos por el margen más amplio posible (2005).

Cabe resaltar que la mejor superficie de decisión es determinada únicamente por un conjunto pequeño de ejemplos de entrenamiento, llamados vectores de soporte. En el caso de que el conjunto no pueda ser linealmente separable, se mide el error en el entrenamiento. Así, el cálculo

de este hiperplano se reduce a un problema de optimización de errores en el que se aplican restricciones. Las restricciones fuerzan a que todos los ejemplos de entrenamiento sean clasificados correctamente por encima de un error mínimo.

Característica

Una característica fundamental de estos clasificadores es que son independientes de la dimensión del espacio de características. Así, la medida de la complejidad no depende del número de rasgos, sino de cómo puedan separarse los datos. Esto significa que siempre que los datos sean separables, podría generalizarse la presencia de muchos rasgos en una determinada categoría.

Aplicación

Las máquinas de vectores de soporte han mostrado conseguir buen desempeño de generalización sobre una amplia variedad de problemas de clasificación, destacando recientemente en problemas de clasificación de textos, donde se aprecia que SVM tiende a minimizar el error de generalización.

1.6 Tecnología a utilizar.

El uso apropiado de herramientas puede mejorar la efectividad y eficiencia con la que se desarrollan los softwares; al mismo tiempo, su uso beneficia la calidad del mismo. La automatización puede aumentar los beneficios obtenidos con el uso de las herramientas al disminuir el tiempo necesario para terminar un proyecto, al evitar el tedio asociado con ciertas tareas y al seguir procedimientos consistentes.

1.6.1 Lenguaje de programación.

Un lenguaje de programación es un conjunto limitado de palabras y símbolos que representan procedimientos, cálculos, decisiones y otras operaciones que puede ejecutar una computadora. Los mismos se pueden dividir en dos grupos fundamentales: Descriptivo (Declarativo) o Prescriptivo (Imperativo). Existen otras agrupaciones para los lenguajes de programación y una de las más comunes es su agrupación por lenguajes de alto y bajo nivel.

Hoy día existen numerosos lenguajes de programación, cada uno de ellos fueron creado para un fin determinado, por los que algunos son mejores que otros a la hora de realizar una tarea

específica. Elegir un lenguaje de programación para el desarrollo de una aplicación no es tarea sencilla ya que hay que tener en cuenta muchos aspectos para saber a priori si el seleccionado es el adecuado para resolver el problema dado. Una de las posibles causas del fracaso de un proyecto es una mala selección del lenguaje de programación que se utilizará.

El presente trabajo propone para su desarrollo al lenguaje de programación C++ elegido fundamentalmente por ser uno de los más usados a escala mundial (14), su abundante documentación, cuenta con muchas bibliotecas para solucionar diversos problemas, lo que agiliza el desarrollo, y por la experiencia de los desarrolladores en este lenguaje, lo que permite un desarrollo mas fluido.

Breve historia de C++

El lenguaje C++ esta basado en el lenguaje de programación C, el cual a su vez está basado en dos lenguajes muy primitivos (BCPL y B). Sus fundadores fueron Martin Richards² (1976) y Ken Thompson³ (1970). Dos años más tarde de la creación del lenguaje B Dennis Ritchie implemento el diseño de BCPL, B y creo C; que se dió a conocer por ser el lenguaje de programación de desarrollo de UNIX.

A principio de la década del 80, Bjarne Stroustrup⁴ empezó a desarrollar C++, que recibiría formalmente su nombre a finales de 1983. En octubre de 1985, apareció la primera divulgación comercial del lenguaje y la primera edición del libro *The C++ Programming Language*⁵.

Desde ese momento, C y C++ se han usado para la creación de sistemas operativos, por lo que su popularidad ha ido creciendo a lo largo de los años, y actualmente es uno de los lenguajes de programación más usados por los programadores.

Características del lenguaje

C++ cuenta con características importantes que sirvieron de apoyo para su selección como lenguaje de programación para el desarrollo de la aplicación, estas características se describen a continuación:

²Martin Richards: Programador informático y teórico de la informática británica. Es conocido por haber desarrollado el lenguaje informático BCPL.

³ Ken Thompson: Pionero de en las ciencias de la computación. Se le adjudica la creación de Unix junto a Dennis Ritchie.

⁴ <http://www.research.att.com/~bs>.

⁵ Escrito por el propio Bjarne Stroustrup.

Alto nivel: Este tipo de lenguaje permite al programador abstraerse de la arquitectura y funcionamiento del hardware.

Orientado a objetos: La posibilidad de orientar la programación a objetos permite al programador diseñar aplicaciones desde un punto de vista más cercano a la vida real. Además, permite la reutilización del código de una manera más lógica y productiva.

Portabilidad: Un código escrito en C++ puede ser compilado en muchos sistemas operativos⁶.

Brevedad: El código escrito en C++ es muy corto en comparación con otros lenguajes, sobretodo porque en este lenguaje es preferible el uso de caracteres especiales que las “palabras clave”.

Programación modular: Un cuerpo de aplicación en C++ puede estar hecho con varios ficheros de código fuente que son compilados por separado y después unidos. Además, esta característica permite unir código en C++ con código producido en otros lenguajes de programación como Ensamblador o el propio C.

Velocidad: El código resultante de una compilación en C++ es muy eficiente, gracias a su capacidad de actuar como lenguaje de alto y bajo nivel y a la reducida medida del lenguaje.

1.6.2 Bibliotecas disponibles.

Dado que el software va ganando en complejidad cada día, se hace necesaria la búsqueda y utilización de herramientas que agilicen el desarrollo de software, como son las bibliotecas, paquetes de clases, frameworks, módulos, etc. C++ es un lenguaje con una gran historia por lo que cuenta con muchas bibliotecas para la solución de disímiles problemas. Para poder resolver el problema planteado en este trabajo se necesita una biblioteca capaz de manejar imágenes. Mediante el estudio realizado se descubrió la existencia de una biblioteca con excelentes características para implementar redes neuronales multicapas llamada FANN. Además se utiliza la opencv para el manejo de las imágenes.

⁶ Para esto es necesario usar el estándar ANSI para C++. En ocasiones deben hacerse pequeños cambios.

1.6.2.1 Librería FANN

La Fast Artificial Neural Network Library (FANN) es una librería desarrollada bajo software libre que proporciona una interfaz en C para implementar redes neuronales multicapa. Esta librería fue desarrollada en el 2003 por Steffen Niessen en la Universidad de Copenhague. Entre sus principales características esta que, es un librería fácil de utilizar, bien documentada y funciona en cualquiera de las principales plataforma. Aparte de C, existen enlaces para los lenguajes de programación más usados.

1.6.2.2 Librería OpenCV

Las siglas OpenCV provienen de los términos anglosajones “*Open Source Computer Vision Library*”. Por lo tanto, OpenCV es una librería de tratamiento de imágenes, destinada principalmente a aplicaciones de visión por computador en tiempo real.

- Es una librería de código abierto, gratuita, tanto para uso comercias como no comercial.
- Es una librería multiplataforma, rápida, de fácil uso y en continuo desarrollo.
- Implementa una amplia variedad de funciones en C/C++ para el tratamiento de imágenes.

Dado a las características que pose y las facilidades que brinda, además de su amplia documentación de adopta la librería OpenCV como base para el tratamiento de las imágenes.

1.6.3 IDE de desarrollo

Las ventajas principales de un IDE radican en la automatización de algunas tareas asociadas al desarrollo de aplicaciones y al poseer algunas funcionalidades que facilitan el desarrollo de software disminuyendo así el tiempo de construcción. La selección un IDE para el desarrollo de aplicaciones en C++ en el sistema operativo GNU/Linux se hace hasta cierto punto difícil, ya que ninguno de los IDEs existentes completa código lo cual es una de las características que más agiliza el desarrollo de software al no tener que memorizar y teclear todas las funciones existentes en otras clases. Finalmente se encontró uno que si hace esta tarea y es el caso de Eclipse con un plugging llamado CDT el cual tiene las siguientes características:

Editor de C/C++ (Funcionalidades básicas, resaltamiento de sintaxis, completamiento de código, etc.)

Depurador C/C++ (Usando GDB)

Lanzador C/C++ (Lanzadores y aplicaciones externas)

Parser⁷

⁷ Programa analizador sintáctico.

Herramienta de búsqueda

Proveedor de asistencia de contenido.

Generador de Makefile⁸

Adicionalmente a estas características mencionadas, pertenecientes al pluging CDT; se tienen otras ventajas del Eclipse, como IDE de desarrollo, las cuales merecen ser mencionadas:

Posee una excelente integración con SVN⁹

Posee un potente administrador de proyecto

Es multiplataforma

Altamente configurable

La única desventaja hasta el momento es su gran consumo de recurso. Para su uso se recomienda un ordenador con al menos 512 MB de RAM con un procesador a una velocidad de 2.4 GHz o superior.

1.6.4 Herramienta Case.

Visual Paradigm.

Visual Paradigm es una herramienta para UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos.

Visual Paradigm tampoco constituye la solución al problema planteado ya que para poder utilizarse es necesario introducir una licencia. Las licencias de *Visual Paradigm* son *Enterprise*, *Professional*, *Standar*, *Modeler*, *Personal* y *Community*. En dependencia de la licencia proporcionada la aplicación permite al usuario acceder a todas sus funciones o no. Las opciones de diseño de interfaces de usuario necesitan la licencia empresarial para poder ser utilizadas, de lo contrario esta opción ni siquiera es mostrada y si no se proporciona una licencia la aplicación se cierra y no permite su utilización.

⁸ Un fichero de configuración usado por el programa **make** definiendo la ubicación de código fuente, como deberá ser compilado y vinculado para crear un programa ejecutable.

⁹Control de versiones SubVention.

1.6.5 Lenguaje de modelación.

UML

UML, Lenguaje Unificado de Modelado (*Unified Modeling Language*, en inglés). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Fue desarrollado en *Rational Software Corporation* por Grady Booch, James Rumbaugh e Ivar Jacobson con contribuciones de otros metodólogos líderes, vendedores de software y muchos usuarios. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML ofrece funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos. En la actualidad una de las mejores y más utilizadas en el mercado mundial que provee el modelado visual basado en UML es Rational Rose por lo que es la usada en la modelación de este proyecto.

1.6.6 Otras Herramientas.

Es justo mencionar algunas herramientas auxiliares utilizadas para la confección del presente documento que, aunque no formaron parte del desarrollo del software, si contribuyeron a disminuir el tiempo y aumentar la calidad del trabajo; ellas son:

LTEX: Es un sistema tipográfico para generar documentos científicos de alta calidad, lo cual permitió centrarse en el documento dejando a este la tarea de generar un documento que cumpliera con la calidad requerida para una tesis de grado.

Kile: Es un IDE para LTEX el cual posibilitó un desarrollo fluido del documento.

Dia: Es un software de dibujo utilizado para generar las figuras del presente documento.

Gimp: Es un software de edición de imágenes. Usado en el desarrollo del documento para editar las imágenes generadas por Dia y mejorar su estética. Usando en el desarrollo del software para observar el histograma de una imagen, selección de umbral, cortar una región de una imagen entre otras tareas.

1.7 Metodología de desarrollo.

Una metodología de desarrollo de software es un proceso, el cual define qué, cuándo, y cómo alcanzar un determinado objetivo. Este debería ser capaz de servir como guía para todos los participantes (clientes, usuarios, desarrolladores y directores ejecutivos), evolucionar durante muchos años permitiendo limitar su alcance en un momento del tiempo dado a las realidades de la tecnología, herramientas, personas y patrones de organización.

Ventajas de tener una metodología (15).

Mejora de los procesos de desarrollo.

- Todas las personas del proyecto trabajan bajo un marco común.
- Estandarización de conceptos, actividades y nomenclaturas.
- Actividades de desarrollo apoyadas por procedimientos y guías.
- Resultados del desarrollo predecibles.
- Uso de herramientas de Ingeniería Software.
- Planificación de actividades en base a un conjunto de tareas definidas y a la experiencia en otros proyectos.
- Recopilación de mejores prácticas para proyectos futuros.

Mejora de los productos.

- Se asegura que los productos cumplen con los objetivos de calidad propuestos.
- Detección temprana de errores.
- Se garantiza la trazabilidad de los productos a lo largo del desarrollo.

Mejora de las relaciones con el cliente.

- El cliente percibe el orden en nuestros procesos.
- Facilita al cliente el seguimiento de la evolución del Proyecto.
- Se establecen mecanismos para asegurar que los productos desarrollados cumplen con las expectativas del cliente.

Metodologías Ágiles

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software conocidos anteriormente como metodologías livianas, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil como la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Las Metodologías Ágiles se basan en los siguientes principios:

- Realizar entregas cortas en el tiempo y continuas.
- Dar la bienvenida a los cambios.
- Entregas periódicas y frecuentes que funcionen.
- Los clientes forman parte del equipo de desarrollo.
- Equipo con individuos motivados. Darles para ello el ambiente, apoyo y confianza.
- La comunicación directa es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo. Intenta evitar el teléfono, correos electrónicos, fax, etc.
- La medida principal de progreso es el software que funciona.
- Desarrollo sostenible. Es indispensable que exista paz y armonía en el equipo para que el proyecto tenga éxito.
- Buen diseño y calidad técnica.
- La simplicidad es algo básico.
- Equipos auto-organizados.
- El equipo debe realizar reflexiones periódicamente para plantearse cómo llegar a ser más efectivo.

Entre las más conocidas están:

- XP (Extreme Programming).
- SCRUM.
- Metodología Crystal.
- Dynamic Systems Development Method (DSDM).
- Adaptive Software Development (ASD).
- Feature Driven Development (FDD).
- Lean Development (LD).

- BUP.
- AUP.
- SXP.

Dada a las características y necesidades del proyecto, así como a los resultados que se han obtenido con el uso de esta metodología en varios proyectos de nuestra facultad, como las ventajas y características que ofrece esta metodología, se decidió el uso de la metodología ágil SXP y dado que surgió de una investigación en la Universidad de las Ciencias Informática, por la ingeniera Gladys Marsi Peñalver Romero siendo una metodología de desarrollo de software que toma los mejores aspectos de XP y Scrum, adaptándolo a las características de los proyectos productivos de la facultad 10.

SXP

La metodología a usar es SXP ya que es la metodología usada por el proyecto actualmente. Es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto.

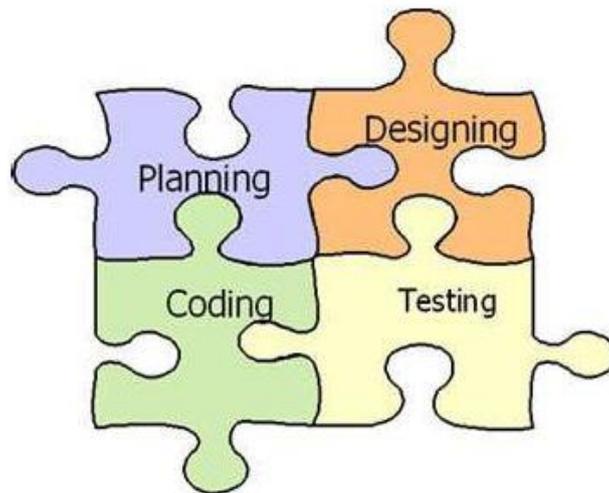


Figura 1.2: Fases de XP.

Las principales **características** son:

- ❖ Comunicación, entre usuario y desarrollo.
- ❖ Simplicidad, al desarrollar y codificar los módulos del sistema.
- ❖ Realimentación (Feedback), concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Esta metodología se basa en:

1. **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
2. **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
3. **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Que propone esta metodología?

- ❖ Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- ❖ El manejo del cambio se convierte en parte sustantiva del proceso.

- ❖ El costo del cambio no depende de la fase o etapa.
- ❖ No introduce funcionalidades antes que sean necesarias.
- ❖ El cliente o el usuario se convierten en miembro del equipo.

Derechos del Cliente:

- ❖ Decidir que se implementa
- ❖ Saber el estado real y el progreso del proyecto. Añadir, cambiar o quitar requerimientos en cualquier momento
- ❖ Obtener lo máximo de cada semana de trabajo
- ❖ Obtener un sistema funcionando cada 3 o 4 meses

Derechos del Desarrollador:

- ❖ Decidir como se implementan los procesos
- ❖ Crear el sistema con la mejor calidad posible
- ❖ Pedir al cliente en cualquier momento aclaraciones de los requerimientos
- ❖ Estimar el esfuerzo para implementar el sistema
- ❖ Cambiar los requerimientos en base a nuevos descubrimientos (Mendoza, M. A. S. 2004).

Fases de Desarrollo

Planificación-Definición: donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

Desarrollo: es donde se realiza la implementación del sistema hasta que este listo para ser entregado.

Entrega: puesta en marcha.

Mantenimiento: es donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. La metodología en un principio se llamaba MA-GMPR-UR2 pero como el nombre no era comercial se le dio el nuevo nombre de SXP.

1.7 Conclusiones

En el capítulo se definieron una serie de conceptos, características relacionadas con el tema tratado. Luego de analizar el proceso de segmentación se evidencia que desempeña un rol importante en el tratamiento de imágenes y en diversas áreas de aplicación. Se abordaron los principios, características, necesidades, objetivos de la metodología y herramientas a utilizar, sentándose las bases para el desarrollo de una aplicación que sea capaz de clasificar de forma automática los píxeles en piel o no piel. A continuación presentamos algunos aspectos importantes definidos en el presente Capítulo.

Tecnologías.

Se adoptaron las siguientes metodologías para el desarrollo de este trabajo:

- Como lenguaje de programación C++.
- Como bibliotecas FANN y OpenCV.
- Como IDE de desarrollo Eclipse con el plugin CDT.
- Como herramienta Case Visual- Paradigm.

2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

2.1 Introducción.

La segmentación es una importante tarea hacia un sistema de análisis de imagen de alto nivel. Ésta permite la cuantificación y visualización de los objetos de interés y de su calidad depende en gran medida el resultado final de la interpretación automática de una escena. Para guiar el desarrollo del presente trabajo es necesario comprender su contexto e identificar las condiciones o capacidades que debe cumplir. En este capítulo se define las características y se describe del algoritmo, realizándose el modelo de dominio con el objetivo. Se especifican sus requisitos funcionales y no funcionales y se expresa la dinámica del proyecto en forma de historias de usuarios.

2.2 Descripción del problema.

Actualmente para llevar a cabo el proceso de segmentación de imágenes por el modulo MCAIP se usa el modelo de segmentación propuesto por Jones and Regh. Este modelo presenta una baja efectividad a la hora de clasificar los píxeles en piel o no piel. Para evitar esto y tratar de aumentar la efectividad del modulo MCAIP a la hora de clasificar las imágenes se a desarrollado un algoritmo basado en Redes Neuronales Artificiales (RNA).

2.3 Modelo de dominio.

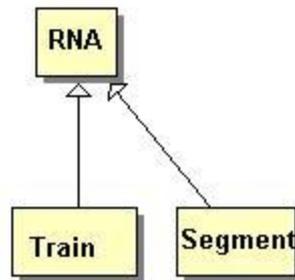
Por la relativa simplicidad del entorno donde está enmarcada la solución, no es necesario profundizar a través de un modelo de negocio. Es suficiente con un Modelo de Dominio para capturar los principales elementos envueltos en la problemática.

El modelo de Dominio (o Conceptual) es una representación visual de los conceptos u objetos del mundo real significativo para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real y no de los componentes de software (16).

Identificar muchos objetos o conceptos forma parte de una investigación del problema. El lenguaje UML contiene la notación en diagramas de estructura estática que explican gráficamente los Modelos de Dominio. El paso fundamental de un análisis orientado a objetos es descomponer el problema en conceptos individuales; ya que es una representación de conceptos en un dominio

del problema se ilustró con un grupo de diagramas de estructura estática donde no se define ninguna operación. Puede mostrarnos:

- conceptos
- asociaciones de conceptos
- atributos de conceptos



2.1 Modelo de dominio del problema

2.4 Solución propuesta.

Se propone un algoritmo detector del color de la piel utilizando RNA. Este permitirá que el proceso de clasificación de los píxeles en piel o no piel sea más eficiente, trayendo como consecuencia que la segmentación y posteriormente la clasificación de las imágenes tenga más efectividad, propiciando una mayor calidad en el filtrado de contenido del proyecto.

Este no contará con una IGU dado que en primer lugar no necesita una interfaz gráfica; en segundo lugar su comportamiento es gestionado mediante un fichero de configuración el cual es simple. El fichero de configuración permitirá configurar el software para satisfacer las necesidades de los usuarios de forma sencilla y flexible sin cambiar una línea de código dentro del sistema, permitiendo que sea posible realizar cambios en la topología de la RNA, así como reestructurar todo el proceso de entrenamiento y cambiar las ubicación de los datos de entrada y salida. Además la ubicación de este fichero no influirá en absoluto y traerá consigo que el sistema sea portable. La propuesta de este se muestra en el anexo.

El sistema cuenta con dos etapas la primera es la de entrenamiento, donde dado una muestra de imágenes se entrena la red introduciéndole los vectores de entrada y especificando el resultado que debe dar la RNA de salida, creándose un fichero de entrenamiento. Luego se puede pasar a segmentar donde dada una imagen el sistema debe ser capaz de clasificar los píxeles, cargando el fichero que genera la Red Neuronal en la etapa de entrenamiento.

En sentido general debe proporcionar un adecuado margen de error para poder aplicarse en MOCIC, ya que si este posee un alto margen de error las clasificaciones de los píxeles serán erróneas.

Aunque este trabajo está dirigido a un algoritmo para MOCIC, éste deberá poseer una autonomía total, permitiendo así que pueda ser reutilizado por otros usuarios que deseen esta funcionalidad en sus sistemas.

2.5 Segmentación.

El objetivo principal de este paso es separa las partes principales de una imagen. La segmentación autónoma¹⁰ es uno de los procesos más difíciles en el procesamiento de imagen. Por una parte una segmentación adecuada facilitará mucho la solución del problema y una segmentación errónea conducirá a fallos. Una completa segmentación de una imagen R comprende la identificación de un grupo finito de regiones ($R_1; R_2; R_3; \dots; R_N$) tal que:

$$R = R_1 \cup R_2 \cup \dots \cup R_N$$

$$R_i \cap R_j = \emptyset, \forall i \neq j$$

$$P(R_i) = \text{Verdadero}, \forall i$$

$$P(R_i \cup R_j) = \text{Falso}, i \neq j$$

La segmentación en nuestro sistema es realizada a partir del color de la piel humana dado que las imágenes pornográficas muestran generalmente personas desnudas. Estas imágenes muestran mucha piel y debido a esto el color de la piel es una característica básica usada para la categorización de tales imágenes. Una desventaja que puede traer escoger esta característica como primaria es que no se podrá trabajar con imágenes en niveles de grises, sin embargo las imágenes del dominio que nos ocupa raramente están en blanco y negro.

2.5.1 Modelado del color de la piel.

Existen numerosos trabajos relacionados con el modelado del color de la piel, que abarcan definición implícita, métodos paramétricos y no paramétricos. También se usan diferentes espacios de color, siendo los más usados el RGB, RGB normalizado y HSV (17). El objetivo de ésta tarea es etiquetar cada píxel de la imagen con dos valores posibles: 1 si el píxel posee color de la piel, 0

¹⁰ Sin necesidad de la intervención de un humano.

si el píxel no posee color de la piel. Como resultado se obtiene una imagen binaria donde se distinguen claramente las regiones correspondientes a la piel humana. Un método para construir un clasificador de piel consiste en definir específicamente los límites de la región correspondiente al color de la piel, a través de unas reglas numéricas. Este método es aplicable en diferentes espacios de color. Se empezó este trabajo usando un modelo explícito del color de la piel en el espacio RGB, que clasifica un píxel según el siguiente criterio:

píxel [R; G; B] tiene color de la piel si:

$$R > 95 \text{ y } G > 40 \text{ y } B > 20 \text{ y}$$

$$\text{Max } \{R; G; B\} - \text{min } \{R; G; B\} > 15$$

$$\text{y } |R - G| > 15 \text{ y } R > G \text{ y } R > B$$

Según algunos estudios, la distribución de colores bajo distintas condiciones del entorno dependen más de la intensidad que del color en si mismo, por esta razón se normalizó el modelo con respecto a la intensidad (R+G+B) obteniendo el modelo:

$$r = R / (R+G+B) \quad g = G / (R+G+B) \quad b = B / (R+G+B)$$

2.6 Perceptrón multicapa.

Las Redes Neurales Artificiales (*Artificial Neural Networks* - ANN) son sistemas de procesamiento de información que poseen la capacidad de aprender, memorizar y generalizar a partir de un conjunto de datos y de su estructura. Se crearon como modelos muy abstractos y simples del sistema nervioso humano para comprender el proceso cognitivo, luego se analizaron como modelos matemáticos capaces de aprender de un conjunto de datos y con la llegada de los computadores se convirtieron en una poderosa herramienta computacional.

Aunque existen muchos modelos y representaciones de Redes Neuronales Artificiales (RNA), cada una de ellas tienen cuatro atributos $\langle N_c, W, \sigma, \delta \rangle$, donde N_c es un número finito de neuronas fuertemente interconectadas con las salidas $n_1, n_2, n_3, \dots, n_k$; W denota un número finito de pesos los cuales representan la fuerza w_{ij} de la interconexión entre las neuronas n_i y n_j ; σ es la regla de propagación que muestra como las señales de entradas a la neurona n se

propagan a través de ella.

Una regla típica de propagación puede ser $\sigma(i) = \sum n_i w_{ij}$ y δ es una función de activación la cual es usualmente una función no lineal como la función sigmoide¹¹.

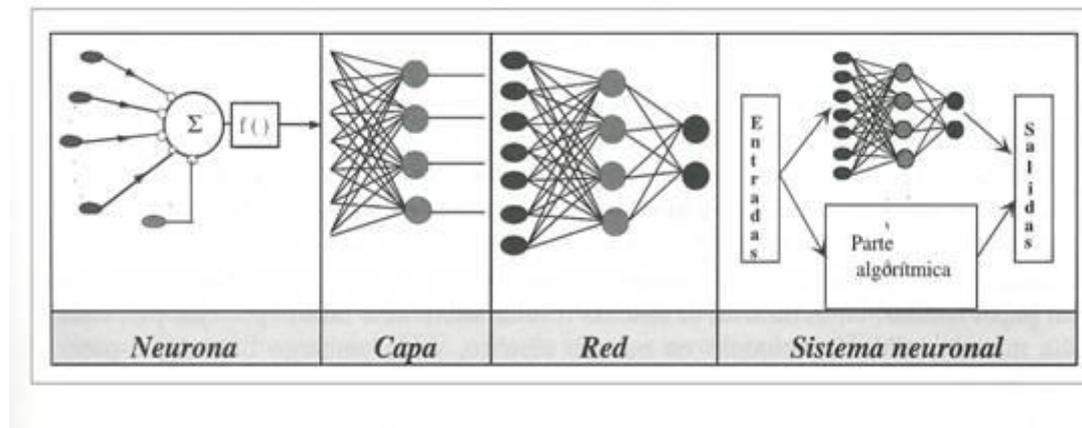


Figura. 2.2 Componentes de una red neuronal

El modelo más popular de redes neuronales es el MLP, el cual es una extensión de un perceptrón de una sola capa propuesto por Rosenblat. MLP, en general, son redes retroalimentadas, teniendo distintas entradas, salida y una capa oculta. La arquitectura de la red neuronal MLP con error de propagación hacia atrás es mostrada en el [Anexo B](#).

2.6.1 Arquitectura de la MLP.

En un problema de M -clases donde los patrones son N -dimensionales, la capa de entrada consta de N neuronas y la capa de salida consta de M neuronas. Puede haber una o más capas ocultas. Se considera el caso de una sola capa oculta, la cual es extensiva a cualquier número de estas. Las capas ocultas constan de p neuronas. La salida de cada neurona en la capa de entrada es conectada a las neuronas de la capa oculta. Ningún cálculo es ejecutado en las neuronas de la capa de entrada. Las neuronas de la capa oculta agrupan las entradas, pasando estas por la función no lineal sigmoide y hace múltiples conexiones con las neuronas de la capa de salida.

¹¹ <http://mathworld.wolfram.com/SigmoidFunction.html>

Muchos autores explican que para un adecuado funcionamiento de las MLP es necesario elegir una buena topología. Algunos determinan que la capa oculta en un comienzo debe tener un valor igual a la semisuma de los valores de entrada y de salida y posteriormente este valor debe ir aumentándose hasta determinar la arquitectura donde la RNA posea mayor efectividad a la hora de clasificar. Otros proponen una topología en forma de cono, donde la cantidad de neuronas en la capa oculta por lo mínimo debe exceder en dos a la capa de entrada. Dado esto se decidió probar ambas topologías en el desarrollo de la solución.

El problema descrito en el presente documento consta de dos clases, piel y no piel. En la solución la RNA. La primera topología consta de tres características, la Red Neuronal tendrá tres neuronas en su capa de entrada correspondiente a los valores del modelo RGB normalizados, y de acuerdo a lo planteado anteriormente una capa oculta de dos neuronas y en su capa de salida tendrá dos neuronas, ya que son dos clases con las que se cuenta. La arquitectura de la red es mostrada en la figura 4.

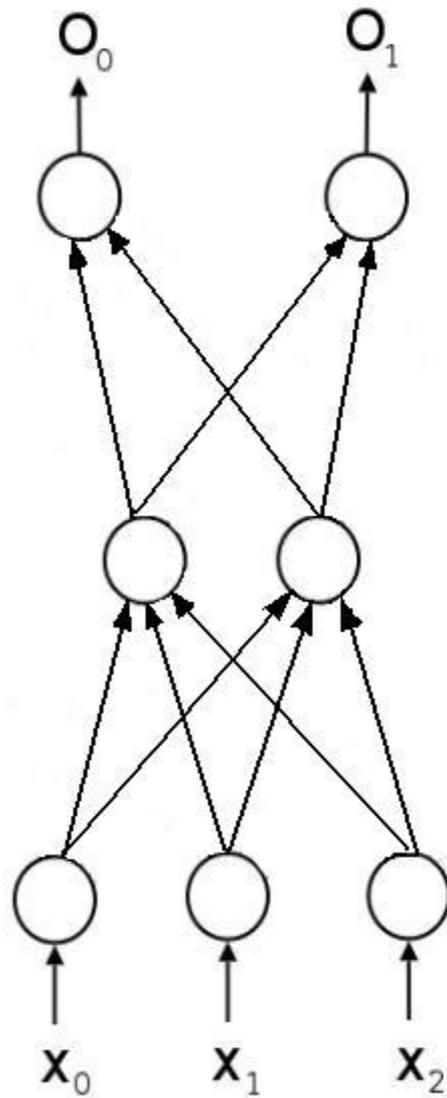


Figura 2.3 Arquitectura de MLP con modelo RGB normalizado

CAPÍTULO 2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

En el segundo caso se presenta una topología donde como entrada se tiene a los valores del RGB binarios, es decir la capa de entrada presenta un valor de 24 neuronas, ya que cada valor de RGB representa 8 bit. En la capa oculta posee 13 neuronas y como salida se tendrá dos neuronas. Para una mayor comprensión ver la figura.

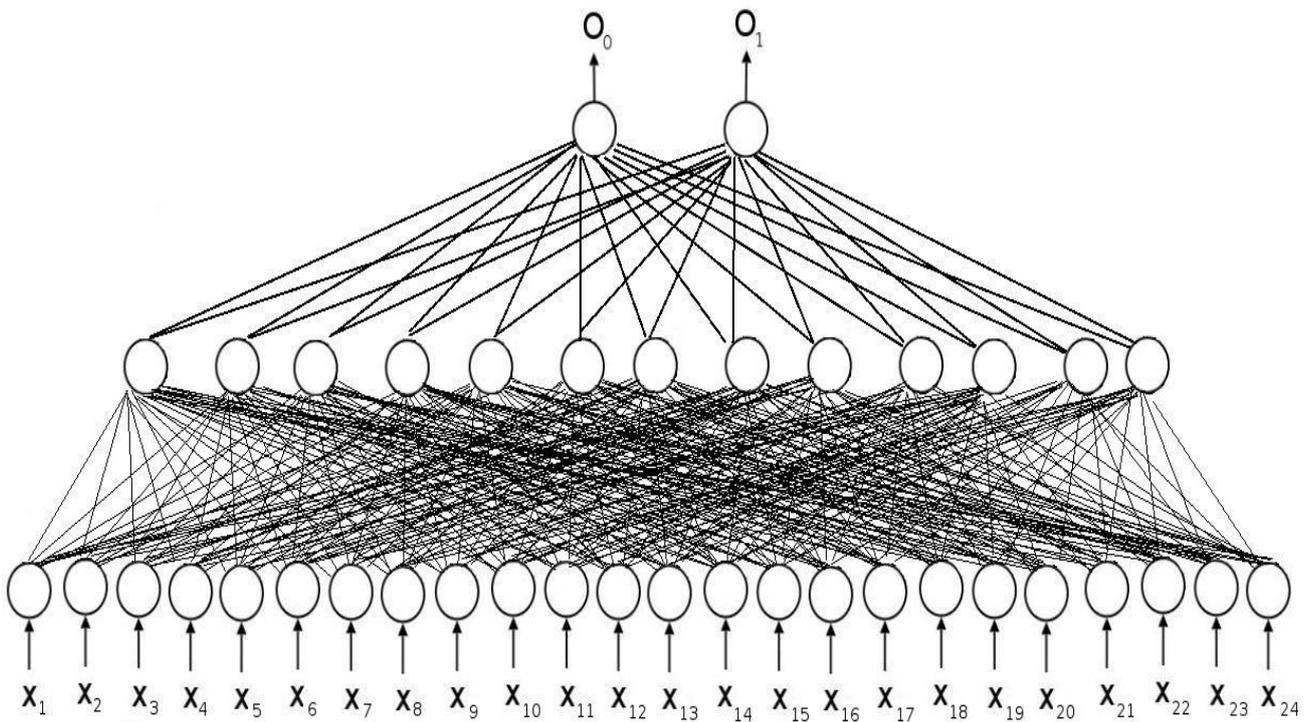


Figura 2.4 Topología de la MLP con los datos de entrada binarios

2.6.2 Fase de Entrenamiento.

El primer componente del entrenamiento de cualquier red neuronal es un buen set de entrenamiento que consta de todos los patrones que servirán como ejemplo a la red para que esta aprenda bien su tarea, nuestro set está compuesto por 105 imágenes tomadas de internet.

El proceso de entrenamiento adapta la red neuronal ante una situación específica. Sin embargo, a nivel estructural, el desarrollador también debe elegir una topología para la red neuronal que refleje el uso para el cual se está diseñando. Los distintos tipos de enlaces entre las neuronas dan como resultado redes con características diferentes.

El Perceptrón Multicapa se caracteriza por presentar una no-linealidad en la salida, capas de neuronas ocultas y un alto grado de conectividad. Es de entrenamiento supervisado. Utiliza el algoritmo de retro-propagación del error. Este algoritmo está basado en la regla de aprendizaje por corrección de error, considerada como una generalización del algoritmo de los cuadrados mínimos (LMS), utilizado en filtrado adaptivo mediante redes lineales simples.

Su operación consta de dos fases, una directa y una inversa o de retroceso. En la fase directa, se ingresa el patrón de actividad en la capa de entrada de la red (vector de entrada), que recorre todas las capas subsiguientes. Se obtiene la respuesta real de la red en la capa de salida. En esta fase, los pesos sinápticos de la red permanecen fijos. En la fase inversa, los pesos sinápticos son ajustados de acuerdo con la regla de corrección del error. Esta regla, conocida como método de Levenberg-Marquardt (Marquardt, 1963), minimiza el cuadrado de las diferencias entre la respuesta o salida deseada y la salida real de la red.

El entrenamiento se llevo a cabo tomando una muestra de 105 imágenes que contienen mayormente personas, donde este presente un buen porcentaje del color de la piel. Estas fueron segmentadas manualmente, formándose una imagen binaria donde el color blanco representa a la piel (valor 1) y de negro los restante (valor 0). Posteriormente se paso a tomar de la imagen original los valores del RGB de cada pixel y compararlo con el valor de la imagen binaria, donde el pixel de la imagen original tiene de acuerdo al valor que presenta (0 o 1) en la imagen binaria entonces es etiquetado como piel o no piel. Creándose un fichero que contiene esta información para luego pasar a entrenar la red. A continuación se muestra un ejemplo de cómo debe ser el fichero, este es de gran importancia ya que de haber un error en este traerá consigo un fallo en el entrenamiento.

5 3 1
0.1234 0.23453 0.123456
0
0.41253 0.37654 0.51222
1
0.98881 0.71234 0.79685
0
0.62413 0.30214 0.46854
1
0.01231 0.12000 0.03214
0

Algoritmo de retropropagación (18).

El procedimiento de retropropagación es una forma relativamente eficiente de calcular qué tanto se mejora el desempeño con los cambios individuales en los pesos. Se conoce como procedimiento de retropropagación porque, primero calcula cambios en la capa final, reutiliza gran parte de los mismos cálculos para calcular los cambios de los pesos de la penúltima capa y, finalmente, regresa a la capa inicial.

Conocido también por su denominación en inglés, *backpropagation*, este algoritmo es de uso muy extendido en las aplicaciones de redes neurales. Se le identifica como parte de los procesos de aprendizaje supervisados, pues requiere que la red disponga de información para parte de ella, usualmente en las entradas y salidas. El algoritmo es iterativo alrededor de los pesos, pues la intención es ajustarlos en cada unidad de tal manera que se reduzca el error entre la salida deseada y la real. De ahí que la variable principal sea conocida como *error derivado de los pesos*, *EW* (por las siglas en inglés).

Al comparar la señal de salida con una respuesta deseada o salida objetivo, $d(t)$, se produce una señal de error, $e(t)$, energía de error. Señal de error en la neurona de salida j en la iteración t $e(t)=d(t) - y(t)$ donde t denota el tiempo discreto.

Pasos del algoritmo.

- INICIALIZACION

Escoger los pesos sinápticos y umbrales de una distribución uniforme cuya media es cero y varianza se escoge de tal forma que haga la desviación estándar de los campos inducidos de las neuronas con valor en la transición entre las partes lineal y saturada de la función de activación sigmoide.

- PRESENTACIONES DE LOS EJEMPLOS DE ENTRENAMIENTO.

Presentar un estado de los patrones de entrenamiento a la red para conseguir la salida.

- CALCULO HACIA DELANTE.

Sea $(x(t), d(t))$ un ejemplo de entrenamiento

$$y_j^{(L)} = o_j(t)$$
$$e_j(t) = d_j(t) - o_j(t)$$

- CALCULO HACIA ATRÁS

Calculo de los gradientes locales.

$$\delta_j^{(l)}(t) = \begin{cases} e_j^{(L)}(t) f'(v_j^{(L)}(t)) & \text{para la neurona } j \text{ en la capa de salida } L \\ f'(v_k(t)) \sum_k \delta_k(t) w_{kj}(t) & \text{para la neurona } j \text{ en la capa oculta } l \end{cases}$$

- AJUSTE DE LOS PESOS.

Ajustar los pesos en la dirección adecuada y en la proporción adecuada para reducir el error de la red en la capa l de acuerdo con:

$$w_{ji}^{(l)}(t+1) = w_{ji}^{(l)}(t) + \alpha [w_{ji}^{(l)}(t-1)] + \eta \delta_j^{(l)}(t) y_i^{(l-1)}(t)$$

Los tres pasos anteriores se repiten a través de un número de ciclos para diferentes patrones de entrada-salida. Normalmente en algunos cientos de ciclos el sistema converge y se obtienen los pesos adecuados. Muy importante el algoritmo de retropropagación es una forma de repartir el cálculo del gradiente entre las distintas unidades, de manera que el cambio en cada peso puede calcularse por la unidad a la que afecta ese peso, y usando sólo información local.

Modo de entrenamiento.

Por lotes (Batch): Actualización de los pesos después de presentar todos los ejemplos de entrenamiento.

$$\Delta w_{ji}(t) = -\frac{\eta}{N} \sum_{t=1}^q e_j(t) \frac{\partial e_j(t)}{\partial w_{ji}}$$

Se considera que el algoritmo ha convergido cuando la norma euclídeana del vector gradiente alcanza un umbral de gradiente suficientemente pequeño. Se considera que el algoritmo ha convergido cuando la norma tasa absoluta de cambio en el error cuadrado promedio por época es suficientemente pequeño.

2.6.3 Fase de ejecución.

En la fase de ejecución (operación normal de la red), los pesos permanecen fijos. En primer lugar, cada neurona (i, j) calcula la similitud entre el vector de entrada x y su propio vector de pesos sinápticos w , según una cierta medida de criterio de error establecido vista en la fase de aprendizaje.

A continuación, se declara vencedora la neurona cuyo vector de pesos w es más similar al de entradas. Obteniéndose un valor que se acerca a una de las dos clases definidas, es decir mientras más se converja a 1 es porque el pixel fue determinado por la red como de tipo piel y al contrario mientras más se acerque a 0 fue procesado como no piel.

2.7 Levantamiento de requisitos.

Una de las actividades más importantes en el desarrollo de un sistema es la captura de requisitos, ya que en esta actividad se recogen las necesidades que tienen los clientes para el desarrollo del sistema. Para recoger estos requisitos se utiliza la Lista de Reserva del Producto

Lista de Reserva del Producto

La Lista de Reserva del Producto no es más que un artefacto generado para, de forma natural recoger las funcionalidades que debe cumplir el sistema para su desarrollo y define una lista de actividades que priorizan el trabajo que se va a realizar permitiendo organizarlo y controlarlo.

Asignado a	Item *	Descripción	Estimación
		Prioridad	Alta
Cesar	1	Entrenar una Red Neuronal para la segmentación de imágenes por el color de la piel	2 semanas.
		Prioridad	Alta
Cesar	2	Implementar un componente que permita segmentar un grupo de imágenes mediante la Redes Neuronales Artificiales.	1 semana.
		Prioridad	Media
Cesar	3	Implementar un componente que permita determinar de un conjunto de imágenes segmentadas por los métodos "manual, Jones y Regh y RNA los falsos negativos y positivos.	1 semana.
RFN (Requisitos no funcionales)			
		La solución debe ser óptima para cualquier hardware, además de que sea manejable, gestionable y optima ofreciendo la funcionalidad requerida.	

Tabla 1 Lista de Reserva del Producto

2.8 Planificación del Proyecto por Roles.

Como SXP define diferentes roles es necesario darle una organización al proyecto utilizando los mismos, quedando estructurado de la siguiente forma:

Rol	Responsabilidad	Nombre
<i>Gerente</i>	<i>Es el responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto.</i>	<i>Cesar R Aguilera Valls</i>
<i>Cliente</i>	<i>Contribuye a definir las historias de usuario y los casos de prueba de aceptación, para validar su implementación</i>	<i>Equipo de desarrollo de MOCIC</i>
<i>Programadores</i>	<i>Define las tareas de ingeniería y produce el código del sistema. Además selecciona el estándar de programación a utilizar, controlando incluso la gestión recambios</i>	<i>Cesar R Aguilera Valls</i>
<i>Analista</i>	<i>Escribe la concepción del sistema y las historias de usuario. Crea el Modelo de historia de usuario del negocio y la LRP. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio</i>	<i>Cesar R Aguilera Valls</i>
<i>Diseñadores</i>	<i>Encargados del diseño del sistema; máximos responsable de supervisar el proceso de construcción.</i>	<i>Cesar R Aguilera Valls</i>
<i>Encargado de Pruebas</i>	<i>Escribe los casos de prueba de aceptación. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas</i>	<i>Cesar R Aguilera Valls</i>

Tabla 2 Planificación de Roles

2.9 Historias de Usuarios y tareas.

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto.

A continuación se darán a conocer las distintas historias de usuarios que están presentes en el sistema; así como su relación con los prototipos de interfaz usuario. Ésta es solo una planificación inicial, el proceso es cambiante para ir adecuándolo a las necesidades y nuevas propuestas.

Historia de Usuario	
Número: M_DP_0.1	Nombre Historia de Usuario: Entrenamiento de la RNA
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Programador (Cesar)	Iteración Asignada: Sprint 1
Prioridad en Negocio: Alta	Puntos Estimados: 2 semana
Riesgo en Desarrollo: Alto	Puntos Reales: 2 semanas (100% culminado)
Descripción: Se tomarán de una muestra de 105 imágenes segmentadas manualmente los píxeles que son piel y los no-piel. Luego con estos datos se les pasara a la red y se procederá a entrenar.	
Observaciones:	

CAPÍTULO 2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Estudiar el proceso de entrenamiento de las RNA	
Tipo de Tarea : Estudio	Puntos Estimados: 1 semana
Fecha Inicio: 19/01/2009	Fecha Fin: 23/01/09
Programador Responsable: Cesar	
Descripción: Se estudiaron las diferentes formas de entrenamiento, para llevar a cabo el proceso.	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Estudio de librería OpenCV y FANN para el trabajo con las imágenes y el trabajo con la RNA	
Tipo de Tarea : Investigativa	Puntos Estimados: 1 semana
Fecha Inicio: 15/4/2009	Fecha Fin: 20/4/2009
Programador Responsable: Cesar	
Descripción: Se hace un estudio de la librería OpenCV para el manejo de las imágenes, extracción del valor RGB de los pixeles. Se llevo a cabo un estudio de la FANN para la creación de la red y el proceso de aprendizaje.	

CAPÍTULO 2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 1
Nombre Tarea: Entrenar una Red Neuronal Artificial.	
Tipo de Tarea : desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 22/4/2009	Fecha Fin: 29/4/2009
Programador Responsable: Cesar	
Descripción: Se desarrolló un algoritmo que permite entrenar una Red Neuronal Artificial	

Historia de Usuario	
Número: M_DP_0.2	Nombre Historia de Usuario: Segmentación mediante RNA
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Programador (Cesar)	Iteración Asignada: Sprint 3
Prioridad en Negocio: Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Alto	Puntos Reales:
Descripción: Se encargará de segmentar las imágenes mediante Redes Neuronales.	
Observaciones: Para que éste proceso se lleve a cabo es necesario que se allá entrenado la red adecuadamente y que este presenta un alta efectividad.	

CAPÍTULO 2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 2
Nombre Tarea: Estudiar el proceso de segmentación de imágenes basados en RNAs	
Tipo de Tarea : Estudio	Puntos Estimados: 1 semana
Fecha Inicio: 25/01/09	Fecha Fin: 4/02/09
Programador Responsable: Cesar	
Descripción: Se estudió el proceso de segmentación de imágenes para una mejor comprensión y dar una solución factible.	

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 2
Nombre Tarea: Implementar un segmentador de imágenes mediante las Redes Neuronales Artificiales.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 30/4/2009	Fecha Fin: 05/5/2009
Programador Responsable: Cesar	
Descripción: Implementar un algoritmo que permita segmentar imágenes por el color de la piel basado en las Redes Neuronales.	

CAPÍTULO 2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

Historia de Usuario	
Número: M_DP_0.3	Nombre Historia de Usuario: Comprobar efectividad de la RNA y el modelo propuesto por Jones and Regh
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Programador (Cesar)	Iteración Asignada: Sprint 3
Prioridad en Negocio: Media	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Bajo	Puntos Reales: (100% culminado)
Descripción: Se tomará una muestra de imágenes y se segmentará por ambos métodos y luego se procederá a sacar los falsos positivos y los falsos negativos, así como la efectividad.	
Observaciones: Es necesario tener una muestra de imágenes segmentadas manualmente y sus originales.	

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 2
Nombre Tarea: Segmentar 100 imágenes	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Cesar	
Descripción: Se segmentaron 100 imágenes mediante los métodos Jones and Regh y las Redes Neuronales de Kohonen	

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 2
Nombre Tarea: Implementar un componente para determinar la efectividad.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Cesar	
Descripción: Se implementó un componente que extrae de un conjunto de imágenes binarias, segmentadas por diferentes métodos (manual, Jones and Regh y Redes de Kohonen) la cantidad de falsos positivos y negativos. Posibilitando comparar la efectividad de los métodos Jones and Regh y Redes Neuronales de Kohonen.	

2.10 Lista de riesgo.

En el ciclo de desarrollo de cualquier software siempre hay situaciones que poner en peligro el desarrollo del sistema, para contrarrestar esto hay que tener una buena gestión de los riesgos que se pueden presentar. Esta tiene como objetivo marcar las estrategias y formas de actuar del equipo de trabajo frente a los riesgos: cómo evitarlos, monitorizarlos, gestionarlos y actuar ante contingencia, estableciendo las tareas principales a realizar en caso de que se llevaran a cabo.

Por tanto después de tener definida las historias de usuarios del sistema la siguiente acción es definir la lista de riesgo que pueden atrasar el desarrollo de la solución así como las tareas para mitigarlo.

CAPÍTULO 2. ANÁLISIS Y DESARROLLO DE LA SOLUCIÓN PROPUESTA.

Riesgo	Tipo de Riesgo	Impacto	Descripción	Probabilidad	Efectos	Mitigación del riesgo
1	Personal	Retraso en la entrega del producto.	Ausencia de un programador, analista, documentador o un diseñador.	Media	Serios	El riesgo es resarcido planificando bien las tareas de cada roll sin crear dependencia extrema de un trabajo con otro.
2	Tecnológico	Retraso del trabajo incidiendo negativamente en la terminación del producto.	Rotura de computadora donde se encuentra información importante del proyecto.	Alta	Serios	Se evita revisando diariamente las computadoras para ver si todo se encuentra en condiciones óptimas para la realización del trabajo, así como realizar salvadas de la información importante.
3	Tecnológico	Desagrado por parte del equipo, amplia el período de culminación.	Afectación del fluido eléctrico en el laboratorio de producción.	Baja	Tolerable	Trabajar el mayor tiempo posible durante la noche ya que hay menos consumo de electricidad durante la noche y la red no se sobrecarga.
5	Personal	Retraso en las actividades que desarrolla esa persona dentro del proyecto.	Enfermedad de uno de los miembros del equipo.	Alta	Tolerable	Tener preparado a más de una persona que sepa realizar las estas actividades para que el trabajo del equipo no se atrase debido a estas consecuencias.
6	Personal	Retraso del trabajo	Falta de Experiencia	Media	Tolerable	Obtener entrenamiento y asesoría de personas expertas y negociar compromisos adecuados

Tabla 3 Lista de Riesgo

2.11 Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc.

A continuación se presenta el diagrama de componente para sistema que se propone. Para una mejor comprensión del funcionamiento de la propuesta se dividió el diagrama en las dos fases del algoritmo.

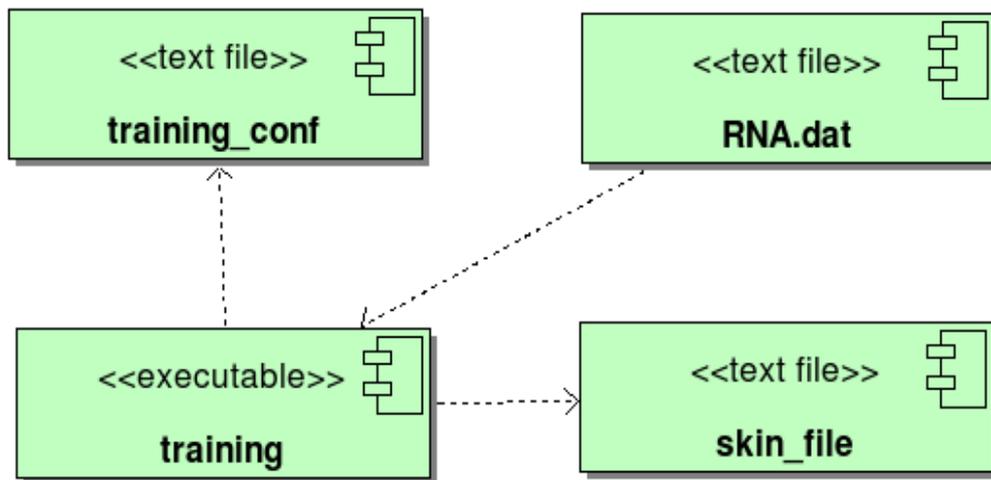


Figura. 2.5 Diagrama de componente fase entrenamiento.

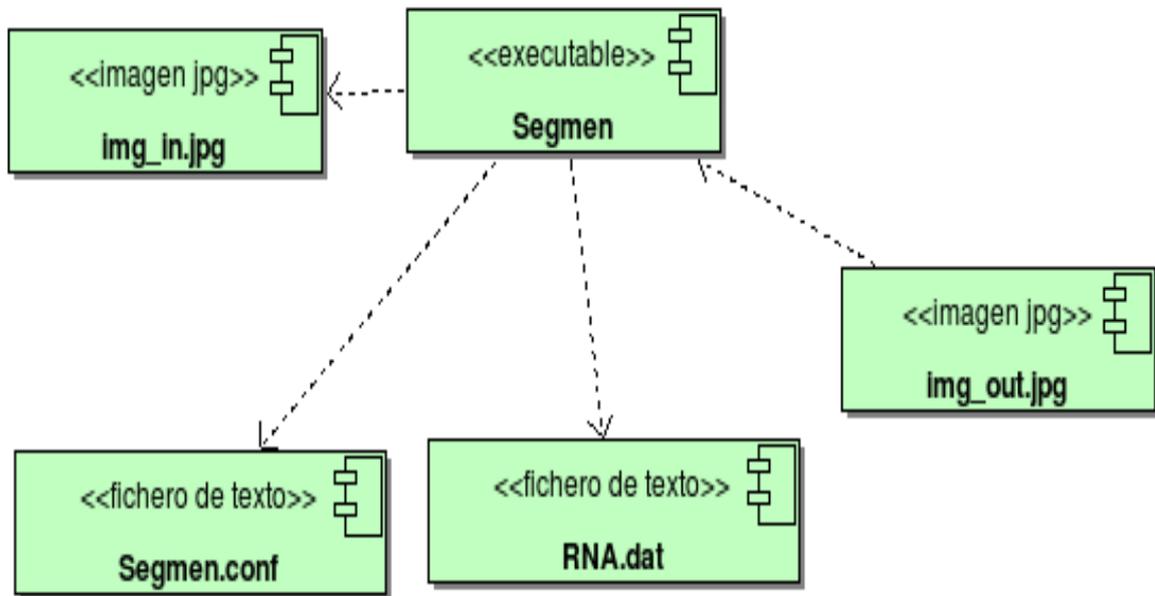


Figura. 2.6 Diagrama de componente fase Segmentación.

2.12 Planificación de Historias de Usuario.

Planificar acertadamente el proyecto es necesario para estimar de forma ideal el tiempo que les tomará a los programadores la codificación de cada una de las historias de usuario. A partir de la prioridad de las historias se decide cuales de ellas se implementarán en las primeras iteraciones, las funcionalidades críticas del sistema deben ser codificadas en iteraciones tempranas del ciclo. Esta planificación queda recogida en la plantilla Plan de Reléase que es una de los artefactos que genera la metodología SXP.

2.12.1 Plan de Releases.

Planificar acertadamente el proyecto es necesario para estimar de forma ideal el tiempo que les tomará a los programadores la codificación de cada una de las historias de usuario. A partir de la prioridad de las historias se decide cuales de ellas se implementarán en las primeras iteraciones, las funcionalidades críticas del sistema deben ser codificadas en iteraciones tempranas del ciclo.

En este paso se define el plan de releases e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha Historia. Como resultado de la priorización de Historias se llegó a la siguiente planificación:

Reléase	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 1	En esta iteración se implemento el componente que permite entrenar la RNA	1	
Iteración 2	En esta iteración el sistema será capas de segmentar una imagen dada.	2	
Iteración 3	Se determinan la eficiencia de la solución. Implementándose un componente que permita determinar los falsos positivos y los falsos negativos.	3	

Tabla 4 Plan de Releases

2.13 Conclusiones.

Se presento una breve explicación de la solución propuesta, logrando una mejor comprensión del proceso de segmentación. Tras efectuar el análisis del tema tratado se comprendieron sus requisitos de manera más precisa. Además se realizo la planificación del proyecto, estableciéndose las historias de usuarios y las principales tareas asociadas a las mismas. Para una mejor visión del problema se desarrollo el diagrama de componentes.

3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

3.1 Introducción.

Una parte muy importante en todo desarrollo de software es la etapa de pruebas, en la metodología SXP se requiere de una etapa de pruebas en cada iteración de desarrollo del producto. En estas etapas se le realizan pruebas al software con el objetivo de verificar si se cumplieron con los objetivos trazados en la iteración, haciéndose el proceso iterativo; e incremental, de pasar el proceso de prueba con éxito en cada iteración para proseguir con el próximo.

En la construcción de la solución se elaboraron un Plan de pruebas que en cada iteración contiene un conjunto de Casos de prueba a los que fue sometida la solución para comprobar el funcionamiento en correspondencia con las Historias de Usuarios definidas.

3.2 Casos de pruebas.

Se definieron casos de prueba para todas las historias de usuario. A continuación se muestran las pruebas que se realizaron mediante tablas definidas en la metodología utilizada para el desarrollo de la aplicación. Los principales elementos que se muestran son las condiciones que el sistema debe cumplir para que se ejecute el caso de prueba así como la entrada y salida de los pasos de ejecución del caso de prueba realizado. Por último se muestra el resultado esperado y la evaluación de la prueba en relación a si fue satisfactoria o no.

3.2.1 Caso de prueba para la historia de usuario M_DP_0.1.

Esta sección cubre el conjunto de pruebas que se le realizaron a la historia de usuario:

Entrenamiento de la RNA.

Este caso de prueba pretende comprobar que la fase de entrenamiento se llevo a cabo satisfactoriamente, creándose los ficheros que se deben generar.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 1	Nombre Historia de Usuario:
Nombre de la persona que realiza la prueba: Programador (Cesar)	
Descripción de la Prueba: Se ejecuta la aplicación y se comprueba si el sistema es capaz de generar el fichero de entrenamiento correctamente.	
Condiciones de Ejecución: La aplicación no puede contener errores de ningún tipo en su compilación. Además debe estar creado el fichero de entrada de datos a la red neuronal.	
Entrada / Pasos de ejecución: Se compila la aplicación y se ejecuta.	
Resultado Esperado: Que se creé el fichero de entrenamiento.	
Evaluación de la Prueba: Prueba satisfactoria.	

3.2.2 Caso de prueba para la historia de usuario M_DP_0.2

Este caso de prueba pretende comprobar la efectividad que presenta el sistema determinando los falsos positivos y los falsos negativos, a la hora de segmentar una muestra de imágenes. Este es el principal caso de prueba de la solución propuesta. En este dado una muestra de imágenes originales que han sido segmentadas manualmente, éstas se segmentan mediante la RNA luego se compara los pixel de la misma imagen segmentada manualmente y la resultante del proceso de segmentación determinándose los falsos positivos y los falsos negativos. Se determina la efectividad teniendo en cuenta el porcentaje que presentan estos con respecto al total de pixel analizados ver anexo.

CAPÍTULO 3.VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Caso de Prueba de Aceptación	
Código Caso de Prueba: 2	Nombre Historia de Usuario:
Nombre de la persona que realiza la prueba: Programador (Cesar)	
Descripción de la Prueba: Se ejecuta la aplicación y se determinan los falsos negativos y falsos positivos, luego se determina el por ciento de efectividad del proceso de segmentación de acuerdo al total de píxeles analizados.	
Condiciones de Ejecución: La aplicación no puede contener errores de ningún tipo en su compilación. Además debe estar segmentada manualmente las imágenes que se van a segmentar mediante la aplicación.	
Entrada / Pasos de ejecución: Se compila la aplicación y se ejecuta. Se introducen una muestra de imágenes.	
Resultado Esperado: Que el proceso de segmentación sea aceptable.	
Evaluación de la Prueba: Prueba satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 3	Nombre Historia de Usuario:
Nombre de la persona que realiza la prueba: Programador (Cesar)	
Descripción de la Prueba: Se determinan los falsos negativos y falsos positivos del modelo propuesto por Jones and Regh, luego se determina el por ciento de efectividad de acuerdo al total de píxeles analizados.	
Condiciones de Ejecución: Para lograr el objetivo se debe segmentar manualmente una muestra de imágenes, que posteriormente se le pasaran al componente. Se debe asegurar de guardar las mismas imágenes.	
Entrada / Pasos de ejecución: Se compila la aplicación y se ejecuta. Se introducen la ubicación de la muestra de imágenes segmentadas manualmente y las originales..	
Resultado Esperado: Que el proceso de segmentación sea aceptable.	
Evaluación de la Prueba: Prueba satisfactoria.	

3.3 Resultados obtenidos.

En este epígrafe se relacionan los resultados alcanzados durante el proceso de desarrollo del presente trabajo. Como fruto de este trabajo se ha obtenido un sistema que permite segmentar imágenes de forma automática, del cual se encuentra disponible una primera versión que se integró al proyecto MOCIC, permitiendo una mayor efectividad a la hora de clasificar las imágenes.

Acerca de las funcionalidades obtenidas.

Las funcionalidades que brinda el sistema son:

- Es capaz de entrenar una Red Neuronal Artificial de tipo MLP de forma viable, gestionada por un fichero de configuración.
- Es capaz de segmentar imágenes a color mediante una MLP.

3.4 Acerca del tiempo de desarrollo.

El problema de la necesidad de implementar un segmentador basados en Redes Neuronales Artificiales era una situación que se venía dando desde el desarrollo del modulo MCAIP, debido a que el proceso de de segmentación de imágenes esta basado en el método propuesto por Jones and Regh y este método presenta una baja efectividad.

Debido a la necesidad de implementar una herramienta capaz de suplir las carencias existentes surge la idea de implementar este sistema. Al principio del 2009 se comienza el desarrollo del segmentador de imágenes con tareas de investigación. El producto estará sujeto a mejoras ya que posteriormente se puede optimizar el código. Para la realización de este componente se estudiaron librerías de procesamiento de imágenes y el trabajo con Redes Neuronales Artificiales. El tiempo de desarrollo fue relativamente corto y se cumplieron los objetivos trazados.

3.5 Conclusiones.

Se presentaron los casos de pruebas más importantes que guiaron la calidad del sistema, y determinaron en cada momento si se estaba o no en condiciones de continuar avanzando. En el análisis de los resultados obtenidos se muestran las funcionalidades alcanzadas por el sistema en el período que se ha estado trabajando en su desarrollo.

CONCLUSIONES GENERALES

Con la culminación del presente trabajo de diploma se cumple con el objetivo trazado en el mismo, mediante la implementación de un sistema capas de segmentar imágenes de forma automática. Se cumplió además con el siguiente aspecto:

- Se realizó un estudio de los algoritmos de segmentación basados en Redes Neuronales Artificiales más usados a nivel mundial.
- Se eligió como algoritmo para segmentar las imágenes un Perceptrón Multicapa. El cual presenta una mayor efectividad que el algoritmo propuesto por Jones and Regh y permitirá que aumente la efectividad del modulo MCAIP a la hora de clasificar imágenes
- Se implementó un algoritmo flexible, cómodo y fácil de usar, capaz de segmentar imágenes.

RECOMENDACIONES.

Después de la investigación realizada y del desarrollo del sistema se recomienda:

- ❖ Continuar el estudio de las tendencias actuales en el campo de la segmentación de imágenes y en el de las RNA, con vista a desarrollar nuevos algoritmo más efectivos y rápidos.
- ❖ Mejorar el método y continuar con la optimización del sistema para lograr mejoras en la clasificación de imágenes.
- ❖ Estudiar otros posibles algoritmos de entrenamiento que permitan una mejor efectividad y una mayor rapidez a la hora de clasificar.

REFERENCIAS BIBLIOGRÁFICAS.

1. **EMC.** "Estadísticas que reflejan el tamaño de Internet". Disponible en:<<http://www.averlo.com/notas/ciencia/0307/que-grande-internet-actualidad.html>> [Adcedida 02 febrero 2009].
2. **De Armas, C., T. Delgado, E. Hernández, H. Vega y A. Mena.** *TELEMAP 2.0: Software for Geosciences*. Havana, Proceedings : En Regional Conference of Latin American and Caribbean Contries International Geographical Union, 1994.
3. **García, E. B.** *Empleo de las redes neuronales artificiales en la clasificación de imágenes multiespectrales de teledetección*. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. La Habana : s.n., 1996.
4. **Gil, J. L.** *Procedimientos para el análisis de textura en imágenes. Un estudio*. Tesis presentada en opción al título de Master en Informática Aplicada a la Ingeniería y la Arquitectura, ISPJAE. La Habana : s.n., 2005.
5. **M.B. Vega, E.B. García, R. Sánchez, L. Jiménez.** *Software Tn Estudio V2.0: Integration of digital image processing tools with pattern recognition to support the remote sensing studies*. En *International Archives of Photogrammetry and Remote Sensing*. Amsterdam : s.n., 2000. Vol. XXXIII.
6. **Castro, Kiuver Kaddiel Ibanez.** *Diseño e implementacion de un categorizador automatico de imagenes pornograficas*. Ciudad de la Habana : s.n., 2008.
7. **Gonzalez R., Woods R.** *Digital Image Processing, Addison-Wesley*. 2005.
8. **Martin, Marcos.** *Tecnicas Clasicas de Segmentacion de Imagen*. 2004.
9. **Véneré, Mariana del Fresno. Marcelo J.** *Segmentación de imágenes médicas por crecimiento de regiones con conocimiento adicional*. 2009.
10. **K, Castleman.** *Digital Image Processing*. Prentice Hall : s.n., 2006.
11. *Documento, Estudio de campos de golf mediante tecnicas de segmentacion*. s.l. : Revista de Teledeteccion, 2001.
12. **Gonzalez, Mary Carmen Trejo y Jose Gustavo.** *Software para el Descubrimiento de conocimiento en Grandes Bases de Datos de informacion Cientifico-Tecnologica*. 2005.
13. *Redes neuronales artificiales. Fundamentos, modelos y aplicaciones*.
14. **TIOBE.** *TIOBE Programming Community Index*. Disponible en <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. 2008.
15. **Guillemí Martín, Jessie.** *Las Metodologías de Desarrollo de Software. Análisis y Modelado de la Solución Informática para el proceso de Inscripción en los Registros Mercantiles de la República*

REFERENCIA BIBLIOGRÁFICA.

Bolivariana de Venezuela. Caracas : s.n., 2007.

16. *Modelo de Dominio*[Internet]. Disponible en: http://ie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_declase2.pdf .

17. **Jesus Angulo, Jean Serra**. *Segmentación de imágenes en color utilizando histogramas bi-variables en espacios color polares*. 2005 : Revista Computación y Sistemas. Vol. 8, No. 4.

18. **Tobias Mojica Ph.D, Roberto A. Pava Ing.** *Redes Neuronales supervisadas y no supervisadas. Algoritmo de retropropagación*. 2005. Disponible en http://www.virtual.unal.edu.co/cursos/ingenieria/2001832/lecciones/cap_4/redes3.htm >

BIBLIOGRAFÍA.

- José H. Canós, Patricio Letelier, y Msc Carmen Penadés. Metodologías Ágiles en el desarrollo de Software. 2005. Disponible en <<http://www.willydev.net/descargas/prev/TodoAgil.Pdf>>. [Accedida 08 Mayo 2009].
- Dr. Cs. Carlos Álvarez de Zayas. Metodología de la investigación científica. Centro de estudios de educación física Manuel F. Gran, Santiago de Cuba 1995. Disponible en <<http://teleformacion.uci.cu/mod/resource/view.php?id=2740> > [Accedida 25 noviembre 2008].
- Colectivo de autores. Metodología de la investigación educacional. Desafíos y polémicas actuales 2003. Disponible en <<http://teleformacion.uci.cu/mod/resource/view.php?id=2740>> [Accedida 10 diciembre 2008].
- Raúl Mata Botana. Tablas en LaTeX. Disponible en: <<http://www.lug.fi.uba.ar/documentos/tablas/tablas.pdf>> [Accedida 08 mayo 2009].
- George Gratzer. *Math into LaTeX*. Disponible en: <<ftp://ftp.tex.ac.uk/tex-archive/info/mil/mil.pdf>> [Accedida 28 junio 2008].
- Vice-Rector de Formación Departamento de Practica Profesional. Propuesta de Guía para la presentación del Trabajo de Diploma. 2005-2006 [Accedida 25 noviembre 2008].
- Kiuver Kaddiel Ibáñez Castro. Diseño e implementación de un categorizador automático de imágenes pornográficas. 2008 [Accedida 08 mayo 2009].
- Raúl Igual y Carlos Medrano. Tutorial de OpenCV 23/04/08. Disponible en <http://eupt2.unizar.es/cmedrano/tutorial_opencv.pdf> [Accedida 20 abril 2009].
- Metodología de la Programación/Informática. Curso 2008-2009. Ficheros de textos. Disponible en <https://belenus.unirioja.es/~beperev/METP_INF/practicas/p11.pdf.> [Accedida 20 abril 2009].
- Curso de C++. Trabajar con ficheros [Internet]. Disponible en <<http://c.conclase.net/curso/index.php?cap=039>>[Accedida 18 abril 2009].
- Darío de Miguel Benito. Detección automática del color de la piel en imágenes bidimensionales basado en el análisis de regiones. 2004/2005[Internet]. Disponible en <<http://www.escet.urjc.es/~jjpantrigo/PFCs/MemoriaPielFeb05.pdf>>

BIBLIOGRAFÍA.

- C. Pérez, M. A. Vicente, C. Fernandez, O. Reinoso, A. Gil. Aplicación de los diferentes espacios de color para detección y seguimiento de caras. Universidad Miguel Hernández. España [Internet]. Disponible en <http://www.ceautomatica.es/actividades/jornadas/XXIV/documentos/viar/117.pdf>.
- Jesús Angulo, Jean Serra. Segmentación de imágenes en color utilizando histogramas bi-variables en espacios color polares luminaria/saturación/matiz. Revista "Computación y Sistemas", Vol. 8, No 4, Junio 2005 [Internet]. Disponible en http://cmm.ensmp.fr/~angulo/publicat/AnguloSerra_CyS_05.pdf
- Vicent Caselles, Alejandro Fragi. La segmentación de imágenes. El Método de los contornos activos geométricos [Internet]. Disponible en <http://www.encuentros-multidisciplinares.org/Revistan%BA23/Vicent%20Caselles%20Costa%20y%20Alejandro%20Frangi%20Caregnato.pdf>
- Marcos Martín. Técnicas Clásicas de Segmentación de Imagen. 4 de mayo de 2004. Disponible en <http://www.elai.upm.es/spain/Asignaturas/Robotica/PracticasROVA/prROVA5Segmentacion.pdf>
- Técnicas de segmentación de imágenes (III). Disponible en <http://apphys.uned.es>. [Accedida 15 enero 2009].
- Antonio Mosquera González. Segmentación. Tratamiento Computacional de Imágenes. Disponible en http://ocw.usc.es/export/sites/default/es/servizos/ceta/opencourseware/Lic_Fisica/tci/material/teoria/segmentacion [Accedida 16 enero 2009].
- Mary Carmen Trejo Ávila, José Gustavo González Ángeles. Software para el Descubrimiento de Conocimiento en Grandes Bases de Datos de Información Científico-Tecnológica. Disponible en <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH019e.dir/doc.pdf> [Accedida abril 2009].
- Alma Rosa Aguilar Sánchez. Las redes neuronales artificiales como una herramienta alternativa o complementaria a los métodos de clasificación tradicionales [Internet], Agosto 2008. Disponible en [Alma Rosa Aguilar Sánchez. Las redes neuronales artificiales como una herramienta alternativa o complementaria a los métodos de clasificación tradicionales](http://www.almarosaaguilar.com/AlmaRosaAguilarSánchez.Lasredesneuronalesartificialescomounaherramientaalternativaocomplementariaalosmétodosdeclasificacióntradicionales) [Accedida 4 mayo 2009].

Anexo A

Flujo de reconocimiento de imágenes.

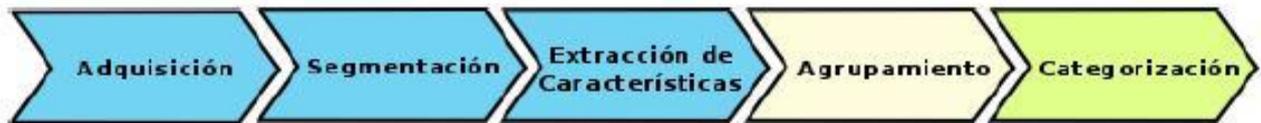


Figura A1 Flujo de reconocimiento de imágenes.

Anexo B

Estructura de las neuronas

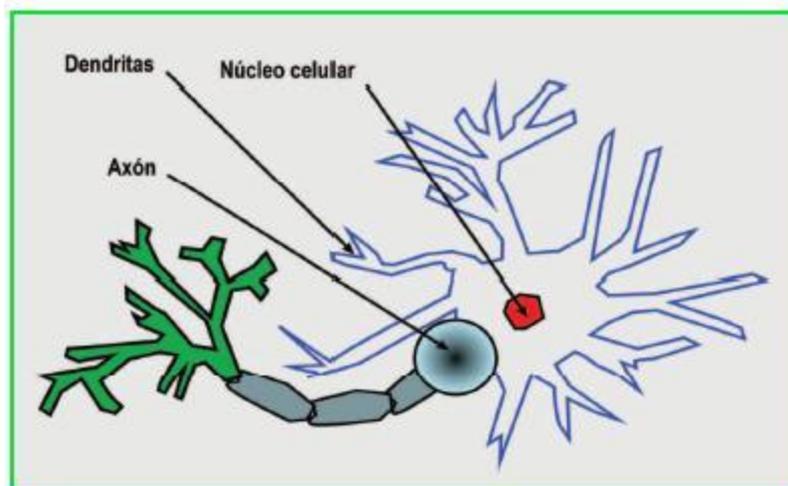


Figura B. 1 Los neurocientíficos fundamentan la capacidad del cerebro humano para reconocer patrones o predecir estados de sistema que son difíciles de calcular en la disposición ramificada de las células nerviosas.

Perceptrón Multicapa.

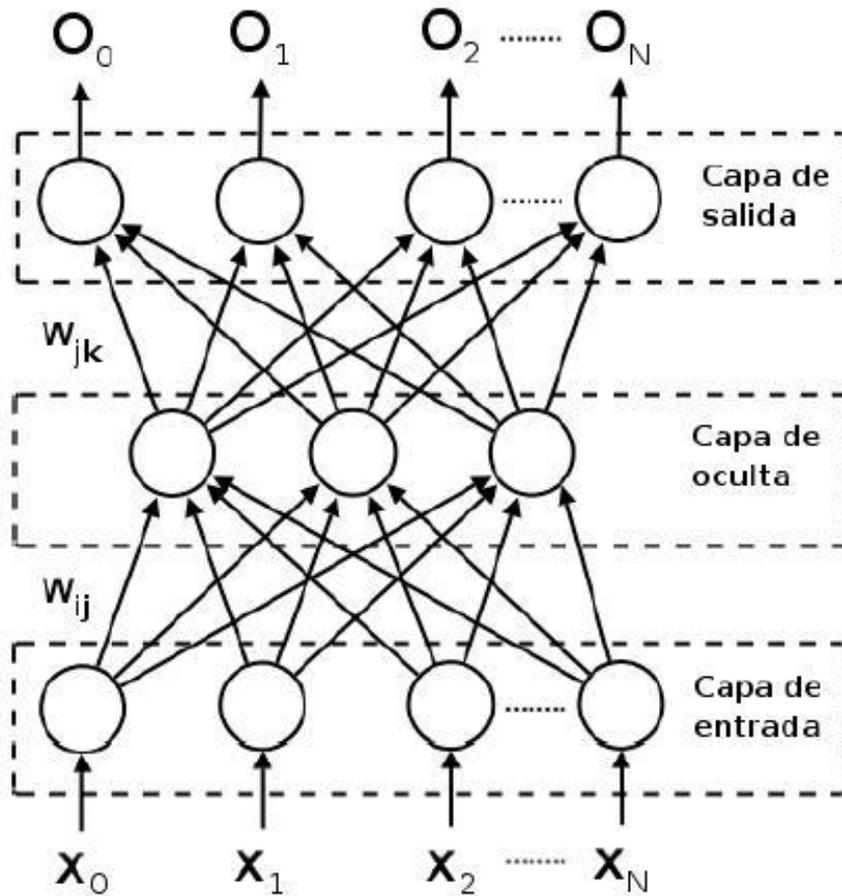


Figura C.1 Arquitectura de una red Neuronal con propagación hacia atrás

Arquitectura del Proyecto MOCIC

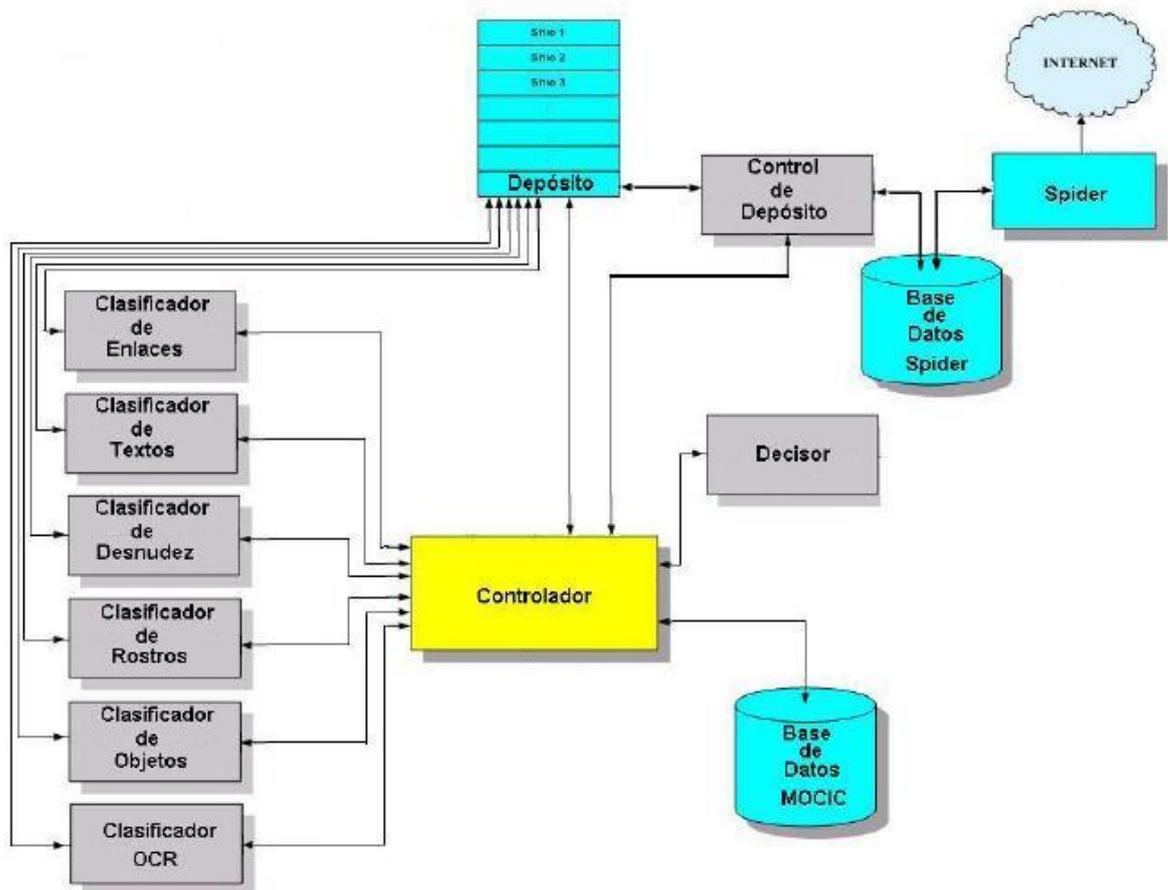


Figura D. 1 Arquitectura de MOCIC

Imágenes original



Imágenes segmentada por RNA



GLOSARIO DE TÉRMINOS

UCI	Universidad de las Ciencias Informáticas.
FILPACON	Filtrado de Paquetes por Contenido: Es un sistema que pretende ser flexible y fiable para regular los contenidos nocivos de Internet.
IA	Inteligencia Artificial Rama de la informática que desarrolla procesos que imitan a la inteligencia de los seres vivos.
MCADHTML	Motor de Categorización Automatizada de Documentos HTML.
OSRI	Oficina de Seguridad para las Redes Informáticas.
BDUC	Base de Datos de URLs Categorizadas.
HTML	HyperText Mark Language: Lenguaje de Marcas de Hipertexto. Es el lenguaje de marcado predominante para la construcción de páginas Web.
Web	Red: La traducción literal de esta palabra inglesa es tela de araña, pero en términos informáticos significa mucho más que eso.
RNA	Redes Neuronales Artificiales
UML	<i>Unified Modeling Language:</i> Lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.
URL	<i>Uniform Resource Locator:</i> Se refiere al texto que identifica a una página Web.

GLOSARIO DE TÉRMINOS

GPL	<p><i>General Public License:</i></p> <p>Licencia Pública General. Es una licencia creada por la <i>Free Software Foundation</i> (FSF) y orientada principalmente a los términos de distribución, modificación y uso del software.</p>
MLP	<p>Multilayer Perceptron:</p> <p>Perceptrón multicapa. Topología de Red Neuronal definida por Rosenblatt en el año 1957.</p>
IDE	<p><i>Integrate Development Enviroment:</i></p> <p>Entorno de desarrollo integrado. Herramienta que se usa para facilitar el desarrollo de software.</p>
IGU	<p>Interfaz Gráfica de Usuario:</p> <p>Software que gestiona la interacción con el usuario de manera gráfica mediante el uso de iconos, menú, ratón, etc.</p>
RGB	<p><i>Red, Green and Blue:</i></p> <p>Rojo, verde y Azul. Es un modelo de color en el cual es posible representar un color mediante la mezcla de estos tres colores primarios.</p>
HSV	<p><i>Hue, Saturation and value:</i></p> <p>Modelo de color basado en tres parámetros tono, saturación y valor.</p>
WWW	<p>World Wide Web:</p> <p>Telaraña o malla mundial. Sistema de información distribuido con mecanismos de hipertexto. Es el universo de servidores HTTP, que permiten mezclar texto, gráficos y archivos de sonido juntos.</p>
CT	<p><i>Computed Tomography</i></p> <p>Procedimiento de diagnóstico que utiliza un equipo de rayos X especial para crear imágenes transversales del cuerpo.</p>
MRI	<p><i>Magnetic resonance imaging</i></p> <p>Técnica no invasiva que utiliza el fenómeno de la resonancia magnética para</p>

GLOSARIO DE TÉRMINOS

obtener información sobre la estructura y composición del cuerpo a analizar.

BCPL

Basic Combined Programming Language.

Es un lenguaje de programación ordenado, potente y muy fácil de adaptar a diferentes arquitecturas.

UNIX.

Sistema operativo portable, multitarea y multiusuario.

CDT

C/C++ Development Tools:

Es un plugin para Eclipse el cual permite la programación en el lenguaje C/C++

GNU

GNU is Not Unix:

Sistema operativo cuyo nombre es un acrónimo recursivo que significa “GNU No es Unix”.

Linux

Es el núcleo (kernel) del sistema operativo libre GNU.

XP

eXtreme Programming. Metodología de desarrollo de software basada en valores como simplicidad, comunicación, retroalimentación y coraje.

MCAIP

Modulo de Categorización Automática de Imágenes Pornográficas.

MOCIC

Motor de Clasificación Inteligente de Contenido.