UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

TÍTULO:

DESARROLLO DE UN SISTEMA BASADO EN VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE MATRÍCULAS VEHICULARES EN EL AEIRNNR DE LA UNL.

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

AUTORES:

JORGE ANIBAL MALES CHALÁN

RONALD FABRICIO ROJAS LIVISACA

DIRECTOR:

Ing. HENRY PATRICIO PAZ ARIAS, Mg. Sc.

LOJA - ECUADOR

2015

CERTIFICACIÓN DEL DIRECTOR

Ing.
Henry Patricio Paz Arias, Mg. Sc.
DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS, DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Haber asesorado y revisado detenida y minuciosamente durante todo su desarrollo, el Proyecto de Fin de Carrera titulado: " **DESARROLLO DE UN SISTEMA BASADO EN VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE MATRÍCULAS VEHICULARES EN EL AEIRNNR DE LA UNL" .** Realizado por los postulantes Jorge Anibal Males Chalán y Ronald Fabricio Rojas Livisaca.

Por lo tanto, autorizo proseguir los trámites legales pertinentes para su presentación y defensa.

Ing. Henry Patricio Paz Arias, Mg. Sc.
DIRECTOR DEL PFC

AUTORÍA

Nosotros **JORGE ANIBAL MALES CHALÁN Y RONALD FABRICIO ROJAS LIVISACA**, declaramos ser autores del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestra tesis en el Repositorio Institucional – Biblioteca Virtual.

Firma: 

Cédula: 1900653658.

Fecha: 19/06/2015.

Firma: 

Cédula: 1900614064.

Fecha: 19/06/2015.

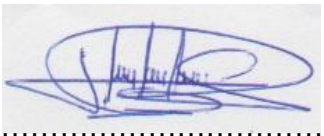
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LOS AUTORES, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Nosotros **JORGE ANIBAL MALES CHALÁN** y **RONALD FABRICIO ROJAS LIVISACA,** declaramos ser autores de la tesis titulada: **"DESARROLLO DE UN SISTEMA BASADO EN VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE MATRÍCULAS VEHICULARES EN EL AEIRNNR DE LA UNL",** como requisito para optar al grado de **INGENIEROS EN SISTEMAS;** autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 19 días del mes de junio del dos mil quince.



Firma: …….

Autor: Jorge Anibal Males Chalán.

Cédula: 1900653658.

Dirección: Yacuambi, Zamora Chinchipe.

Teléfono: 3035367.

Celular: 0986508870.

Correo Electrónico: jamalesc@unl.edu.ec

Firma: …….....

Autor: Ronald Fabricio Rojas Livisaca.

Cédula: 1900614064.

Dirección: Yanzatza, Zamora Chinchipe.

Teléfono: 2324077.

Celular: 0982536919.

Correo Electrónico: rfrojasl@unl.edu.ec

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Henry Patricio Paz Arias, Mg. Sc.

Tribunal de Grado: Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Ing. Alex Vinicio Padilla Encalada, Mg. Sc.

Ing. Franco Hernán Salcedo López, Mg. Sc.

AGRADECIMIENTO

Primeramente agradecemos a Dios por permitirnos lograr el sueño de nuestras vidas, por regalarnos salud y bienestar, y sobre todo por brindarnos la capacidad de adquirir los conocimientos necesarios para poder desarrollar el presente proyecto.

A nuestros padres, hermanos y hermanas, que con su apoyo económico y moral supieron alentarnos a que lleguemos a culminar el desarrollo de nuestro proyecto y de esta manera poder finalizar una nueva etapa de formación profesional.

De manera muy especial, agradecemos a la Universidad Nacional de Loja, a sus autoridades y a todo el personal Docente y Administrativo que labora en tan prestigiosa Institución Educativa, que con su apoyo incondicional supieron guiarnos a cumplir con nuestras metas.

También queremos agradecer a nuestro Director de Proyecto de Fin de Carrera, Mg. Sc. Henry Patricio Paz Arias, por su dedicada dirección y el habernos brindado una excelente asesoría durante el desarrollo del presente trabajo.

LOS AUTORES

DEDICATORIA

Jorge Males

El presente trabajo de investigación lo dedico con infinito amor y humildad, primeramente a Dios, por haberme regalado la dicha de la vida, a mi madre que con su cariño y bondad supo guiarme por el buen camino de la sabiduría y la educación, a mi padre que desde el cielo supo cuidarme y protegerme de los males, a mi abuelita que para mí es mi segunda madre, que con todo el amor del mundo supo cuidarme y enseñarme lo bueno de la vida.

También quiero dedicar este trabajo, a mis hermanos y a toda mi familia que con su apoyo moral hicieron posible el desarrollo del presente proyecto, y así poder alcanzar otra meta de mi vida.

Ronald Rojas

La presente investigación dedico primeramente a Dios, la luz que me ilumina y la fuerza que me hace continuar en la construcción de mis sueños; a mis padres que con su amor y su apoyo incondicional me enseñaron el valor de la educación; a mis hermanos y amigos que han estado junto a mí en momentos de alegría y dificultad. Son todos estos seres quienes fueron mi apoyo para poder culminar con éxito me meta propuesta.

CESIÓN DE DERECHOS

Jorge Anibal Males Chalán y **Ronald Fabricio Rojas Livisaca,** autores intelectuales del presente Proyecto de Fin de Carrera, autorizan a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales no Renovables, y específicamente a la carrera de Ingeniería en Sistemas, el total acceso a su contenido en lo que consideren necesario.

Jorge Anibal Males Chalán **Ronald Fabricio Rojas Livisaca**

a. Título

“Desarrollo de un Sistema basado en Visión Artificial para el reconocimiento de matrículas vehiculares en el AEIRNNR de la UNL”.

b. Resumen

El presente Trabajo de Titulación está enfocado en el uso y aplicación de uno de los sub campos de la Inteligencia Artificial muy conocidos y populares en la actualidad, como es la Visión Artificial o Visión por computador, lo cual consiste básicamente en lo que es el reconocimiento de patrones, procesamiento digital de imágenes, etc.

Si hacemos una analogía, en términos más específicos estaríamos simulando las funciones que realiza el ojo humano, como es el caso de identificar o hacer el seguimiento de algún objeto, razón por la cual surgió una motivación en desarrollar un Sistema que sea capaz de reconocer las placas de los vehículos que ingresan al Área de la Energía, las Industrias y los Recursos Naturales no Renovables. Este tipo de sistemas dentro del mercado es conocido como ANPR (Automatic Number Plate Recognition en inglés) y de manera general se basa en dos algoritmos; uno que corresponde a la detección de vehículos y el otro al reconocimiento de las placas.

Es evidente que para el desarrollo del presente Trabajo de Titulación se hace la detección de vehículos y reconocimiento de placas, para lo cual se deben aplicar técnicas de procesamiento digital de imágenes, las cuales están inmersas dentro de la Visión Artificial.

Finalmente, para desarrollar el sistema se utilizó una metodología de programación común para desarrollar cualquier tipo de sistemas, conocida como metodología con enfoque en cascada. Debido a que se trata de un proyecto basado en Visión Artificial, para lo cual se requiere realizar mucha más investigación de tipo consulta que programación.

Summary

This Project Thesis is focused on the use and application of one of the subfields of the well-known popular Artificial Intelligence and, commonly as is Artificial Vision or Computer Vision, which basically consists on the pattern recognition, image digital processing, etc.

If we make an analogy, in more specific terms we would be simulating the functions performed by the human eye, like the case of identifying and keeping track of an object, that is why emerged the reason to develop a system capable of recognizing emerged the plates of vehicles that enter in the Área de la Energía, las Industrias y los Recursos Naturales no Renovables. This kind of systems in the market area is known as ANPR (Automatic Number Plate Recognition) and generally it is based on two algorithms; one corresponding to vehicle detection and the other to the recognition of the plates.

It is clear that for the development of this project Thesis it is done the vehicle detection and recognition of plates, for which we need to apply some techniques of digital image processing, which are embedded within the Artificial Vision.

Finally, to develop the system common programming methodology we used a common methodology system, which is known as cascade approach methodology. Due to the project is based on artificial Vision, which is necessary to do a lot more research about programming.

(Certificado de traducción por Lic. Pastora Isabel Sarango Lozano (Ver Anexo 5))

Índice de Contenidos

Índice General

Índice de Figuras

Índice de Tablas

c. Introducción

El sentido de la vista permite al ser humano conocer el medio que lo rodea, relacionarse con sus semejantes y contar con los elementos adecuados para captar e interpretar las señales provenientes de ellos. Según Aristóteles la “visión” es saber que hay y donde mediante la vista. Gibson dice que la “visión” es recuperar de la información de los sentidos (vista) propiedades validas del mundo exterior. Marr opina que la “visión” es el proceso que produce, a partir de las imágenes del mundo exterior, una descripción que es útil para el observador y que no tiene información irrelevante”. De hecho se calcula que más de 70% de las tareas del cerebro son empleadas en el análisis de la información visual. Las imágenes visuales obtenidas a través del ojo proporcionan información sobre el color, la forma, la distancia, posición y movimiento de los objetos. En la actualidad todas las disciplinas científicas emplean útiles gráficos para transmitir el conocimiento. Por ejemplo en el desarrollo de software los ingenieros utilizan los modelos Lenguaje de Modelo Unificado (UML) para describir la arquitectura de un sistema informático, en ingeniería electrónica se emplean esquemas de circuitos, a modo gráfico para describirlos. Las grandes empresas utilizan imágenes en alta resolución para realizar publicidad y hacer conocer sus productos. Se podría hacerlo mediante texto, pero para la especie humana resulta mucho más eficiente procesar imágenes que texto, dando una mayor interpretación al conocido refrán popular de “Una imagen vale más que mil palabras” que tiene mucho que ver con los aspectos cognitivos de la especie humana. Hoy en día, se carece de una teoría que explique cómo los humanos perciben el exterior a través de la vista.

Es así que han aparecido términos tecnológicos como visión artificial o visión por computador. Uno de los grandes eventos de las últimas décadas fue la invasión de los medios digitales dentro de todos los aspectos de la vida cotidiana, la computadora ha tomado una gran importancia en el procesamiento de datos, entre ellos las imágenes.

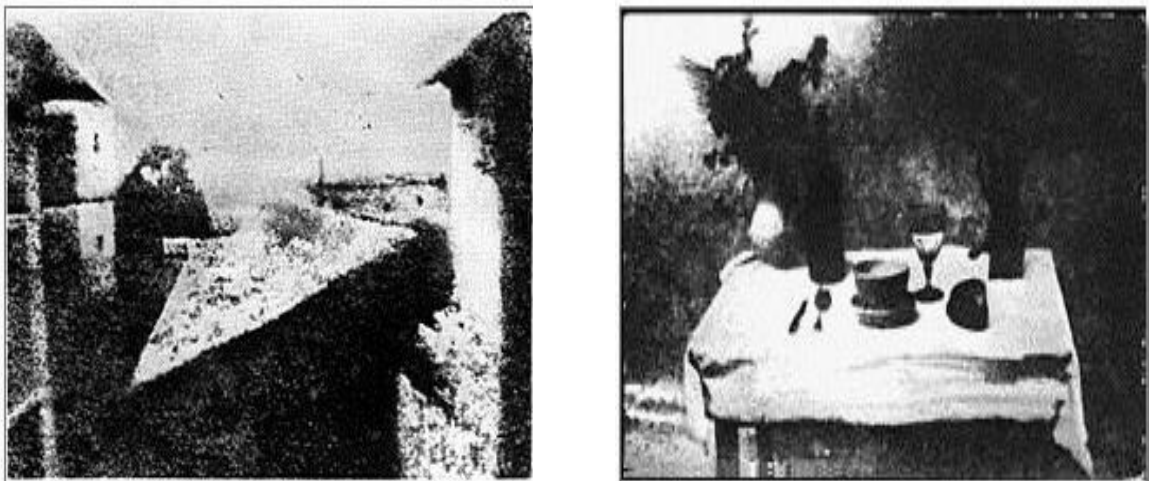
La estructura formal del informe de tesis se ajusta a las disposiciones legales que constan en el Reglamento del Régimen académico de la Universidad Nacional de Loja, la cual contiene: Titulo, Resumen, Introducción, Revisión Literaria, Materiales y Métodos, Resultados, Discusión, Conclusiones y Recomendaciones.

d. Revisión de Literatura

Capítulo 1: Visión Artificial

1.1. Antecedentes

En el año 1826 el químico francés Niépce (1765-1833) llevó a cabo la primera fotografía, colocando una superficie fotosensible dentro de una cámara oscura para fijar la imagen. Posteriormente, en 1838 el químico francés Daguerre (1787-1851) hizo el primer proceso fotográfico práctico. Daguerre utilizó una placa fotográfica que era revelada con vapor de mercurio y fijada con trisulfato de sodio [1], en la Figura 1 se pueden apreciar las primeras fotografías.



a) b)

Figura 1: Primeras Fotografías: a) Niépce 1826, “Punto de vista desde la ventana del Gras”. Es la fotografía más antigua que se conoce. b) Niépce. Este bodegón fue considerado durante mucho tiempo como la foto más antigua.

Desde que se inventó la fotografía se ha intentado extraer características físicas de las imágenes. La Fotogrametría dio sus primeros pasos desde imágenes capturadas en globos. La Astronomía avanzó enormemente con el análisis de imágenes recibidas por los telescopios. El análisis de radiografías transformó la Medicina. Se podrían citar muchos más ejemplos que durante décadas han transformado la percepción de la Ciencia con el procesamiento de las imágenes, algunas veces por separado y otras de

forma multidisciplinar [1], en la Figura 2 se muestran fotografías de algunas de las aplicaciones referentes al procesamiento de imágenes.

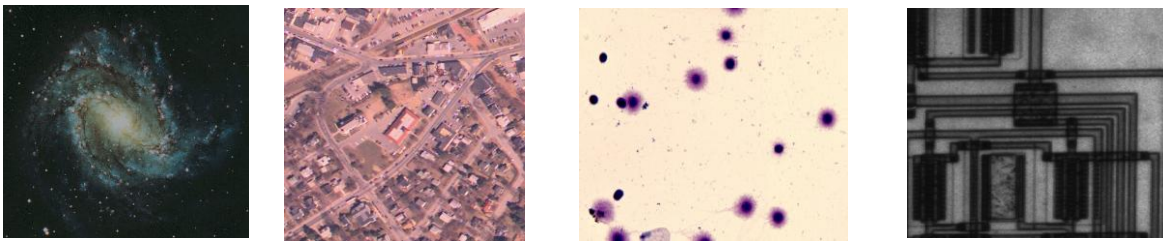


Figura 2: Aplicaciones del procesamiento de imágenes: a) Astronomía, b) Fotogrametría, c) Medicina, d) Industria.

Sin embargo, el momento histórico que hace que estas técnicas confluyan y den un cuerpo de conocimiento propio, surge en la década de los 80. La revolución de la Electrónica, con las cámaras de vídeo CCD y los microprocesadores, junto con la evolución de las Ciencias de la Computación hace que sea factible la Visión Artificial [1].

Una de las primeras aplicaciones de las imágenes digitales se dio en la industria del periódico cuando se enviaron fotografías a través de cable submarino entre Londres y Nueva York, en la primera parte de la década de los veintes. El sistema Bartlane tomaba cerca de tres horas y constaba de equipo de impresión especializado que codificaba las fotografías para su envío por cable y las reconstruía del otro lado [2].

Figura 3: Fotografía digital de 1921 obtenida con una impresora telegráfica.

Esta técnica se abandonó rápidamente, favoreciendo otro tipo de reproducción fotográfica basada en cintas que se perforaban en la terminal telegráfica receptora. Las primeras computadoras suficientemente potentes para desarrollar tareas de PDI

(Procesamiento Digital de Imágenes) significativas aparecieron en los comienzos de los sesentas, junto con el programa espacial estadounidense. El laboratorio Jet Propulsión de Pasadena California inició los trabajos en 1964 cuando un equipo procesó varias fotos de la luna transmitidas por el Ranger 7, para corregir varios tipos de distorsión que producía la cámara de a bordo (Figura 4). Estas investigaciones servirían de base para desarrollos posteriores en subsecuentes misiones como Surveyor, Mariner y Apollo.

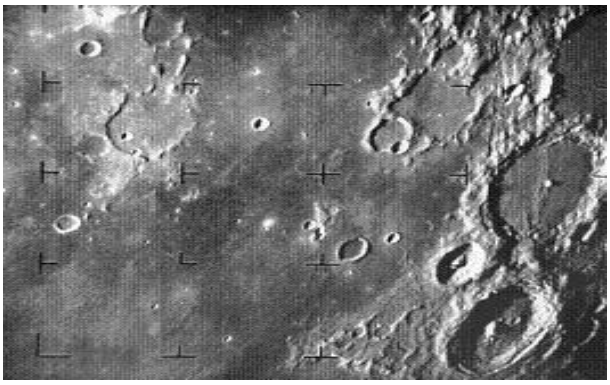


Figura 4: La primera fotografía de la luna de una nave espacial estadounidense.

1.2. Definiciones

En el siguiente apartado se abordarán algunos conceptos básicos sobre la Visión Artificial, los cuales fueron de gran importancia para el desarrollo del presente proyecto, y además nos permitieron tener una idea general de lo que es la Visión Artificial.

1.2.1. ¿Qué es la Visión Artificial?

De manera general, la visión representa una de las formas de sensores para la percepción más importantes que poseen los seres humanos y también los seres biológicos, para poder detectar e identificar los objetos en el medio que los rodea, sean éstos de acuerdo a su forma, color, textura u otras maneras.

En cambio la Visión Artificial, comúnmente conocida como "Visión por Computadora", representa un conjunto de todas aquellas técnicas y modelos que nos permiten la

adquisición, procesamiento, análisis y explicación de cualquier tipo de información espacial del mundo real obtenida a través de imágenes digitales [3].

También podríamos decir, que la visión por computador es la capacidad que tiene una máquina para ver el mundo que lo rodea, para reducir la estructura y las propiedades del mundo tridimensional en base a una o más imágenes bidimensionales. Pero debemos considerar que la introducción de habilidades en una máquina como son: la adquisición de imágenes, detección y reconocimiento de objetos, representan procesos muy complejos, debido a que no existen algoritmos eficaces que se adapten a todo tipo de problema y en cualquier ambiente propicien el reconocimiento efectivo de todo tipo de objetos. Pese a todo aquello debemos reconocer que gracias a la visión artificial o por computadora, se ha logrado liberar al hombre de las tareas rutinarias muy trabajosas y también peligrosas que desde hace tiempos remotos ha tenido que enfrentar [4].

1.3. ¿Para qué se utiliza?

En los últimos años, los sistemas de visión artificial han evolucionado tanto tecnológicamente como en la propia filosofía del sistema de visión. Esto ha implicado cambios sustanciales en la forma de interpretar la visión como una herramienta estándar para el análisis de procesos [5].

Así por ejemplo tenemos:

- Inspección continua
- Inspección del 100% de los productos
- Criterios constantes
- Tiempo real
- Análisis de errores
- Errores de aspecto, color, forma, etc…

En la **Figura 5** podemos ver como se aplica la visión artificial en los robots en la parte industrial.

Figura 5: Robot industrial fabricante de autos.

1.3.1. ¿Dónde se utiliza?

Existen mercados emergentes (Ver Figura 6) como:

- Transporte / Guiado de vehículos
- Identificación facial, Biometría
- Industria del cine
- Inspección de madera
- Aplicaciones en el rango no-visible
- Sistemas 3D
- Aplicaciones de tráfico
- Marketing
- Ocio y entretenimiento
- Entretenimiento.

Figura 6: Aplicaciones de la visión Artificial.

1.3.2. ¿Desde cuándo se utiliza y quién utiliza la Visión Artificial?

Los sistemas de visión hasta hace poco tiempo eran una apuesta por la innovación y por la calidad en el proceso de fabricación. En la actualidad un sistema de visión es una necesidad referente a la calidad del producto y a la reducción de costes a nivel productivo. La innovación sigue existiendo pero a un nivel de aplicación más exigente. En muchos casos la visión viene impuesta por el cliente final [5].
En el pasado, el usuario típico de un sistema de visión era un experto en la materia. Tanto los sistemas de visión como sus aplicaciones estaban orientados a un perfil muy técnico y con amplios conocimientos tanto en el hardware específico como en el proceso a controlar. Actualmente existen productos de visión que se pueden poner en funcionamiento por usuarios no expertos con una cierta formación respecto al producto. Los interfaces gráficos y la simplicidad en las cámaras han orientado la visión a este tipo de procesos a controlar [5].

1.3.3. ¿Cómo se implementa?

Los componentes de un sistema de visión siguen siendo los mismos (iluminación, lente, framegrabber, cámara, software, etc), las prestaciones de los componentes eran limitadas y el coste del producto de visión muy alto con respecto a las ventajas que presentaban.

Las prestaciones tanto del hardware como del software van en progresivo aumento, aparecen sistemas compactos y sistemas LowCost que resuelven muchas de las aplicaciones actuales. Existe hardware dedicado para la visión no utilizado anteriormente que nos permite llegar a aplicaciones no tenidas en cuenta hasta la fecha [5].

1.4. Visión Artificial en tiempo real

Sus inicios los podemos encontrar hace unos 30 años, y han evolucionado muy rápidamente ayudados a su vez por el rápido avance de los ordenadores y su potencia de cálculo. En el pasado más reciente no era posible hacer los procesos en tiempo real debido a que los ordenadores no eran lo suficientemente rápidos para realizar los cálculos con las imágenes. Incluso hasta hace aproximadamente cinco años no era

posible realizar la visualización de las imágenes debido al ancho de banda del bus ISA (Arquitectura Estándar de la Industria).

Los procesos en tiempo real en ese momento se debían hacer en procesadores DSP (Procesador de señal digital) a bordo de las placas, con el fin de poder alcanzar las velocidades requeridas, para la mayoría de aplicaciones. Con la llegada del bus PCI (Interconexión de componentes periféricos) y PCI Express, así como con la rápida evolución de los procesadores de los PC se ha conseguido visualizar las imágenes en tiempo real y realizar la mayoría de procesos en tiempos suficientemente cortos, como para que puedan resolver aplicaciones de visión en entornos científicos e industriales, con los resultados esperados en su justo tiempo.

Hasta hace pocos años la implementación de sistemas de visión requería un extenso conocimiento del software de bajo nivel y del hardware de visión. Actualmente, el panorama ha cambiado radicalmente, ya que se encuentran disponibles numerosos entornos de programación escalables y fáciles de utilizar, que combinados con los nuevos procesadores hacen muy fácil la implementación de un sistema de visión.

La base del software de un sistema de visión es la interpretación y análisis de los píxeles. El resultado final puede ser, desde la medida de una partícula, a la determinación o lectura de una serie de caracteres (OCR), pasando por cualquier otro proceso que podamos imaginar sobre las imágenes.

Dependiendo si la aplicación se realiza en entorno industrial o científico los pasos a seguir en un sistema de visión serán algo distintos, mientras que en las aplicaciones industriales la velocidad a la que se realizan las medidas es fundamental, ya que se deben evaluar todas las piezas producidas en tiempo real, en las aplicaciones científicas la se busca la determinación de los resultados en imágenes más complejas [6].

1.5. Aplicaciones Software de la Visión Artificial

En la actualidad existen un sin número de aplicaciones de la visión artificial, debido a que permite obtener y analizar relevante información espectral que está asociada con el color, información espacial que tiene que ver con los aspectos como posición y forma (dimensiones) y temporal que está asociada con los aspectos estacionarios sobre distintos objetos [7].

En el mercado existen herramientas y paquetes de software cuya finalidad principal son las aplicaciones industriales, a continuación se detallan algunos ejemplos:

- **NI LabVIEW.-**. Proporciona extensas herramientas que se necesitan para construir cualquier aplicación de medida o control en mucho menos tiempo, es el entorno de desarrollo ideal para la innovación, descubrimiento y resultados acelerados. Combina la potencia del software LabVIEW con hardware modular y reconfigurable para resolver la creciente complejidad involucrada de proporcionar sistemas de medida y control a tiempo y dentro del presupuesto [8].

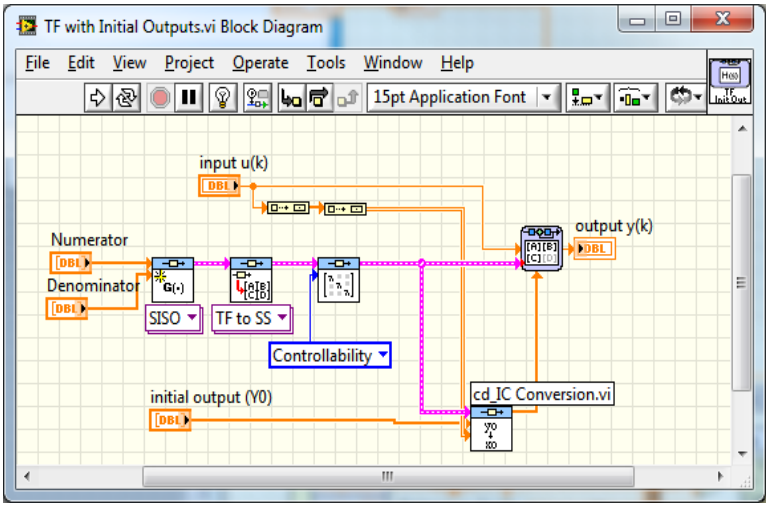


Figura 7: Interfaz del software NI LabVIEW.

- **Image Processing Toolbox.-** Ofrece un conjunto completo de los algoritmos de referencia estándar, funciones y aplicaciones para el procesamiento de imágenes, el análisis, la visualización y el desarrollo de algoritmos. Puede llevar a cabo el análisis de imágenes, segmentación de imágenes, mejora de imagen, reducción de ruido, transformaciones geométricas, y el registro de imágenes [9].

- **Matrox Imaging Library (MIL).** Es una colección de herramientas de software para el desarrollo de la visión artificial, análisis de imágenes y aplicaciones de software de imágenes médicas. MIL incluye herramientas para cada paso en el proceso: desde la viabilidad de aplicación, para la creación de prototipos, hasta el desarrollo y finalmente la implementación [10].

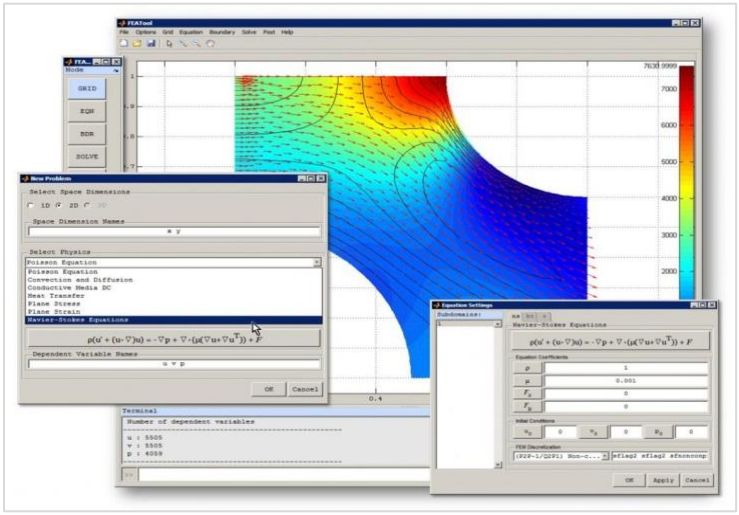


Figura 8: Toolbox de Matlab para Visión Artificial.

Figura 9: Software Matrox Imaging Library (MIL).

- **Software NI Visión.-** ofrece cientos de algoritmos de procesamiento de imagen de primera clase y las funciones de adquisición de imágenes que se pueden utilizar a través de toda la cartera de hardware de visión de NI para resolver cualquier aplicación de visión artificial [11].

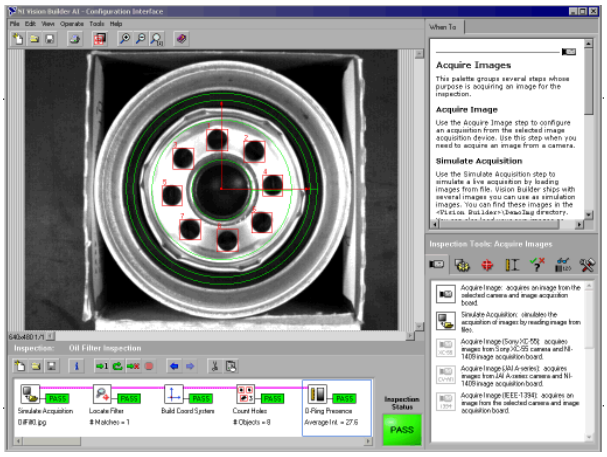


Figura 10: Interfaz Software NI Visión.

1.6. Campos de Aplicación de la Visión Artificial

Como ya se mencionó anteriormente, la Visión Artificial es parte de la Inteligencia Artificial y que a su vez tiene algunos campos de aplicación debido a que es multidisciplinaria. Por tal motivo se detallan a continuación algunos campos en los cuales tiene lugar la Visión Artificial, lo cual nos permitirá diferenciar nuestro campo de aplicación de los demás.

1.6.1. Navegación Robótica

En este caso, la visión es un elemento de un Sistema Multisensorial. La información procedente de la visión es validada, comparada y finalmente integrada con el resto de la información proporcionada por otro tipo de sensores.

Figura 11: Navegación de robots en convoy utilizando Visión Artificial.

Para la navegación en robótica se recurre generalmente a técnicas de visión estereoscópica con el fin de poder reconstruir la escena 3-D. Si a esto se le añade algún módulo de reconocimiento 3-D con el fin de identificar la presencia de determinados objetos, hacia los que debe dirigirse o evitar, tanto mejor. La utilización del movimiento basado en la visión constituye un magnífico recurso puesto que el propio sistema está ya de hecho en movimiento. Naturalmente, cualquier otra información que pueda extraerse con ayuda de la visión puede proporcionar una gran ayuda para conseguir el movimiento del robot. Además cabe recalcar que en esta aplicación, la visión artificial no es de uso exclusivo, más bien podría aplicarse en lo que es el guiado automático de máquinas o la detección de objetos en movimiento [12].

1.6.2. Biología, Geología y Meteorología

En el campo de la biología podríamos distinguir entre aplicaciones microscópicas y macroscópicas. Para aplicaciones microscópicas la estrategia a seguir es, mediante técnicas de segmentación orientadas a regiones, la identificación de diferentes propiedades de las células (color, forma, tamaño, etc.) o bien, para contar el número de microorganismos o células presentes en una muestra de la que se ha obtenido una imagen [12].

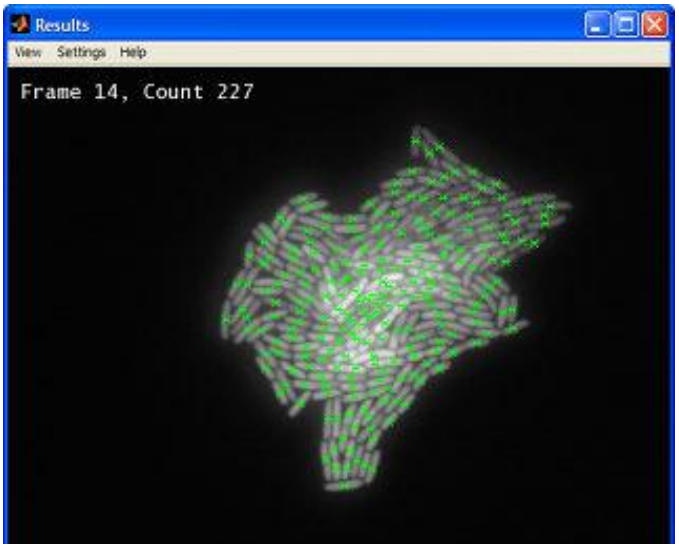


Figura 12: Visión Artificial aplicada en Biología

En cuanto a las aplicaciones macroscópicas de lo que se trata es de la identificación de determinados tipos de texturas en vegetales, o características de diferentes áreas naturales por rasgos característicos, como color, grado de floración o crecimiento de especies por diferencia de imágenes por ejemplo. En concreto este último método también se emplea en la geología para detectar movimientos de regiones [12].

Puede resultar de interés el hecho de identificar el grado de floración de un determinado cultivo, por ejemplo, de un campo de girasoles. Para ello se extrae de la imagen la componente de color amarillo, para luego proceder a binarizar la imagen y obtener el número de píxeles blancos o negros, que nos permite obtener el porcentaje de floración. La transformada de Fourier puede resultar útil, por ejemplo para determinar la naturaleza de una determinada formación geológica o de una especie biológica.

En Meteorología podríamos utilizar las técnicas de detección y predicción del movimiento, las cuales nos permiten observar, por ejemplo, la evolución de ciertas masas nubosas, u otros fenómenos meteorológicos, a través de imágenes recibidas vía satélite [12].

1.6.3. Medicina

En el campo de la medicina, la visión artificial tiene muchas aplicaciones en las que aparece el procesamiento de imágenes, a menudo orientadas hacia el diagnóstico de dolencias o enfermedades (Figura 13), entre las que se incluyen radiografías, resonancias magnéticas, tomografías etc.

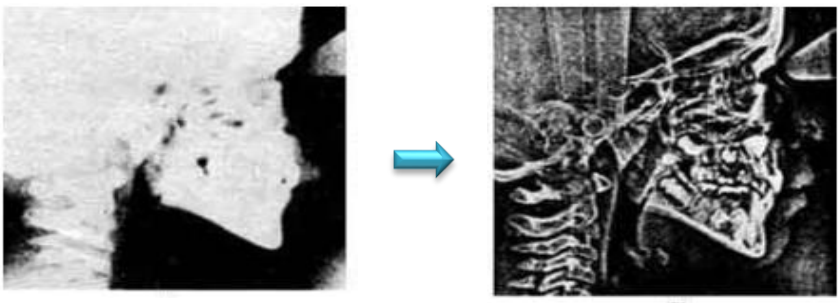


Figura 13: Cefalograma: Procesamiento de imágenes en el diagnóstico de enfermedades.

Desde hace algunos años, se está desarrollando el "Brain Port Vision", se trata de un instrumento, en fase de investigación, pero que probablemente llegue pronto al mercado. Todavía no fue aprobado por FDA, pero se ha ensayado en grupos de pacientes voluntarios con amaurosis con resultados alentadores.

Este dispositivo permitiría a pacientes con ceguera, orientarse, identificar objetos estáticos y en movimiento e incluso poder leer. La información visual es recogida por una cámara de video digital, escondida en un par de anteojos, esta imagen, es sometida a un transductor, que la convierte en impulsos eléctricos de baja intensidad y derivada a una plaqueta (que se coloca en la lengua) conteniendo 144 micro electrodos [13].

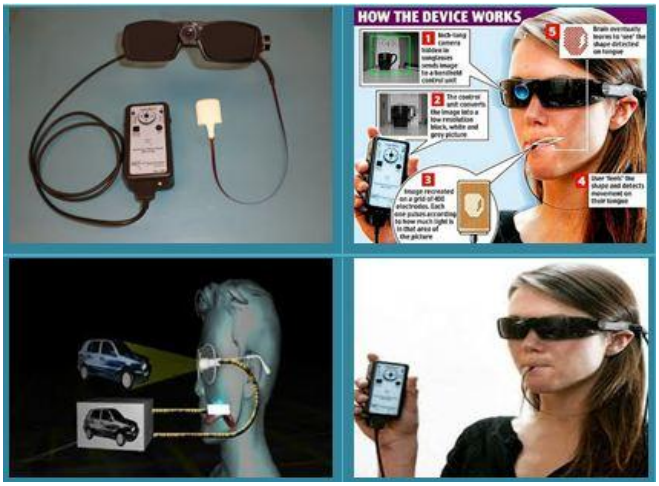


Figura 14: Visión Artificial para personas no videntes

1.6.4. Identificación de construcciones, infraestructuras y objetos en escenas de exterior

Mediante imágenes aéreas o de satélite se puede determinar la presencia de ciertas regiones a través de la segmentación de las mismas así como detectar la presencia de ciertas construcciones (edificios) o infraestructuras (carreteras, canales, puentes) a través de técnicas de extracción de bordes o contornos combinadas con la segmentación de regiones [12].

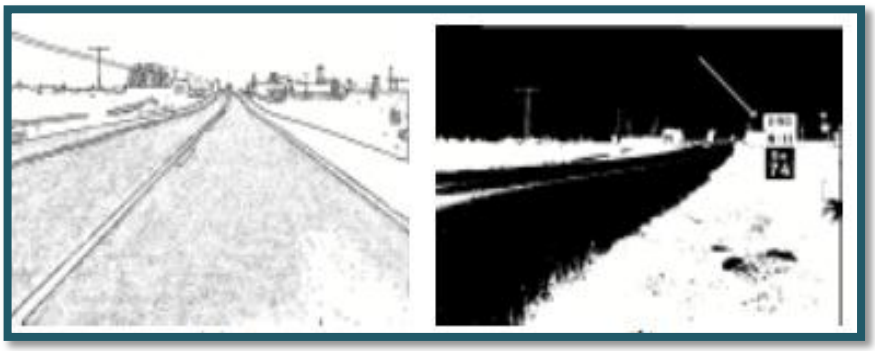


Figura 15: Identificación del punto kilométrico, mediante técnicas de reconocimiento de caracteres

Un ejemplo puede ser la reconstrucción de tejados de casas urbanas (por su estructura geométrica mejor definida que los rurales) por medio de relaciones topológicas y geométricas teniendo una base de datos de modelos a partir de los cuales se reconstruye el tejado que está siendo inspeccionado.

1.6.5. Reconocimiento y clasificación

Una aplicación de la visión artificial puede ser la clasificación de objetos. De acuerdo a su tamaño y en su caso el recuento de los mismos. Por ejemplo, para contar monedas en función del área de la moneda [12].

Figura 16: Reconocimiento de monedas y etiquetado de su valor.

El reconocimiento de huellas dactilares también es posible mediante visión artificial, para lo cual se utiliza un conjunto de máscaras para obtener un vector numérico representando cada huella digital como un vector de atributos multidimensional, luego mediante optimización se realiza la clasificación [12].

Figura 17: FPrint: Software para la identificación de huellas dactilares.

Para la detección de caras existen infinidad de aplicaciones, tanto para seguridad como para otro tipo de aplicaciones, como por ejemplo en cámaras de fotos, para que el enfoque automático sepa donde tiene que enfocar [14].

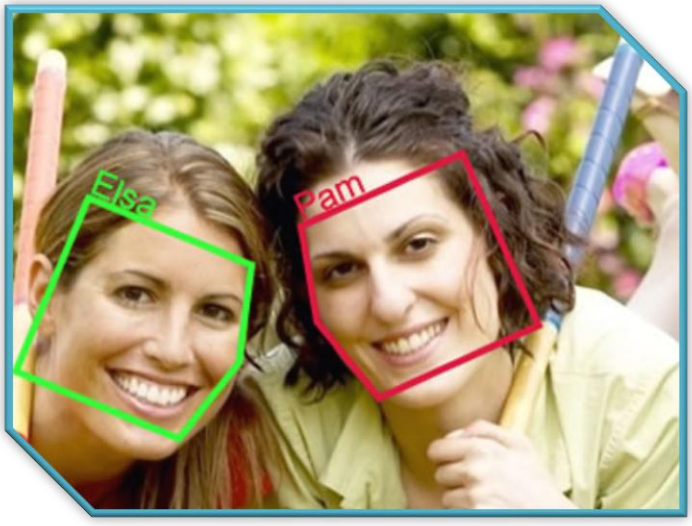


Figura 18: Visión Artificial aplicada en el reconocimiento facial de personas.

Quizás una de las áreas de aplicación más importantes sea el Reconocimiento Óptico de Caracteres (en inglés OCR). Existe mucha información bibliográfica sobre esta área de aplicación de la Visión Artificial.

Figura 19: Visiomat: Software utilizado para la detección de matrículas vehiculares.

Otra aplicación que se está teniendo una gran expansión tanto para aplicaciones de vídeo-vigilancia como de interacción persona-computador es la del seguimiento de personas en secuencias de imágenes y la interpretación automática de las actividades que desarrollan. Para la realización de estas tareas se combinan métodos de seguimiento visual con el reconocimiento de patrones y aprendizaje [14].

Figura 20: Seguimiento de actividades humanas.

1.6.6. Inspección y control de calidad

La inspección de un objeto manufacturado puede tomar muchas formas, que podría involucrar las siguientes tareas:

a. verificar la presencia de cada característica esperada
b. verificar las dimensiones de esas características (por ejemplo radio y longitud de un cilindro)
c. verificar las interrelaciones entre características (por ejemplo distancias entre centros de gravedad y ángulos entre planos) [12].

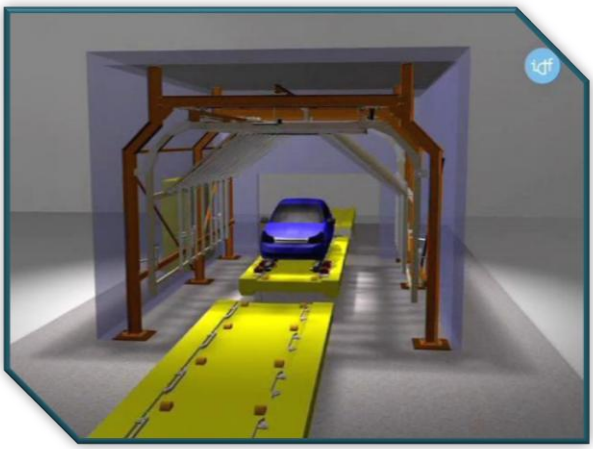


Figura 21: Control de calidad aplicando Visión Artificial.

La inspección en el sentido más amplio se refiere a la verificación de si un objeto cumple con determinados criterios. Esto implica comparar el objeto con algún objeto modelo que describe las características relevantes del objeto. Para muchos tipos de datos existen tolerancias definidas dentro de las cuales las medidas realizadas pueden considerarse como aceptables [12].

Una de las finalidades de los controles de calidad consiste en detener la producción de algún producto si el sistema de producción comienza a generar productos que no cumplen con las normas estándares generales. Imaginemos que para la fabricación de galletas se exige que las mismas tengan un tamaño con unos márgenes de confianza, mediante la segmentación de la misma se determina el área, pero si además es preciso que la forma sea circular, la transformada de Hough nos facilita el cumplimiento de dicho requisito [12].

1.6.7. Cartografía

Mediante el uso de imágenes estereoscópicas aéreas o de satélite es posible obtener las elevaciones del terreno a través de técnicas de correspondencia basadas en el área [12].

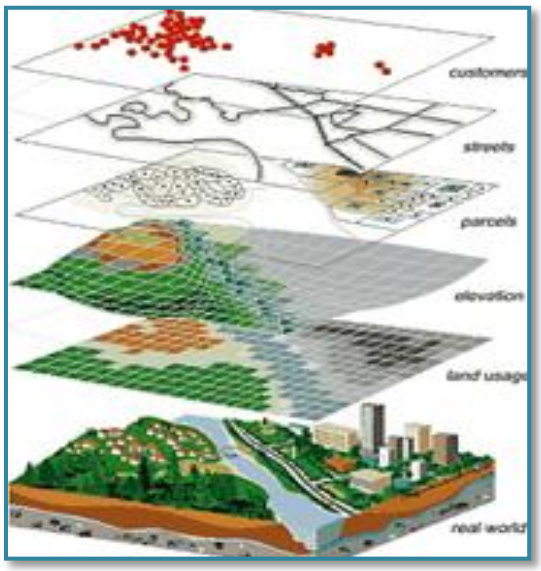


Figura 22: Visión Artificial aplicada en la cartografía.

Por otro lado, de cara a la elaboración de los catastros, particularmente en las zonas rurales, la utilización de imágenes aéreas permiten una fácil identificación de las diferentes parcelas y sus delimitaciones tras el correspondiente tratamiento de las imágenes mediante técnicas de extracción de bordes y regiones, así como sus descripciones tal y como se ha visto a lo largo de los capítulos 6 a 9. Si los sensores que captan las imágenes están perfectamente calibrados se puede llegar a determinar la superficie real de las parcelas basándose en el área de las imágenes medida en píxeles, simplemente aplicando la correspondiente función de transformación [12].

1.6.8. Foto interpretación

La fotointerpretación es la ciencia que trata del análisis de las imágenes por parte de un experto para extraer de ellas la información de interés o relevante, por ejemplo, ver las construcciones existentes en una determinada imagen de satélite o una imagen aérea. Cuanto mejor sea la calidad de la imagen mejor será el resultado del análisis. Por ello las imágenes pueden ser tratadas con todas las técnicas encaminadas a

mejorar la calidad de la imagen. Además, puede ocurrir que se desee obtener una imagen en color por combinación de las bandas roja, verde y azul procedentes de un sensor multi espectral de satélite, en este caso se puede recurrir a las diferentes técnicas [12].

Figura 23: Foto interpretación de la estructura del Castelo, sobre el mapa de pendientes, y sobre la foto de satélite de Google Earth.

1.7. Componentes de la Visión Artificial

A pesar de que cada aplicación basada en visión artificial tiene sus elementos para formar el sistema, se puede decir que existe una base común de etapas entre ellas. No es necesario cubrir todas en una implementación específica.

Muchas veces únicamente es necesario tener un subconjunto de fases que se presentarán a continuación. No obstante la exposición muestra un encadenamiento temporal de una etapa sobre otra, para facilitar la comprensión y puesta en práctica entre las distintas fases que existen dentro de la Visión Artificial.

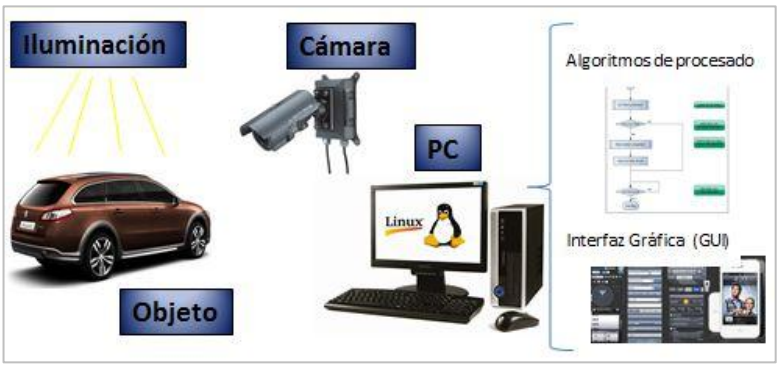


Figura 24: Componentes de un Sistema de Visión Artificial.

1.7.1. Iluminación

Este componente es una parte muy crítica de un sistema de visión artificial, ya que una adecuada imagen de entrada al sistema permite reducir los costes computacionales del procesado posterior. En cada caso es fundamental tener en cuenta para el diseño de este sistema el hecho de cómo es la superficie a iluminar, del tipo de materiales que se están tratando [15].

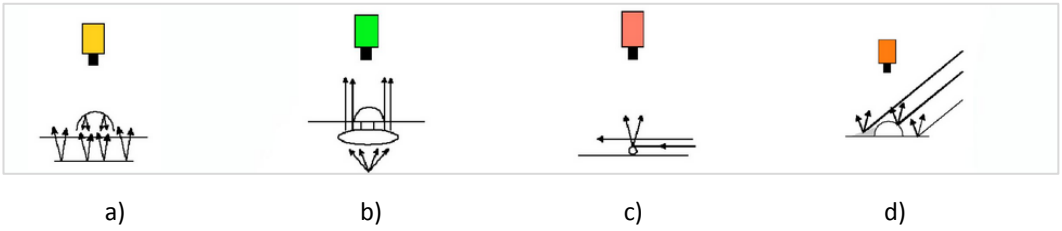


Figura 25: Tipos de iluminación: a) Iluminación posterior difusa, b) Iluminación posterior direccional, c) Iluminación frontal oblicua, d) Iluminación frontal direccional

1.7.2. Cámaras

Es muy importante que para cada aplicación que se desarrolle, dependiendo de las necesidades de cada empresa se debe seleccionar el tipo de cámaras y óptica apropiadas. En el mercado existen algunos tipos de cámaras, por ejemplo:

- Lineales.
- Matriciales
- En color.
- En blanco y negro.

Y también existen algunos tipos de lentes:

- Teleobjetivos.
- Gran angular.
- Telecéntricas.

1.7.3. Ordenador o PC

Es sin lugar a dudas uno de los componentes de un Sistema de Visión Artificial más importantes, debido a que es la parte pensante del sistema, se encarga no solo de recoger y mostrar las imágenes capturadas, si no de procesarlas para llevar a cabo su objetivo. Entre las tareas que realiza están:

- Recibir todas aquellas señales de sincronización para que se pueda realizar correctamente la captura de imágenes.
- Realizar la lectura de las imágenes.
- Procesar los datos proporcionados por las cámaras para realizar el análisis de imagen.
- Realizar las interfaces gráficas de usuario
- Comunicar con los sistemas productivos, para detener el proceso en caso de la aparición de algún defecto
- Controlar el buen funcionamiento de todos los elementos hardware [15].

Capítulo 2: Procesamiento digital de imágenes

Dentro de la Visión Artificial se utiliza el procesamiento digital de imágenes, sea para realizar una detección de objetos o para realizar el reconocimiento de objetos. En este capítulo se abordan temas referentes al Procesamiento digital de imágenes.

1.1. Fundamentos de las imágenes

El campo del procesamiento digital de imágenes está fundamentado en las bases matemáticas y probabilísticas, pero la intuición y análisis humanos juegan un importante papel al momento de escoger una técnica u otra. Esta elección se basa usualmente en juicios visuales subjetivos.

Una imagen puede ser definida matemáticamente como una función bidimensional, **f(x, y),** donde **x** e **y** son coordenadas espaciales (en un plano), y **f** en cualquier par de coordenadas es la intensidad o nivel de gris de la imagen en esa coordenada. Cuando **x, y**, y los valores de **f** son todas cantidades finitas discretas, decimos que la imagen es una imagen digital. Una imagen digital se compone de un número finito de elementos, cada uno con un lugar y valor específicos. Estos elementos son llamados pels o píxeles [4].

En la **Figura 26** se muestra lo que es un píxel y en la **Figura 27** su representación en una matriz.

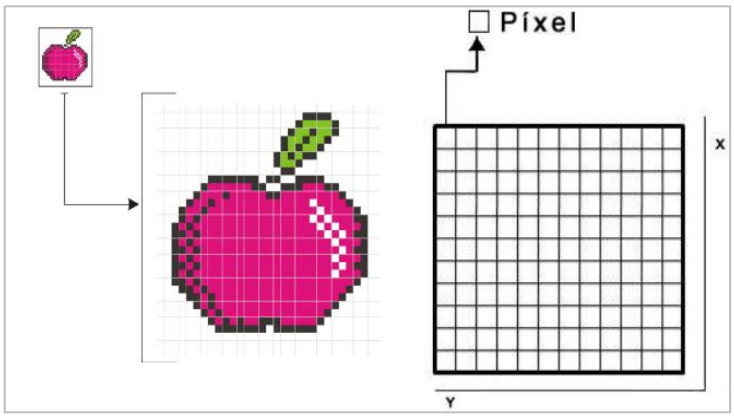


Figura 26: Píxeles: Los puntos de una imagen.

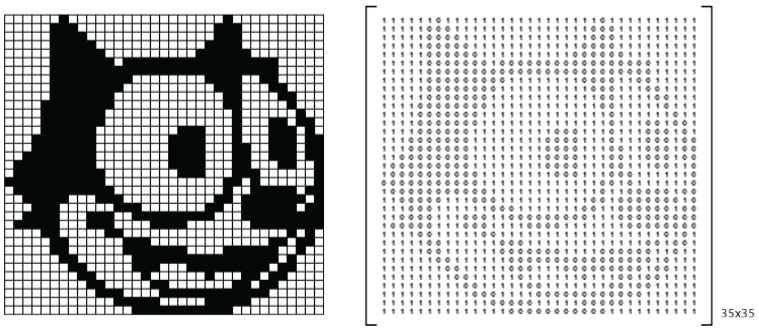


Figura 27: Imagen que representa una matriz de 35 x 35 (X * Y) píxeles.

1.1.1. Píxeles y resolución de una imagen

La imagen de mapa de bits o bitmap consiste en una cuadrícula, también llamada ráster, grilla o matriz, donde los píxeles se ordenan en las correspondientes celdas con valores correspondientes a los ejes **x** e **y**, horizontal y vertical. La representación de una imagen cualquiera se obtiene asignando y almacenando un único color en cada uno de los píxeles, es decir que cada uno tendrá un color y una ubicación dentro de la cuadrícula. A diferencia de la imagen vectorial, la imagen bitmap no puede escalarse sin consecuencias que alteren su aspecto y calidad. A la hora de escalar una de estas imágenes, habrá que tener en cuenta la resolución, el modo de color, y el formato de compresión entre otros factores [16].

La nitidez o la calidad visual en una imagen de mapa de bits están dada por su resolución. Esta nos permite apreciar mayor o menor cantidad de detalles. Para comprender este concepto es importante saber que el Píxel es la unidad mínima de color homogéneo que constituye una imagen digital bitmap, sea ésta una fotografía, un gráfico o un fotograma de video digital.

Las dimensiones altas y anchas de la superficie que ocupa una imagen se miden en píxeles. Por lo tanto, las dimensiones de la imagen estarán ligadas a la resolución y al peso, es decir, al espacio que ocupa la imagen en nuestro ordenador. Debemos tomar en cuenta sin embargo, que el píxel, a pesar de ser una unidad de medida, es un concepto inmaterial que no tiene una medida concreta. Sería, por lo tanto, imposible determinar si un píxel mide un milímetro o un centímetro.

En principio es solamente una unidad de medida variable por medio de la cual se dividen las celdas de una rejilla [16].

Es así que si tenemos una imagen de 200 × 100 píxeles sin saber su tamaño físico real, lo único que sabemos es que la hemos dividido en 20.000 celdas (píxeles). Sin embargo, cuando a esa imagen le asignemos una resolución, entonces sí sabremos el tamaño del píxel para esa imagen y para esa resolución dada. Por ejemplo, si decimos que la imagen consta de 100 píxeles por pulgada, tendremos 100 píxeles cada 2,54 centímetros. Es decir, que habrá 100 celdillas por cada 2,54 centímetros, con lo cual cada píxel medirá 0,254 milímetros. Pero si dijéramos que esa imagen tiene una resolución de 1 píxel por pulgada, entonces cada celda o píxel tomará el valor de 2,54 centímetros [16].

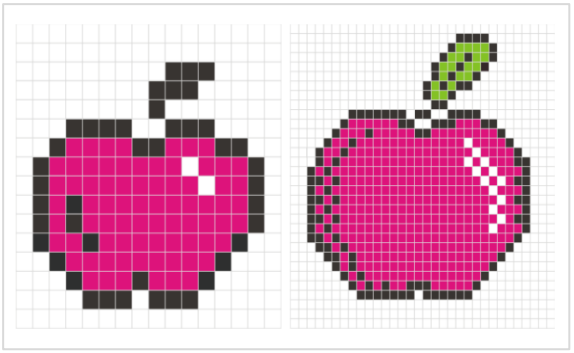


Figura 28: Resolución de una imagen.

La resolución de una imagen nos indica el número de píxeles que caben en cada unidad de longitud. Es decir que a mayor resolución e igual tamaño, mayor será la cantidad de píxeles y menor el tamaño de éstos y por lo tanto, la imagen tendrá mayor nitidez. Al comparar las imágenes vemos que ambas tienen el mismo tamaño pero el número de píxeles difiere una de la otra [16].

1.1.2. Elementos de la percepción visual

Como ya se mencionó anteriormente que la Visión Artificial más o menos es una simulación de lo que hace el ojo humano, es muy importante conocer los elementos que posee la percepción visual para conocer más a fondo lo que es la Visión Artificial y facilitar el desarrollo del presente proyecto.

A continuación se detallan cada uno de los elementos que corresponden a la percepción visual.

1.1.2.1. Estructura del ojo humano

El ojo humano es un sistema óptico que posee la capacidad de captar imágenes en alta resolución (nitidez), detectar objetos en movimiento, en base a la profundidad, el color, la distancia, etc.

Prácticamente es una esfera de un diámetro de 2.5 cm. En su parte frontal tiene un sistema óptico de precisión. La luz entra a través de la pupila, un pequeño orificio cuyo diámetro varía según la acción de los músculos del iris, acción que depende de la intensidad de la luz [4]**.**

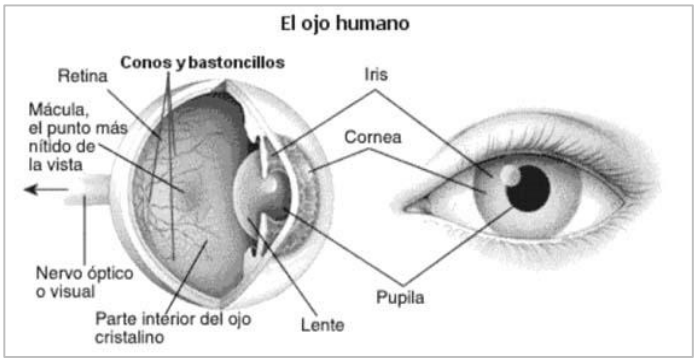


Figura 29: Fisiología del ojo humano.

1.1.2.2. Formación de imágenes en el ojo

Existe una diferencia entre una lente óptica ordinaria y el cristalino, donde este último es flexible, y su forma es controlada por la tensión de las fibras del cuerpo ciliar. Para enfocar objetos lejanos, se aplana, para enfocar objetos cercanos, se ensancha. La distancia entre el centro del cristalino y la retina (distancia focal), varía de aproximadamente 17mm a 14mm [4].

Gracias a esta información es posible calcular el tamaño del objeto reflejado en la retina.

El ejemplo de un observador que mira una palmera de 15m de altura a una distancia de 100m (al punto focal C). Si **h** es la altura en **mm** del objeto en la imagen retinal, por geometría obtenemos que:

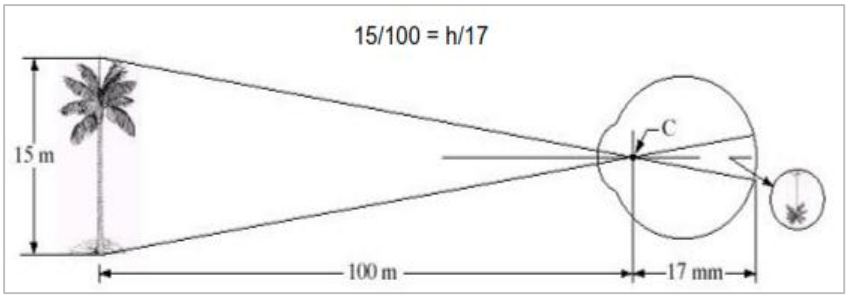


Figura 30: Percepción del tamaño de una imagen en la retina del ojo.

Por lo tanto h=2.55 mm, el tamaño de la palmera reflejada en la retina. La percepción tiene lugar cuando los diferentes receptores son excitados, es decir, transforman la energía radiante a impulsos eléctricos que son enviados al cerebro para su decodificación e interpretación [4].

Para tener un mejor entendimiento de cómo funciona un ojo frente a una cámara fotográfica, es posible realizar una analogía entre estos dos tipos de sensores y conocer cómo se relacionan sus elementos [17].

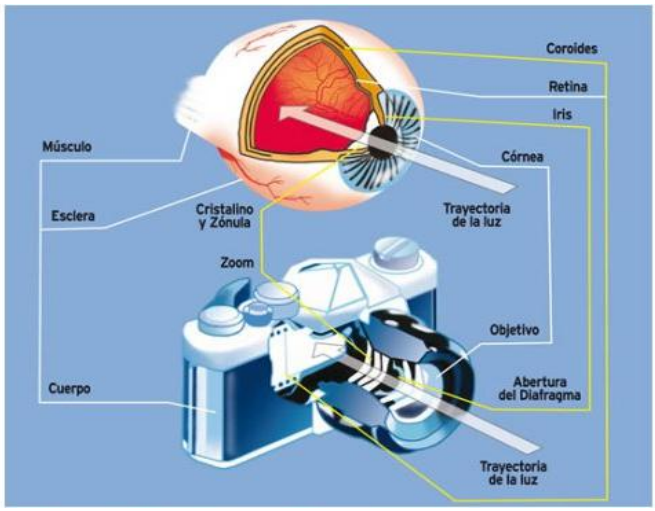


Figura 31: Comparación de la anatomía del ojo y partes de una cámara fotográfica.

Existen muchos ejemplos de fenómenos de percepción humana, conocidos como ilusiones ópticas, en las que el ojo da información que no existe o percibe erróneamente propiedades geométricas [2].

En la **Figura 32** se puede apreciar claramente un ejemplo de ilusión óptica:

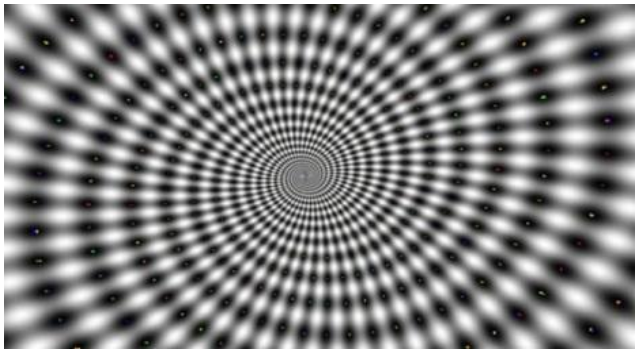


Figura 32: Ejemplo de ilusiones ópticas.

1.1.2.3. Percepción del color

Como ya se dio a conocer en la **Figura 29,** en lo que respecta a la fisiología del ojo, podemos decir que el ojo posee dos tipos de receptores: que son los bastones y los conos.

Los bastones cumplen la función de transmitir diferentes intensidades de gris y los conos permiten que el cerebro pueda percibir la tonalidad de los colores. Básicamente existen tres tipos de conos, el primero es sensible a la luz rojo/naranja, el segundo a la luz de color verde y el tercer cono a la luz azul/violeta. Cuando uno de los conos es excitado (estimulado), el cerebro percibe el color respectivo [4].

Cuando se realiza una percepción de color, el ojo no puede diferenciar entre un color amarillo espectral y alguna combinación de los colores rojo y verde, al igual que en los colores cian, magenta u otros de los colores espectrales intermedios. Y es por ello que el ojo puede percibir una amplia gama de tonalidades de colores en base a solo tres colores, conocidos en inglés como RGB (Red, Green, Blue) y en castellano Rojo, Verde y Azul. Se puede utilizar un prisma para poder analizar espectralmente cualquier color, utilizando los tres colores básicos: Rojo, Verde y Azul (RGB) [4].

En la **Figura 33** se puede observar los colores básicos y sus combinaciones o los que se derivan a partir de los mismos.

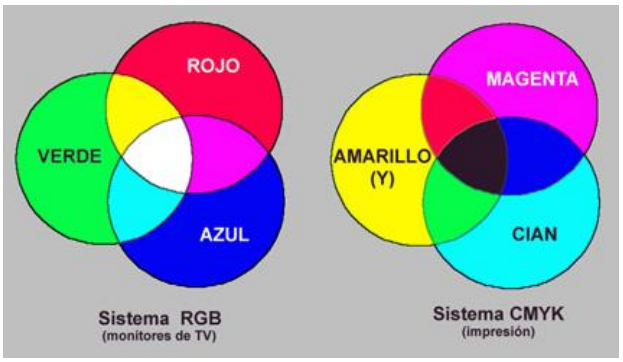


Figura 33: Colores básicos y sus combinaciones.

1.1.2.4. Tipos de imágenes y formatos

De manera general existen dos tipos de imágenes digitales; Imágenes vectoriales y las imágenes de mapas de bits.

Las **imágenes vectoriales** son imágenes constituidas por objetos geométricos autónomos (líneas, curvas, polígonos, etc.), definidos por ciertas funciones matemáticas (vectores) que determinan sus características (forma, color, posición, etc.) [18].

Las **imágenes de mapa de bits** están formadas por una serie de puntos (píxeles), cada uno de los cuales contiene información de color y luminosidad.

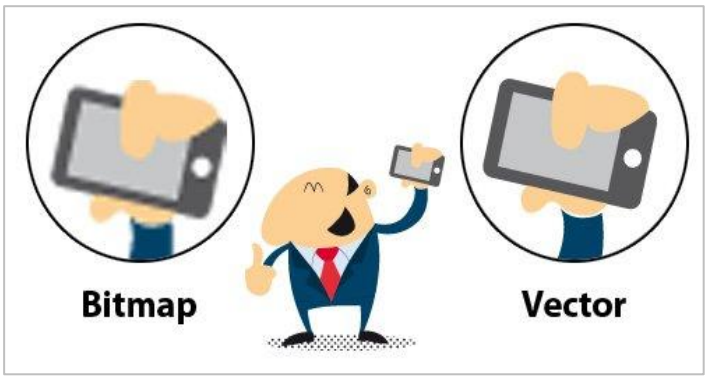


Figura 34: Formatos de imágenes digitales.

Las imágenes vectoriales se crean con programas de diseño o dibujo vectorial (Adobe Illustrator, Corel Draw, Inkscape) y suelen usarse en dibujos, rótulos, logotipos. Su principal ventaja es que una imagen puede ampliarse sin sufrir el efecto de *"*pixelado*"* que tienen las imágenes de mapa de bits al aumentarse [18].

Figura 35: Ampliación de imágenes.

En la imagen (vectorial) del ratón de la izquierda puede apreciarse que al ampliar una zona no hay perdida de detalle, mientras que en la fotografía del busto Nefertiti (mapa de bits) al ampliar mucho una zona, se observan los píxeles y la imagen se degrada.

Las imágenes de mapa de bits presentan una mayor gama de colores y de tonos que las vectoriales, por lo que son el tipo de imágenes usado en fotografía y, se crean con las cámaras de fotos, los escáneres y con programas de edición de imagen y dibujo (Adobe Photoshop, Gimp, etc.) Las imágenes mapa de bits generan archivos que ocupen mucha más memoria (bytes) que las imágenes vectoriales [18].

Para poder reproducirse o utilizarse en un ordenador u otros dispositivos las imágenes vectoriales y de mapa de bits se guardan en archivos o ficheros (conjunto de datos que se almacenan en algún medio, como un disco duro, DVD, flash memory) Cada archivo gráfico, se identifica además de por su nombre, por su extensión, que indica el tipo o formato de que se trata.

Algunos formatos de imagen vectorial son: AI (Adobe Illustrator), CDR (Corel Draw), DXF. (Autodesk), EMF, EPS, ODG Office Draw), SVG (Inkscape), SWF (Adobe flash), WMF (Microsoft).

Algunos formatos de mapa de bits son los siguientes:

- **BMP**. Formato introducido por Microsoft y usado originariamente por el sistema operativo Windows para guardar sus imágenes.

- **GIF**. Formato bastante antiguo desarrollado por Compuserve con el fin de conseguir archivos de tamaño muy pequeños. Admite solo 256 colores por lo que no es adecuado para imágenes fotográficas pero si es muy apropiado para logotipos, dibujos, etc. Permite crear animaciones (gif animado) y transparencias.

- **JPEG**. Es uno de los formatos más conocido y usado para fotografías digitales ya que admite millones de colores. Lo admiten la mayor parte de las cámaras fotográficas y escáneres y es muy utilizado en páginas web, envío de fotografías por correo electrónico, presentaciones multimedia y elaboración de vídeos de fotografías.

- **PNG**. Formato creado con el fin de sustituir a GIF. Utiliza sistemas de compresión gratuitos, y admite muchos más colores que GIF. También admite transparencias pero no animaciones. Al admitir más colores es posible crear imágenes transparentes con mayor detalle.

- **PSD.** Es el formato por defecto del editor de imágenes Adobe Photoshop y por tanto es un formato adecudo para editar imágenes con este programa y otros compatibles. Admite millones de colores, capas, canales.

- **RAW.** Formato “en bruto”. Esto quiere decir que contiene todos los píxeles de la imagen captada, tal y como se han tomado. Es el formato que ofrece la mayor calidad fotográfica y suele ser admitido por cámaras de gama media y alta (réflex, y compactas) indicadas para fotógrafos aficionados avanzados y profesionales.

Las cámaras que guardan las fotos en otros formatos (Tiff y JPEG) procesan la imagen captada para dar una interpretación de ella (balance de blanco, niveles de luminosidad, contraste) En el formato RAW, los píxeles no se procesan y se mantienen

en bruto para ser procesados posteriormente por un software específico conocido como “revelador RAW”.

Un archivo RAW, no sufre ninguna compresión, por lo que mantiene el máximo detalle de la imagen a costa de ocupar mucho espacio (Mbytes).
Los distintos fabricantes de cámara suelen llamar a los archivos RAW con distintos nombres. Por ejemplo, las cámaras Nikon los denominan archivos NEF.

- **TIFF**. Formato utilizado para el escaneado, la edición e impresión de imágenes fotográficas Es compatible con casi todos los sistemas operativos y editores de imágenes. Como PSD, admite millones de colores, capas, canales alfa y también lo incluyen algunas cámaras y la mayoría de los escáneres [18].

1.2. Adquisición de imágenes

La adquisición o captación de una imagen se realiza a través de sensores que se encuentran dentro de una cámara. Estos sensores son componentes sensibles a la luz que modifican su señal eléctrica en función de la intensidad luminosa que perciben.

Los tipos de imágenes que nos interesan se generan por una combinación de una fuente de “iluminación” y la reflexión o absorción de energía de esta fuente por parte de los elementos de la escena [4].

Esta “iluminación” puede ser también radar, infrarrojo, o alguna fuente que no es parte del Espectro Electro Magnético (EEM), como ultrasonido. Dependiendo de la naturaleza de la fuente, la energía “luminosa” puede ser reflejada o transmitida por los objetos.

En algunas aplicaciones, la energía transmitida o reflejada se enfoca a un foto convertidor (p. Ej. Una pantalla de fósforo) que convierte la energía a luz visible [2].

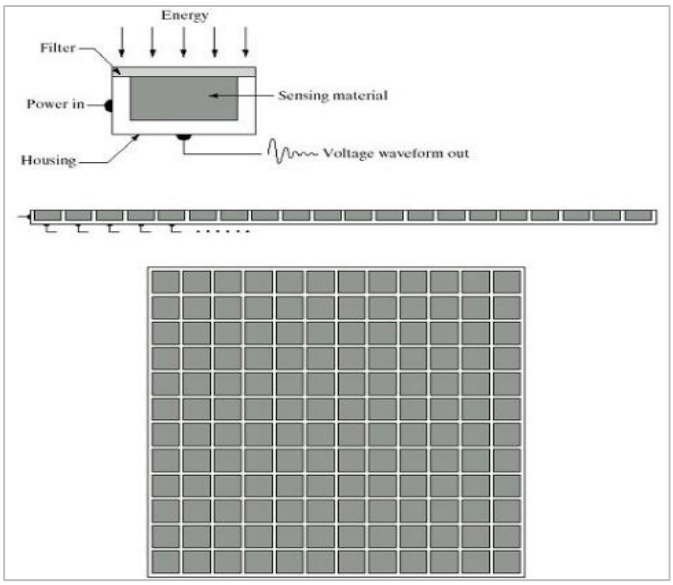


Figura 36: Sensores: Sencillo, en Línea y en Arreglo.

El funcionamiento de cada uno de estos sensores es simple: La energía entrante se transforma a un voltaje por la combinación de electricidad de entrada y el material del sensor, sensible al tipo de energía que se quiere detectar. La onda de voltaje de salida es la respuesta del sensor, y una cantidad digital se obtiene de cada sensor digitalizando su respuesta.

1.2.1. Adquisición a través de un solo sensor

El sensor más familiar de este tipo es el fotodiodo, de silicón, cuya voltaje de salida es proporcional al de la luz. A veces se utiliza un filtro frente al sensor para mejorar la selección de ciertas longitudes de onda (p. Ej. Un filtro verde favorece la banda verde del espectro) [2].

Para generar una imagen 2D, se requieren desplazamientos relativos en las direcciones **x** e **y** del área a capturar. En la figura vemos un sensor montado en un tornillo que le da movimiento en direcció perpendicular. Este método es una manera barata pero lenta de obtener imágenes de alta resolución [2].

Arreglos similares usan una “cama” plana (flat bed), con el sensor moviéndose en 2 direcciones lineales.

A este tipo de sensores se les llama microdensitómetros.

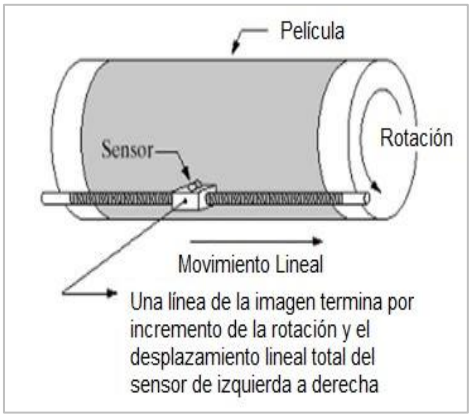


Figura 37: Obtención de una imagen 2D.

1.2.2. Adquisición por medio de bandas o líneas de sensores

Consiste en un arreglo de sensores en línea, formando una banda que provee elementos de la imagen en una dirección. La dirección perpendicular se obtiene por el movimiento de la banda. Este es el tipo de arreglo utilizado en la mayoría de los escáneres de “cama” [2].

Se utilizan rutinariamente en aplicaciones de imágenes aéreas, en las que el movimiento perpendicular es dado por el movimiento del avión. Se utilizan lentes u otro tipo de esquemas de enfoque para proyectar el área a escanear en los sensores [2].

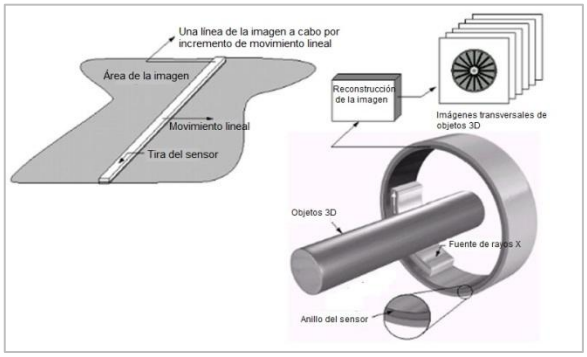


Figura 38: Sensores en línea y circular.

1.2.3. Adquisición a través de arreglos de sensores

Numerosos aparatos de sensado electromagnéticos y algunos ultrasónicos recuentemente se ordenan en forma de arreglos.
Este es también el tipo de ordenamiento de sensores que encontramos en las cámaras digitales. Un sensor típico de estas cámaras es el arreglo CCD (Dispositivos de Acoplamiento de Carga).

La respuesta de cada sensor es proporcional a la integral de la energía luminosa proyectada en la superficie del sensor. Esta propiedad se utiliza en aplicaciones astronómicas que requieren imágenes con bajo nivel de ruido. La reducción del ruido se obtiene dejando al sensor “integrar” señales luminosas por minutos y hasta horas.

La ventaja de los arreglos es que una imagen completa puede obtenerse con sólo enfocar el patrón de energía en la superficie del arreglo (no requiere movimiento) [2].

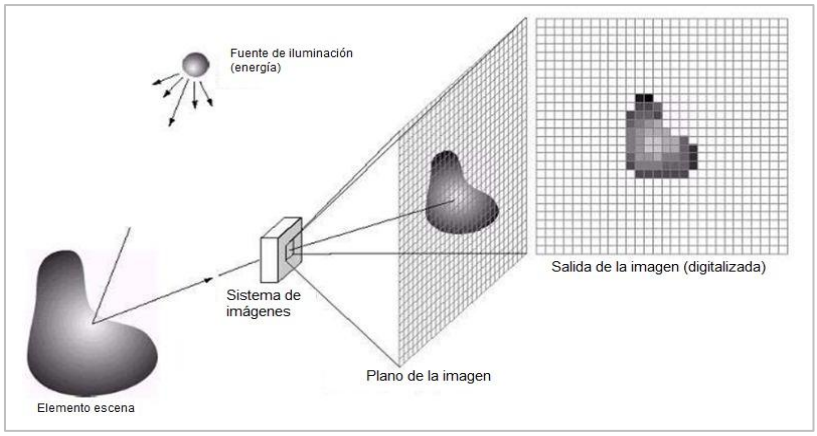


Figura 39: Adquisición de imágenes por medio de arreglos de sensores.

1.3. Procesamiento de imágenes

El procesamiento de imágenes tiene como objetivo, mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar. La imagen puede haber sido generada de distintas maneras, por ejemplo, fotográficamente o electrónicamente. El procesamiento de las imágenes es posible mediante métodos ópticos o digitales (en una computadora) [19].

En Matlab una imagen a escala de grises es representada por medio de una matriz bidimensional de **m** x **n** elementos en donde **n** representa el número de píxeles de ancho y **m** el número de píxeles de largo. Cada elemento de la matriz de la imagen tiene un valor de 0 (negro) a 255 (blanco).

Una imagen de color RGB (la más usada para la visión computacional, además de ser para Matlab la opción default) es representada por una matriz tridimensional **m** x **n** x **p**, donde **m** y **n** tienen la misma significación que para el caso de las imágenes de escala de grises mientras **p** representa el plano, que para RGB que puede ser 1 para el rojo, 2 para el verde y 3 para el azul.

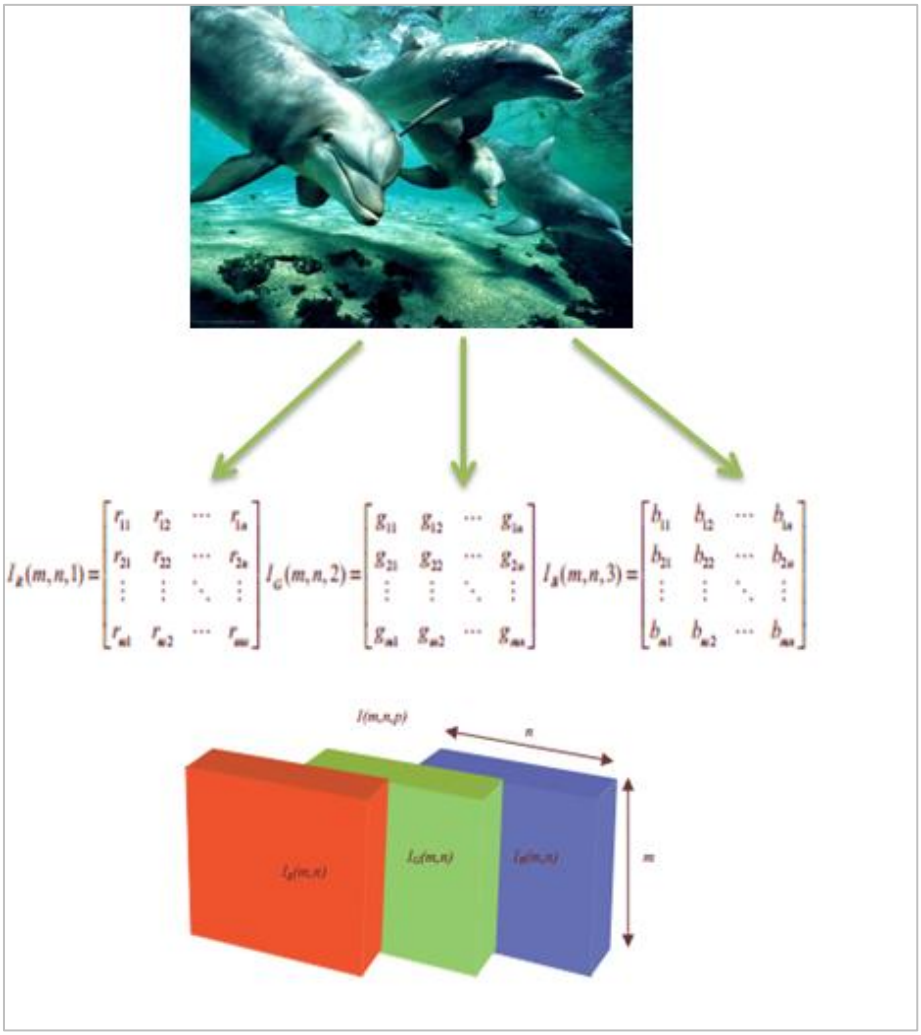


Figura 41: Representación de imágenes a color RGB.

1.3.1. Procesamiento óptico de imágenes

Este tipo de procesamiento de imágenes surge cuando se desarrolló la teoría de la difracción de la luz y que empezó a ponerse en práctica por la década de los sesenta, cuando nació el rayo láser.

En sí, el procesamiento óptico de imágenes se basa en el hecho de que la difracción establecida por Fraunhofer de una transparencia colocada en el plano focal frontal de una lente es una distribución luminosa que representa la distribución de las frecuencias de Fourier que forman parte de la imagen, conocida como transformada de Fourier.

1.3.2. Utilidades del procesamiento de imágenes

La utilidad de realizar el procesamiento de imágenes radica en muchos campos, tal es el caso de las imágenes que son obtenidas con fines de diagnóstico médico, otras para realizar exámenes de terrenos.

Además se pueden realizar algunas operaciones con las imágenes, como por ejemplo:

- Las operaciones de punto
- Las operaciones de entorno
- Las operaciones con dos o más imágenes

1.3.3. Reconocimiento de imágenes

Hoy en día existen en el mercado Sistemas que son cada vez más robustos y económicos que permiten al hombre crear y expresar nuevas formas de arte, cosa que no era común en épocas remotas, incluso algunas formas de arte pertenecientes a la época antigua puede resultar beneficiado por medio de novedosas técnicas asistidas por computador.

El proceso de reconocimiento de imágenes avanza al igual como avanza la tecnología y que está presente en diferentes campos [19].

1.3.4. Reconocimiento de caracteres

La tecnología OCR (Reconocimiento Óptico de Caracteres) proporciona a los sistemas de reproducción por escáner y sistemas de imágenes la habilidad de convertir imágenes de caracteres en letra de máquina, en caracteres capaces de ser interpretados o reconocidos por una computadora. Así, las imágenes de caracteres en letra de máquina son extraídas de un mapa de bits de la imagen reproducida por el escáner.

El proceso OCR envuelve varios aspectos como segmentación, extracción de características y clasificación [20].

$$r = \frac{\sum_m \sum_n \left(A_{mn} - \overline{A}\right)\left(B_{mn} - \overline{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \overline{A}\right)^2\right)\left(\sum_m \sum_n \left(B_{mn} - \overline{B}\right)^2\right)}}$$

Figura 42: Ecuación para realizar la clasificación, basada en la correlación 2D.

1.4. Filtrado de imágenes

Se utiliza para la modificación de imágenes, sea esta para detectar los bordes de una escena o modificar el aspecto, o también para eliminar el ruido (efecto de luminosidad en las imágenes digitales) de las imágenes.

Al utilizar los filtros estamos realizando una convolución de una matriz respecto de un pixel y la vecindad de este. Por ejemplo si tenemos una imagen de 200 x 300 píxeles y el filtro está representado por una matriz de 3 x 3 que se irá desplazando pixel a pixel, desde la posición (1,1) hasta llegar a la posición (199,299), para lo cual se utiliza la siguiente ecuación $|\overline{\boldsymbol{G}| = \sqrt{\boldsymbol{Gx}\text{ˆ}\boldsymbol{2} + \boldsymbol{Gy}\text{ˆ}\boldsymbol{2}}}$ [19].

Para la detección y extracción de bordes se puede utilizar el gradiente de Sobel. Para calcular la derivada de la ecuación podemos utilizar las diferencias de primer orden entre dos píxeles adyacentes.

$$G_x = \frac{f(x+\nabla x) - f(x-\nabla x)}{2\nabla x}$$

$$G_y = \frac{f(y+\nabla y) - f(y-\nabla y)}{2\nabla y}$$

$$g(x,y) = \begin{cases} 1 & si \quad G[f(x,y)] > T \\ 0 & si \quad G[f(x,y)] \leq T \end{cases}$$

Figura 43: Cálculo de la derivada de la ecuación.

En la Figura 42, **T** representa un umbral no negativo entre 0 y 255 para las imágenes en niveles de gris. Los píxeles que resultan importantes son aquellos que exceden el valor de **T.**

Las derivadas basadas en los operadores de Sobel son:

$$G_x = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$
$$G_y = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

Figura 44: Operadores de Sobel.

Donde **Z** representa los niveles de gris de los píxeles solapados por las máscaras en cualquier localización de la imagen, y **Gx** y **Gy** son los gradientes para las respectivas máscaras, los cuales nos permitirán decidir si un determinado punto es de borde o no para luego obtener una imagen binaria [2].

1.5. Procesamiento digital de imágenes

Hasta las décadas de los años 50's y 60's comenzó el desarrollo de técnicas de procesamiento de imágenes de manera formal, debido a que fueron necesarias para la transmisión y procesamiento de imágenes desde los satélites. Uno de los precursores fue el Jet Propulsión Laboratory de Estados Unidos que desarrolló técnicas de transmisión, realce y restauración de imágenes obtenidas por los satélites en el espacio [21].

Durante la década de los setenta se financiaron proyectos de investigación para el desarrollo de técnicas que pudieran estudiar los mecanismos de la visión y que pudieran analizar imágenes digitales. Se comenzaron a aplicar estas técnicas en la investigación de los rayos X, microscopía óptica y electrónica. También el MIT en conjunto con la Universidad de Stanford enfocó sus trabajos en la visión aplicada en la Robótica con el proyecto Hand-Eye. Además hubo un proyecto japonés denominado PIPS (Patter Information Processing System) que desarrollaron este tipo técnicas, pero con fines militares [21].

Con el pasar de los años estas técnicas de procesamiento de imágenes se fueron perfeccionando cada vez más y más, dando lugar a los algoritmos que hoy en día nos permiten resolver problemas de interpretación de imágenes, con la ayuda de la Inteligencia Artificial.

1.6. Técnicas del procesamiento digital de imágenes

La utilización de las técnicas de procesamiento digital de imágenes básicamente se centra en dos aspectos: mejoramiento de la información de la imagen para su interpretación y procesamiento y para su reconocimiento por medio de Sistemas computarizados.

1.6.1. Bancarización y Umbralización de imágenes

La binarización de una imagen es una conversión que se realiza de un nivel de grises a una imagen de blanco y negro. También se la suele entender como una comparación de los niveles de grises que se hallan presentes en una imagen con un valor (umbral) predeterminado. Y es aquí donde se decide si el nivel de gris de la imagen es menor al umbral establecido (predeterminado), se le asigna al pixel de la imagen binarizada el valor de 0 (color negro), de lo contrario el valor de 1 (color blanco) [21].

a) b)

Figura 45: a) Imagen original, b) Imagen binaria. [23]

Este proceso (técnica) es muy útil al momento que se desea separar objetos de interés desde el fondo de una imagen y realizar algunas operaciones como son: el reconocimiento de objetos o la clasificación de objetos que nos interesan.

Cabe recalcar que para algunas aplicaciones que requieren otros niveles de gris, esta técnica no nos será útil, para lo cual necesitamos obtener un valor de pixel adecuado, conocido como valor umbral y cuya operación de selección se denomina umbralización (Tresholding) [21].

$$B(x,y) = \begin{cases} 0 & I(x,y) \leq Th \\ 1 & I(x,y) > Th \end{cases}$$

Figura 46: Operación básica de binarización

Donde **B(x,y)** representa la imagen binaria, **I(x,y**) es un pixel que se halla en la posición **(x,y)** de la imagen de nivel gris y **Th** es el valor del umbral.

Para obtener una buena imagen binaria, podemos utilizar la **Umbralización global sencilla,** donde el valor del umbral (Th) se lo obtiene utilizando la información del histograma de la imagen, pero con la limitación de que la iluminación sea constante.

También podemos utilizar la **Umbralización óptima,** la cual se basa en la función de densidad de probabilidad de iluminación **p (z).** Debemos considerar que si la imagen a analizar consiste en dos grupos de iluminación: la parte oscura que pertenece al fondo de la imagen y la parte clara que pertenece al objeto, para este caso utilizaremos la siguiente ecuación: $\boldsymbol{p(z) = P1p1(z) + P2p2(z)}$.

Para obtener un valor umbral óptimo es necesario realizar una división de la imagen en sub imágenes de tamaño **bxb**, y solo en lagunas sub imágenes que tenga un histograma bimodal, se calcula el valor del umbral utilizando el método de **umbralización global sencilla.** [21]

$$B(x,y) = \begin{cases} 1, & Si \quad I(x,y) \geq Txy \\ 0, & en\ otro\ caso \end{cases}$$

Figura 47: Proceso de binarización.

1.7. Aplicaciones de la identificación de imágenes

Los Sistemas de computadoras son cada vez más potentes y menos costosos, lo que permite crear nuevas formas de arte que antes no era posible, y algunas otras formas de arte antiguas pueden ahora verse beneficiadas con novedosas técnicas asistidas por computadora.
El reconocimiento de imágenes ha evolucionado a medida que mejora la tecnología. Puede encontrarse en numerosos campos.

1.7.1. Identificación de personas para investigaciones policiacas

Aunque las técnicas aún están en desarrollo en este campo, y aún no existe una aplicación totalmente confiable, es evidente la importancia del reconocimiento de imágenes para la identificación de personas en investigaciones policiacas.

Muchas veces en investigaciones de crímenes un testigo puede describir con mucho detalle el rostro de un criminal. Un dibujante profesional convierte la descripción verbal del testigo en un dibujo sobre papel. El trabajo de la computadora consiste en buscar el rostro del criminal en una base de datos de imágenes. En las investigaciones policíacas también se utiliza la búsqueda de huellas dactilares en una base de datos [22].

Figura 48: Identificación biométrica para detectar delincuentes.

1.7.2. Biometría

La biometría es el reconocimiento del cuerpo humano a través de ciertas características físicas, como el tamaño de los dedos de la mano, las huellas dactilares o los patrones en las retinas de los ojos.

Los sistemas de computadoras actuales permiten tener mejores niveles de seguridad utilizando la biometría. Por ejemplo, una persona puede tener acceso a un área restringida, por medio del reconocimiento de las características físicas de su mano en un dispositivo especial. Si en el proceso de validación se verifica que la persona tiene permiso para entrar al área, entonces le permitirá el acceso. Este tipo de sistemas se está volviendo cada vez más utilizado, desplazando los sistemas antiguos de identificación [22].

1.7.3. Información inteligente sobre tráfico vehicular

Esta aplicación nos permite obtener información del flujo vehicular en las calles como datos estadísticos, para de esta manera poder mejorar los parámetros de circulación, tales como: tiempos de espera de semáforos, determinación de tiempos y lugares críticos para operativos de agilización [22].

Actualmente se están desarrollando los denominados semáforos inteligentes que permiten tomar decisiones dependiendo de una serie de parámetros de entrada, como el flujo de vehículos, velocidad media, entre otros, para tratar de evitar el congestionamiento vehicular, tiempos excesivos de viajes, esperas innecesarias, mayor contaminación en el medio ambiente.

En la **Figura 48** un ejemplo de semáforo inteligente, donde se evalúa el tráfico vehicular.

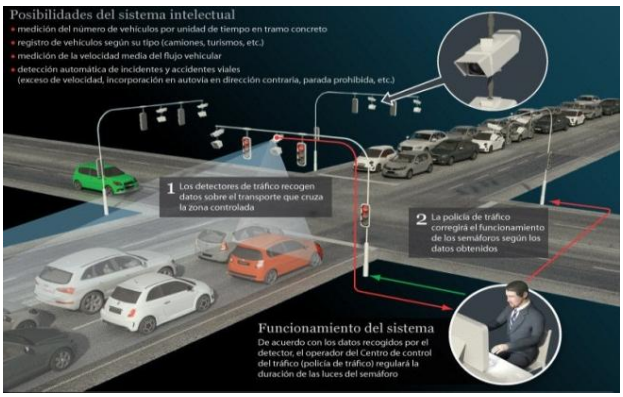


Figura 49: Ejemplo de semáforos inteligentes.

1.8. Sistemas de reconocimiento de placas (matrículas) vehiculares existentes

Los sistemas de Reconocimiento Automático de Matrículas o en inglés Automatic Number Plate Recognition (ANPR) son aplicaciones de Visión por Computador (Visión Artificial) que están compuestos por software y hardware adecuados que permiten la lectura de las placas de los vehículos, su proceso básico radica en que una cámara captura la imagen de la placa y luego con la ayuda de algunas técnicas de procesamiento digital de imágenes se procede a la binarización, segmentación y adelgazamiento de la misma, para finalmente ser reconocida por medio de un algoritmo de Reconocimiento Óptico de Caracteres (OCR) [23].

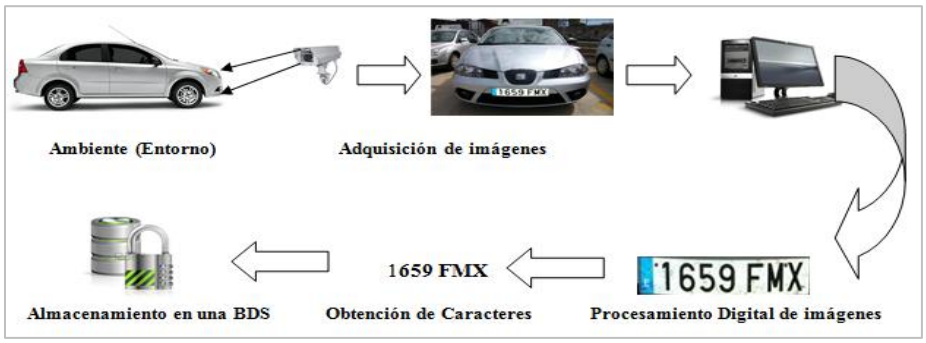


Figura 50: Esquema de un Sistema ANPR.

El ANPR se puede utilizar para almacenar las imágenes capturadas por las cámaras fotográficas, así como el texto de la matrícula, y algunas se pueden configurar para almacenar una fotografía del conductor. Estos sistemas a menudo utilizan iluminación infrarroja para hacer posible que la cámara pueda tomar fotografías en cualquier momento del día. En al menos una versión de cámara fotográfica para la supervisión de intersecciones se incluye un flash de gran alcance, que sirve para iluminar la escena y hacer que el infractor se dé cuenta de su error. La tecnología ANPR tiende a ser específica para una región, debido a la variación entre matrículas de un lugar a otro.

El software del sistema se ejecuta sobre un hardware de PC estándar y puede ser enlazado con otras aplicaciones o bases de datos. Primero utiliza una serie de técnicas de manipulación de la imagen para detectar, normalizar y realzar la imagen del número de la matrícula, y finalmente reconocimiento óptico de caracteres para extraer los alfanuméricos de la matrícula [23].

A los Sistemas de Reconocimiento Automático de Matrículas, también se los conoce con otros términos:

- Identificación Automática de Vehículos (Automatic Vehicle Identification, AVI).
- Reconocimiento de Matrículas de Vehículos (Car Plate Recognition, CPR).
- Reconocimiento de Matrículas (Licence Plate recognition, LPR).

Como tecnología, lo Sistemas ANPR utilizan el reconocimiento óptico de caracteres (OCR) en las imágenes tomadas por las cámaras fotográficas. Algunas matrículas utilizan cambios en los tamaños de las fuentes y en la posición - los sistemas ANPR deben poder hacer frente a estas diferencias para ser verdaderamente eficaces. Algunos sistemas más complicados pueden distinguir variantes internacionales, aunque muchos programas se adaptan a cada país individualmente [23].

1.8.1. Algoritmos comunes utilizados para el reconocimiento de matrículas vehiculares

Existen algunos algoritmos principales que el software necesita para identificar una matrícula:

- Localización de la matrícula que permite encontrar y aislar la matrícula en la imagen.
- Orientación y tamaño de la matrícula, compensa los ángulos que hacen que la matrícula parezca "torcida" y ajusta las dimensiones al tamaño requerido.
- Normalización, ajusta el brillo y el contraste de la imagen.
- Segmentación de los caracteres, encuentra los distintos caracteres presentes en la matrícula.
- Reconocimiento óptico de caracteres.
- Análisis sintáctico y geométrico, comprueba los caracteres encontrados y sus posiciones con las reglas específicas del país al que pertenece la matrícula.
- La complejidad de cada una de estas subdivisiones del programa determina la exactitud del sistema. Durante la tercera fase (normalización) algunos sistemas utilizan técnicas de detección de borde para aumentar la diferencia en la imagen entre las letras y el fondo de la placa. A También se puede utilizar un filtro digital de punto medio para reducir el "ruido" visual de la imagen.

1.8.2. Problemas que impiden el correcto funcionamiento de los Sistemas ANPR

El software debe ser capaz de afrontar diferentes dificultades posibles, que incluyen:

- Resolución de imagen pobre, a menudo porque la matrícula está demasiado lejos, aunque a menudo es resultado del uso de una cámara de baja calidad.
- Imágenes desenfocadas, en particular desenfoque de movimiento y muy a menudo en unidades móviles
- Iluminación pobre y bajo contraste debido a sobreexposición, reflexión o sombras
- Un objeto que oscurece (parte de) la matrícula, a menudo una barra del remolque, o suciedad en la matrícula
- Técnicas de evasión

Figura 51: Los primeros sistemas ANPR eran incapaces de leer letras blancas o plateadas sobre un fondo negro.

Aunque algunos de estos problemas se pueden corregir en el software, se dejan sobre todo en el lado del hardware del sistema para ofrecer soluciones a estos problemas. El aumento de la altura de la cámara puede evitar problemas con los objetos (tales como otros vehículos) que oscurecen la placa, pero introduce y aumenta otros problemas como el ajuste según la oblicuidad creciente de la placa.

Muchos países utilizan matrículas retroreflectivas. Esto devuelve la luz hacia la fuente y mejora así el contraste de la imagen. En algunos países los caracteres de la matrícula no son reflectantes, dando un alto nivel del contraste con el fondo reflectante bajo cualquier condición de iluminación. Una cámara que utiliza imagen infrarroja (con un filtro normal de color sobre la lente y una fuente luminosa infrarroja al lado de ella) beneficia en gran medida, reflejándose las ondas infrarrojas desde la matrícula.

Sin embargo, esto sólo es posible en cámaras ANPR dedicadas, por lo que las cámaras usadas para otros propósitos deben confiar en mayor medida en las capacidades del software. Además, cuando se necesita una imagen a todo color y la captación de detalles es necesario tener una cámara con infrarrojos y una cámara normal (en color) funcionando conjuntamente [19].

Figura 52: Imágenes borrosas dificultan el Reconocimiento Óptico de Caracteres.

1.8.3. Usos de los Sistemas ANPR

Los Sistemas ANPR pueden ser utilizados para:

- La gestión de aparcamientos: usando la matrícula a modo de llave para tener acceso al estacionamiento.
- Control de fraude en autopistas: para determinar si un vehículo fue plagiado o no es autorizada su circulación.
- Control de velocidad media en autopistas.
- Control de camiones: situando un lector de matrículas junto a la báscula que mide la carga del camión.
- Inventariado de vehículos: para poder determinar el estado del mismo en el momento de que ingresa al aparcamiento, en caso de sufrir algún siniestro.

1.8.4. OCR´s en el mercado

Existen muchos software libres para el reconocimiento de patrones, entre ellos tenemos: Cuneiformes, el OCRopus, Tesseract, Ocrad, GOCR; también los hay de propietarios como los son: ExperVision, FineReader, Microsoft Office Document Imaging, OmniPage, Readiris, ReadSoft, SimpleOCR, SmartScore. A continuación se detallan las principales técnicas libres para el reconocimiento de caracteres:

- **GOCR:** Desarrollada por Joerg Schulenburg, se basa en un conjunto de reglas, es portable a diferentes sistemas operativos, el tamaño de las fuentes que soporta esta técnica es entre 20 - 60 píxeles y acepta muchos formatos de imágenes, como pnm, PBM, pgm, ppm, pcx, tga. Sin embargo, tiene inconvenientes con letras cursivas, texto escrito a mano, adicionalmente es muy sensible a imágenes que contenga ruido y grandes ángulos de inclinación, por lo que se necesita de un gráfico de alta calidad para conseguir buenos resultados.

- **OCRAD:** Creado por Antonio Díaz, es un método de extracción de características geométrico. Es muy rápido, sensible al ruido y difícil de adaptar a nuevos símbolos.

- **TESSERACT:** Desarrollado originalmente por Hewlett Packard, luego liberado por Google, considerado uno de los software's libres de OCR más preciso, incluso mucho más potente que algunos software's propietarios, es multiplataforma y tiene un tiempo de ejecución aceptable, sin embargo presentan problemas para la depuración en el caso de que haya un fallo en la segmentación. El formato único que procesa es un TIFF y; se compila y ejecuta en Linux, Windows y Mac OS X [23].

Capítulo 3: Librerías utilizadas en el desarrollo de aplicaciones de Visión Artificial

Al momento de desarrollar aplicaciones que estén basadas en Visión Artificial es muy importante saber elegir una librería como herramienta de desarrollo, aunque no sería muy recomendable utilizar solo una. Es probable que las deficiencias de una puedan ser resueltas por otra, o que para un problema concreto resulte aconsejable usar una librería específica como es el caso de Opencv.
A continuación se detallan algunas de las librerías más utilizadas en el desarrollo de aplicaciones que estén enfocadas en la Visión Artificial.

1.1. Torch3vision

Esta librería está escrita en el Lenguaje de Programación C++ y dispone de un procesamiento básico de imágenes, algoritmos de extracción de características, así como detección de caras empleando Haar-like features. Es libre y posee una licencia BSD.

En su página podemos encontrar documentación, tutoriales y algunos ejemplos. El problema es que parece que su desarrollo se encuentra parado, la última versión, la 2.1, data del 2007, toda una eternidad [24].

1.2. VLX

Esta librería también está escrita en el Lenguaje de Programación C++, incorpora la mayoría de los algoritmos habituales en visión artificial. En realidad no es una única librería, sino más bien un conjunto de ellas que ofrecen una muy completa funcionalidad. Es una opción a tener en cuenta ya que dispone de unas características muy atractivas, una de ellas es la posibilidad de usar únicamente las librerías que nos resulten de utilidad ya que no hay dependencias entre ellas [24].

1.3. RAVL

Está escrita en C++ y nos proporciona los elementos básicos de una librería de visión artificial. Además incorpora algunos elementos diferenciadores tales como soporte para herramientas de audio o interfaces de usuario basadas en GTK. Promete ser de fácil programación y de no requerir grandes conocimientos en C++ [24].

1.4. LTI-lib

Dispone de más de 500 clases, entre las que se incluyen clases para álgebra lineal, clasificación, procesamiento de imágenes y muchas otras características. Ha sido testada bajo Linux gcc y en Windows NT VisualC++. Su licencia es GNU Lesser General Public License [24].

1.5. Opencv

Esta es una de las librerías más popular que existe, posee alrededor de 500 algoritmos entre los que se incluye funciones de propósito general para procesamiento de imágenes, descripciones geométricas, segmentación, seguimiento de objetos, etc.

Además permite el uso de las librerías de Intel (*Integrated Performance Primitives, IPP)* que incluyen una larga lista de funciones optimizadas para procesadores Intel. Si las librerías se encuentran instaladas *OpenCV* hace uso de las mismas, mejorando la velocidad en los cálculos.

Debido a que es muy conocida esta librería dentro de la Visión Artificial, posee muchísima documentación, incluyendo algunos libros. Está disponible para Linux, Windows y Android. Se puede programar en C++, C, Python y Java [24].

Cabe recalcar que las librerías citadas anteriormente no son las únicas, pero si las más comunes debido a su potencia en el desarrollo de aplicaciones que estén inmersas dentro de la Visión Artificial.

1.6. Algoritmos específicos utilizados para el reconocimiento y lectura de placas vehiculares

En este apartado nos introduciremos en conocer algunos algoritmos necesarios para crear una aplicación de reconocimiento de placas vehiculares. Hay diferentes enfoques y técnicas basados en diferentes situaciones, por ejemplo, cámaras de infrarrojos, las posiciones fijas del coche, las condiciones de luz, y así sucesivamente. Podemos proceder a construir una aplicación ANPR para detectar matrículas de automóviles en tiempo real, en condición de luz ambigua, y con el suelo no paralelo con la perspectiva de menor importancia y con distorsiones de la placa del automóvil.

No obstante el objetivo principal de este apartado también es para introducirnos en la segmentación de imágenes y extracción de características, fundamentos de reconocimiento de patrones, y dos algoritmos importantes utilizados para el reconocimiento de placas vehiculares, así por ejemplo está el algoritmo para realizar la detección de las placas y el OCR (Reconocimiento Óptico de Caracteres) que se utilizará para realizar el reconocimiento o lectura de las placas de los vehículos, comúnmente conocido como reconocimiento de patrones.

1.6.1. Máquinas de Vectores de Soporte (SVM)

Este algoritmo se basa en las conocidas Máquinas de Vectores de Soporte (SVM) que utilizan un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

La representación de los datos a clasificar se realiza en el plano x-y. El algoritmo SVM trata de encontrar un hiperplano 1-dimensional que une a las variables predictoras y constituye el límite que define si un elemento de entrada pertenece a una categoría o a la otra.

De manera concreta se denominan vectores de soporte a los puntos que conforman las dos líneas paralelas al hiperplano, siendo la distancia entre ellas (margen) la mayor posible. En la siguiente figura se puede apreciar de una mejor manera todo lo anteriormente dicho:

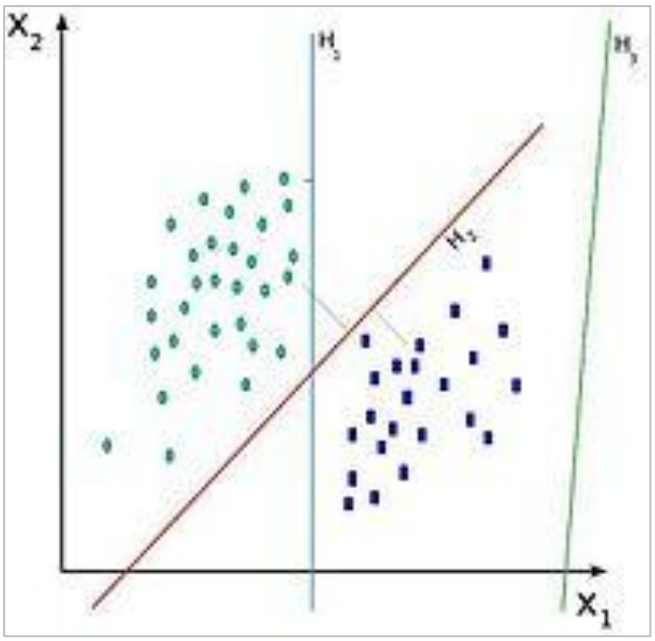


Figura 53: Clasificación que realizan los algoritmos SVM.

De manera oportuna, el modelo basado en SVM debería producir un hiperplano que separe completamente los datos del universo estudiado en dos categorías. Sin embargo, una separación perfecta no siempre es posible y, si lo es, el resultado del modelo no puede ser generalizado para otros datos. Esto se conoce como sobreajuste (overfitting).

Con el fin de permitir cierta flexibilidad, los algoritmos SVM manejan un parámetro C que controla la compensación entre errores de entrenamiento y los márgenes rígidos, creando así un margen blando (soft margin) que permita algunos errores en la clasificación a la vez que los penaliza.

Los algoritmos SVM son muy similares a las Redes Neuronales Artificiales (ANN) debido a que utilizan valores numéricos para realizar las clasificaciones respectivas, representando así matrices de valores numéricos.

Resulta muy interesante realizar una comparativa entre estas técnicas utilizadas dentro de la Visión Artificial, y como no decirlo dentro de la Inteligencia Artificial que es de donde surge la Visión Artificial.

Tabla 1: COMPARATIVA ENTRE ANN Y SVM

Comparación entre ANN y SVM	
ANN	**SVM**
• Las capas ocultas transforman a espacios de cualquier dimensión • El espacio de búsqueda tiene múltiples mínimos locales • El entrenamiento es muy costoso • La clasificación es muy eficiente • Se debe diseñar el número de capas y los nodos • Muy buenos resultados en problemas típicos.	• Kernels transforman a espacios de dimensión muy superior • El espacio de búsqueda sólo tiene un mínimo global • El entrenamiento es muy eficiente • La clasificación es muy eficiente • Se debe diseñar la función kernel y el parámetro de coste C • Muy buenos resultados en problemas típicos • Extremadamente robusto para generalización, menos necesidad de heurísticos para archivos de entrenamiento.

Las placas en Ecuador tienen como medidas 154 mm de alto y 404 mm de ancho y son reflectivas, esto con el fin de mejorar su visibilidad. Las placas de los vehículos en Ecuador están formadas por tres letras y tres o cuatro dígitos, partiendo desde 000 al 9999, en formatos de ABC-123 y ABC-1234, con el nombre del país en mayúsculas en la parte superior. Las placas son emitidas por el ente regulador de Tránsito en el Ecuador la ANT. La primera letra indica la provincia en la que se matriculó por primera vez el vehículo, la segunda identifica el tipo de la matrícula y la tercera es correlativa. Ejemplo: ABC-123 y ABC-1234 [25].

Dependiendo del tipo de vehículo, las matrículas tienen colores diferentes. Con la modificación del reglamento de tránsito en junio de 2012, se modificó en la forma de las matrículas. Para los vehículos no particulares, las nuevas matrículas conservarán

el mismo color diferenciador pero ya no será aplicado en la totalidad de la matrícula, sino solo en el borde superior, siendo el resto de la matrícula de color blanco. Este cambio se realizó de manera que se mejore la visibilidad de las matrículas, en particular ante las cámaras y radares.

Se espera que en un plazo de 5 años todas las matrículas sean remplazadas por las nuevas, sin embargo, las antiguas matrículas seguirán siendo válidas [25].

Figura 54: Placa de un vehículo con sus medidas correspondientes.

La siguiente tabla contiene los nombres de las Provincias de Ecuador con sus respectivas letras, que serán las que deben ir en las placas de los vehículos, dependiendo de la provincia en la que se realice la matriculación de los vehículos.

Tabla 2: PROVINCIAS DEL ECUADOR Y SUS LETRAS EN LA ASIGNACIÓN DE PLACAS

PROVINCIA	LETRA	PROVINCIA	LETRA	PROVINCIA	LETRA
Azuay	A	**Galápagos**	W	**Pastaza**	S
Bolívar	B	**Guayas**	G	**Pichincha**	P
Cañar	U	**Imbabura**	I	**Orellana**	Q
Carchi	C	**Loja**	L	**Sucumbíos**	K
Cotopaxi	X	**Los Ríos**	R	**Tungurahua**	T
Chimborazo	H	**Manabí**	M	**Zamora Chinchipe**	Z
El Oro	O	**Morona Santiago**	V	**Santa Elena**	Y
Esmeraldas	E	**Napo**	N	**Sto. Domingo de los Tsáchilas**	J

e. Materiales y Métodos

Los materiales y métodos empleados en el desarrollo del presente proyecto de fin de carrera se detallan a continuación:

1. Materiales

Para que el desarrollo del presente Trabajo de Fin de Carrera sea exitoso fue necesario realizar una planificación y control en términos económicos de un conjunto de materiales, los cuales están ligados con las actividades que se desarrollaron en el proyecto.

A continuación se describen los materiales utilizados en el desarrollo del proyecto con sus respectivos costos:

1.1. Talento Humano

El presente proyecto se planificó para ser desarrollado por dos investigadores, contando con la asesoría de un docente de la carrera, para el desarrollo del proyecto se estimó un tiempo de ocho meses en el cual se trabajaron 5 horas diarias en días laborables, es decir, se trabajaron cien horas mensuales.

En la siguiente tabla se detallan el total de horas que se emplearon en el desarrollo del proyecto con sus respectivos costos:

Tabla 3: TALENTO HUMANO.

Equipo de trabajo	Tiempo (Horas)	($) Precio / Hora	($) Valor Total
Investigadores	800	4.00	3200.00
Director del PFC	300	0.00	0.00
	SUBTOTAL		3200.00

1.2. Servicios

Además de contar con el talento humano, también se utilizaron servicios básicos que fueron necesarios para el desarrollo del proyecto. Tal es el caso del uso de Internet y el transporte. El Internet fue utilizado como un medio para realizar investigación sobre el proyecto y el transporte para poder asistir a las tutorías planificadas por el Director del Proyecto de Fin de Carrera.

En la siguiente tabla se detallan estos servicios básicos.

Tabla 4: SERVICIOS.

Servicio	Descripción	($) P. Unitario	($) Total
Internet	8 meses	20.00	160.00
Transporte	160 días	1.00	160.00
	SUBTOTAL		320.00

1.3. Recursos Hardware y Software

Durante el desarrollo del Sistema se utilizaron recursos Hardware y Software. Entre los recursos Hardware están: una computadora, una impresora, un pendrive y una cámara, y entre los recursos Software está el paquete de Microsoft office 2010, las librerías que posee Opencv para Visión Artificial y algunas otras herramientas que nos facilitaron realizar toda la documentación del proyecto.

Tabla 5: HARDWARE Y SOFTWARE.

RECURSOS HARDWARE				
	($)P. Unitario	**T. vida (años)**	**T. uso (mes)**	**($) Depreciación**
PC. HP	800.00	6	48	200.00
Impresora	85.00	4	5	65.00
Pendrive (16GB)	18.00	3	2	12.00
Cámara	199.00	10	10	70.00
RECURSOS SOFTWARE				
	Descripción			**($) Total**
Office 2010	Para la documentación y presentación.			60.00
Opencv	Para el desarrollo del Sistema.			0.00
Látex	Para la documentación y presentación.			0.00
SUBTOTAL				407.00

1.4. Materiales de Oficina

Para el desarrollo del proyecto también se emplearon algunos materiales de oficina, en los que se destacan insumos de papelería, anillados, empastados, etc.

En la siguiente tabla se detallan cada uno de estos materiales:

Tabla 6: MATERIALES DE OFICINA

	Cantidad	**($) P. Unitario**	**($) Total**
Insumos de papelería	----	100.00	100.00
Anillados	7	2.50	17.50
Empastados	3	7.50	22.50
SUBTOTAL			140.00

El presupuesto total empleado para el desarrollo del presente Proyecto de Fin de Carrera se detalla a continuación, asumiendo un presupuesto para los imprevistos, el mismo que corresponde al 10% del total del presupuesto.

Tabla 7: PRESUPUESTO FINAL

	($) Total
Talento Humano	3200.00
Servicios	320.00
Recursos Hardware y Software	407.00
Materiales de Oficina	140.00
SUBTOTAL	4067.00
Imprevistos (10%)	406.70
Total PFC	4473.70

2. Métodos

Para el desarrollo del Sistema fue necesario seguir un esquema metodológico, el cual está basado en la utilización de diferentes métodos y técnicas que facilitan la recolección de datos, y que se utiliza para estructurar, planificar y controlar un proyecto de desarrollo y nos fue de gran utilidad para obtener grandes posibilidades de éxito en el desarrollo del presente Trabajo de Fin de Carrera.

2.1. Métodos utilizados

Los métodos que se utilizaron durante el desarrollo del proyecto son los siguientes:

- **Método Científico:** Este método nos ayudó a obtener los conocimientos necesarios sobre la Visión Artificial, y que más tarde integrarían el marco teórico o revisión literaria, y que además nos permitieron plantear soluciones mediante procedimientos, ayudando de alguna manera a comprender ciertos conceptos que se utilizaron en el desarrollo del proyecto.

- **Método Inductivo:** Este método permitió verificar como iba a repercutir en el Área el manejo de un Sistema para el control y registro de matrículas vehiculares, y a través de esto se pudo establecer conclusiones sobre las futuras mejoras que tendrá la institución.

- **Método Deductivo:** Permite partir de los aspectos generales entre temas como es el desarrollo de un Sistema basado en Visión Artificial, obteniendo resultados relacionados con el reconocimiento de matrículas vehiculares en el AEIRNNR de la UNL.

- **Método Descriptivo:** Este método ayuda a describir paso a paso el desarrollo del Proyecto de Fin de Carrera.

2.2. Metodología de Desarrollo del Sistema

Luego de haber analizado cuidadosamente las diferentes metodologías de desarrollo de software que existen se optó por utilizar el **modelo en cascada**, el mismo que representa un enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Por lo tanto, cualquier error de diseño que sea detectado en la etapa de pruebas necesariamente se debe proceder al rediseño y nueva programación del código fuente afectado, incrementando los costos del desarrollo. El nombre **cascada** se debe a la metáfora de la fuerza de la gravedad.

A continuación se detallan las tareas que se realizaron en cada una de las etapas del modelo en cascada:

a) Fase de análisis de requisitos

En esta fase se analizaron en conjunto con el administrador las necesidades para la construcción del Sistema, para lo cual se tuvieron que aplicar diferentes técnicas de recolección de datos, las cuales se detallan a continuación:

- **Observación directa:** Para la utilización de esta técnica fue necesario asistir a la entrada del AEIRNNR, que es por donde ingresan los vehículos y observar cuidadosamente los procesos de registro y control que realiza el personal de seguridad (guardias) y determinar ciertos problemas que existen en la institución, especialmente en los procesos manuales.

- **Entrevista:** Fue aplicada al personal de seguridad (guardias) del Área, para obtener datos informativos sobre la situación actual de la parte tecnológica de seguridad, con la que cuenta la institución.

b) Fase de Diseño del Sistema

Luego de haber obtenido los requerimientos necesarios para la construcción del Sistema, la siguiente fase consiste en realizar un diseño del mismo, es decir, se debe elaborar un bosquejo del Sistema y se desarrolló un prototipo inicial para mostrárselo al personal de seguridad con el fin de conocer su punto de vista y la acogida que tendría, y si se debería realizar algunos cambios.

c) Fase de Implementación del Sistema

En esta fase es donde se implementó el código fuente, haciendo uso del prototipo diseñado en la etapa anterior. Además se tuvo que realizar una investigación sobre los algoritmos que existen para el desarrollo de este tipo de Sistemas y se procedió a codificar los algoritmos respectivos, pero asumiendo antes todas las configuraciones pertinentes en los entornos de desarrollo, así como la selección del lenguaje de programación teniendo en cuenta la eficiencia y eficacia que ofrezca. Cabe recalcar, que en esta etapa también fue necesario realizar pruebas de los elementos ya programados para comprobar que funcionan correctamente y cumplen con los requerimientos.

d) Fase de Verificación del Sistema

Aquí es donde se realiza la ejecución del Sistema, previo a ello se realizaron exhaustivas pruebas para comprobar el funcionamiento del Sistema.

f. Resultados

Luego de haber realizado la revisión literaria referente a la Visión Artificial y haber establecido los materiales y métodos empleados para el desarrollo del presente Proyecto de Fin de Carrera presentamos los resultados obtenidos en cada una de las fases de la metodología en cascada.

1. Fase 1. Análisis de Requerimientos

En esta fase se realizó un análisis acerca de cómo se llevan los procesos para realizar el control y registro de los vehículos pertenecientes a la institución, lo cual nos permitió realizar la recolección de información, y de esta manera poder determinar los requerimientos del Sistema.

Para que la recolección de información sobre los requerimientos sea exitosa fue necesaria la aplicación de ciertas técnicas de recolección de datos, como son:

- **La observación directa:** que nos permitió conocer de manera presencial cómo se realizan estos procesos de registro y control de los vehículos, para lo cual se tuvo que asistir al AEIRNNR de la UNL.

- **La entrevista:** que fue aplicada al personal de seguridad (guardias) para conocer de manera más detallada como se llevan los procesos de registro y control de vehículos que son parte de la institución y poder obtener información más objetiva.

1.1. Requerimientos Funcionales

El Sistema permitirá:

Tabla 8: REQUERIMIENTOS FUNCIONALES DEL SISTEMA.

REF.	Descripción	Categoría	Técnicas de recolección
REF001	Utilizar una cámara para realizar el Reconocimiento de objetos.	Evidente	Observación
REF002	Realizar la detección de vehículos.	Evidente	Observación
REF003	Realizar la detección de matrículas vehiculares.	Evidente	Observación
REF004	Realizar el reconocimiento óptico de caracteres	Oculto	Entrevista
REF005	Mostrar fotos de las matrículas detectadas.	Evidente	Observación
REF006	Almacenar matrículas reconocidas en una Base de Datos.	Oculto	Entrevista
REF007	Generar reportes de las matrículas almacenadas.	Evidente	Entrevista

1.2. Requerimientos no Funcionales

De la misma forma que los Requerimientos Funcionales, también existen los Requerimientos no Funcionales, en donde se dice que el Sistema deberá:

Tabla 9: REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA.

REF.	Descripción	Categoría	Técnicas de recolección
REF001	Poseer una interfaz que sea amigable para el usuario.	Evidente	Entrevista
REF002	Facilitar su uso de manera apropiada.	Evidente	Entrevista

2. Fase 2. Diseño del Sistema

Luego de haber realizado el análisis y determinación de los requerimientos para el desarrollo del Sistema, el siguiente paso es el diseño del mismo, para lo cual se ha desarrollado un algoritmo que está representado mediante un diagrama de flujo. Además se elaboró el modelo del dominio del sistema, y se muestran algunas capturas de las pantallas cuando se realiza la ejecución del Sistema.

2.1. Algoritmo para la detección de matrículas vehiculares

El algoritmo inicialmente realiza una evaluación sobre si existe o no el objeto a detectar, en este caso sería el vehículo. Seguidamente se realiza algo similar, pero en este caso se realiza la detección de las matrículas o también conocidas como placas vehiculares. Todo lo antes mencionado se consigue con la ayuda de un clasificador (Ver sección **3.1.1.1** y **3.1.1.2**) que se obtiene con la ayuda de varios modelos y formas de vehículos y matrículas vehiculares recolectadas con la ayuda de una cámara digital.

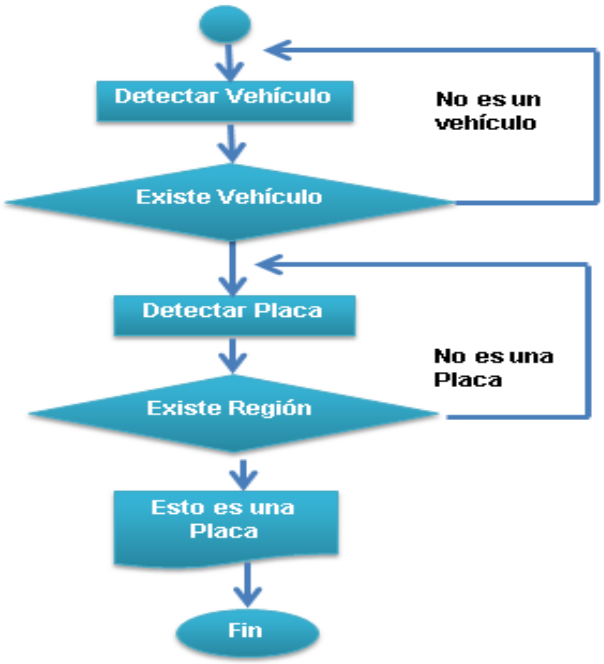


Figura 55: Diagrama de flujo del algoritmo para la detección de matrículas vehiculares.

2.2. Reconocimiento Óptico de caracteres (OCR)

El Reconocimiento Óptico de caracteres (OCR) tiene como objetivo recuperar los datos de la matrícula, es decir, obtener los números y letras de las matrículas. Para cada matrícula detectada, se procede a segmentar cada uno de los números y letras de la misma. En la **Figura 56** se puede apreciar el diagrama de flujo del OCR.

Figura 56: Diagrama de flujo del Reconocimiento Óptico de caracteres de las Placas vehiculares.

2.3. Modelo del dominio del Sistema

Gracias a la ayuda del conjunto de entidades que se emplearon para desarrollar el Sistema se realizó el modelo del dominio, el cual se muestra en la **Figura 57.**

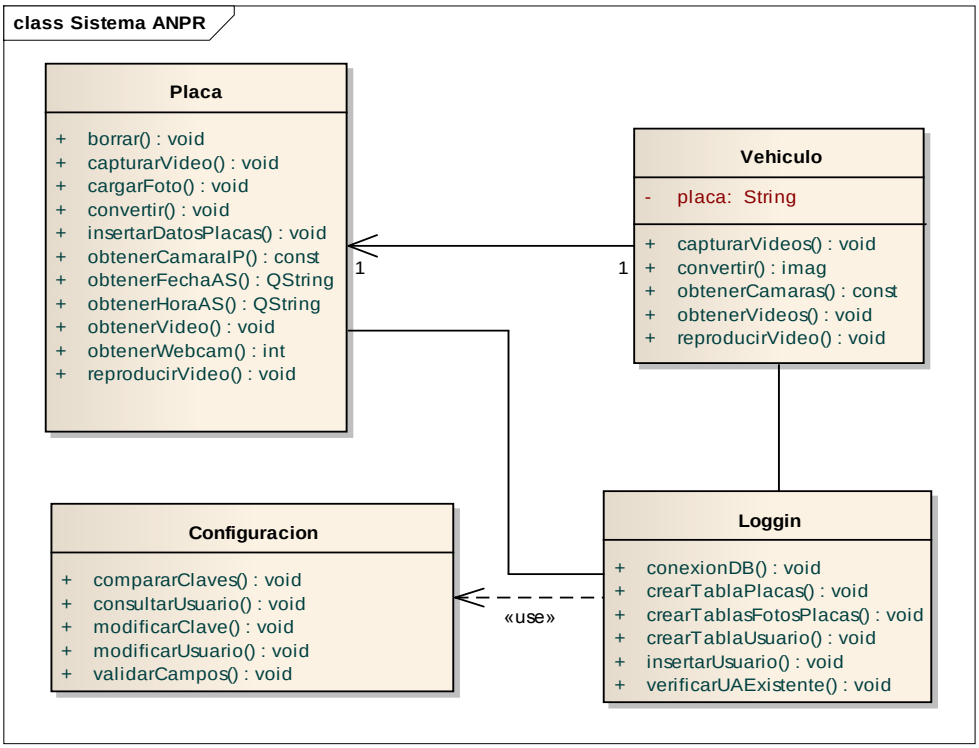


Figura 57: Modelo del dominio del Sistema.

- **Placa:** Clase que se encarga de realizar la detección de placas vehiculares y aplicar el Reconocimiento Óptico de caracteres sobre las mismas.
- **Vehículo:** Clase que permite realizar la detección de vehículos.
- **Configuración:** Esta clase ayuda a realizar la edición de los usuarios, respecto al acceso al Sistema.
- **Loggin:** Clase que se encarga de realizar la conexión del Sistema con la base de datos y permitir acceder al mismo.

2.4. Ventanas principales del Sistema

Luego de que se obtuvieron los requerimientos necesarios para desarrollar el Sistema, se procedió a realizar la construcción de las interfaces gráficas. Por ejemplo para poder acceder al sistema, el usuario tiene que ingresar un **usuario** y una **contraseña,** estos datos son verificados en la base de datos. La **Figura 58** nos muestra el ingreso al sistema.

Figura 58: Pantalla de ingreso al Sistema.

Una vez que se haya ingresado al sistema, el usuario puede escoger alguna de las opciones del menú que se encuentra en la parte superior izquierda de la ventana principal, tal y como se puede ver en la **Figura 59.**

Figura 59: Ventana principal del Sistema.

Si el usuario desea realizar la detección de matrículas vehiculares y uso del reconocimiento Óptico de Caracteres debe seleccionar la opción **Control de Placas,** y podrá hacerlo a través de la siguiente ventana:

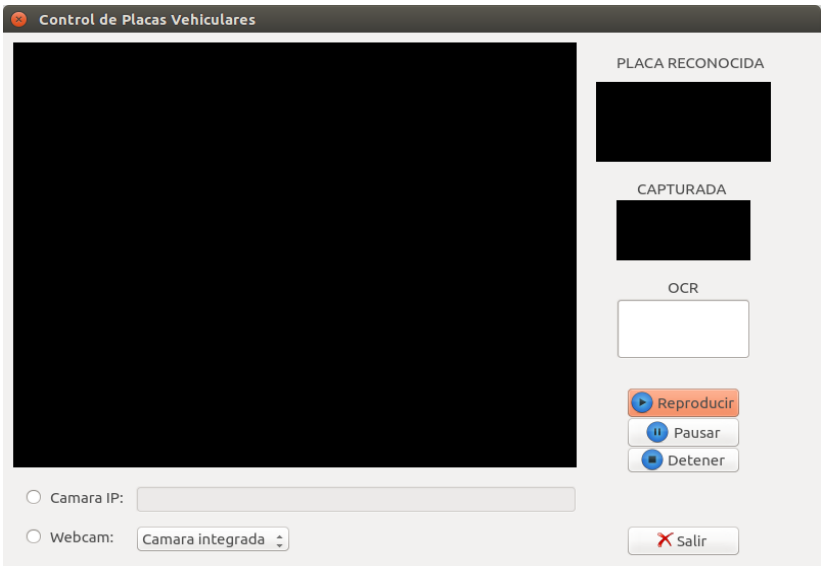


Figura 60: Ventana que permite realizar el Control de Placas Vehiculares.

Y si el usuario desea realizar la detección de vehículos debe seleccionar la opción **Control Vehicular** y podrá hacerlo con la ayuda de la siguiente ventana:

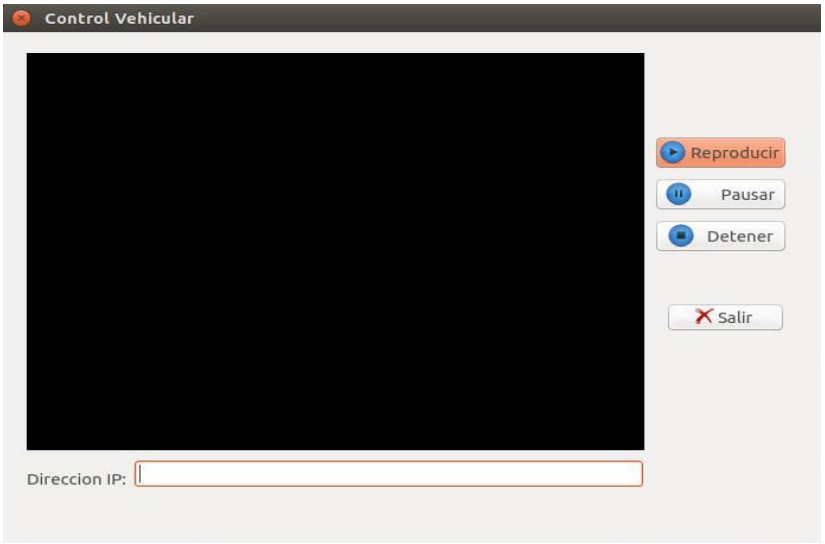


Figura 61: Ventana que permite realizar el Control de Vehículos.

Además existe la posibilidad de que un usuario (administrador) pueda editar los datos de acceso al sistema, tales como datos de **usuario** y **contraseña,** esto es posible al seleccionar la opción **configuración** en la ventana principal del sistema.

Figura 62: Ventana que permite editar datos de acceso al sistema.

3. Fase 3. Implementación del Sistema

En esta fase correspondiente a la metodología en cascada se realizó la codificación del Sistema, para lo cual se utilizó el Lenguaje de programación C++, el IDE QT Creator (Ver Anexo 3) y MYSQL para la Base de Datos.

3.1. Codificación del algoritmo para la detección de placas vehiculares

En esta sección se realizó la codificación del algoritmo para la detección de matrículas vehiculares, para lo cual fue necesaria la construcción de un clasificador en cascada, tanto para la detección de vehículos como para la detección de placas.

3.1.1. Construcción del clasificador en cascada

Como se mencionó anteriormente para el desarrollo del algoritmo correspondiente a la detección de vehículos y matrículas fue necesario construir un clasificador en cascada. Para realizar la detección de vehículos se construyó el clasificador en la plataforma Windows 7, debido a que se emplearon gran cantidad de imágenes (muestras) y para

realizar la detección de matrículas vehiculares se construyó el clasificador en la plataforma Ubuntu 14.04 LTS.

3.1.1.1. Clasificador en cascada para la detección de vehículos

Se realizó la recolección de 2000 imágenes que contengan el objeto a reconocer en nuestro caso vehículos, las cuales las denominamos con el nombre de **positivas** y 5000 imágenes denominadas **negativas**, estas pueden ser, fotos personales, imágenes de playas, paisajes, etc. Para ello se debe tomar en cuenta lo siguiente:

- En las imágenes negativas no debe estar presente el objeto a reconocer.
- Las imágenes positivas y las negativas deben estar numeradas desde el 1 hasta el número de imágenes utilizadas (positivas de 1 a 2000 y negativas 1 a 5000).
- Todas las imágenes en formato mapa de bits (bmp).

Para la construcción del clasificador en el Sistema Operativo Windows se utilizó una herramienta denominada **tools**, la cual posee archivos **.bat** y **.exe** que facilitan la construcción del clasificador. Para obtener esta herramienta visitamos el sitio de descarga [27].

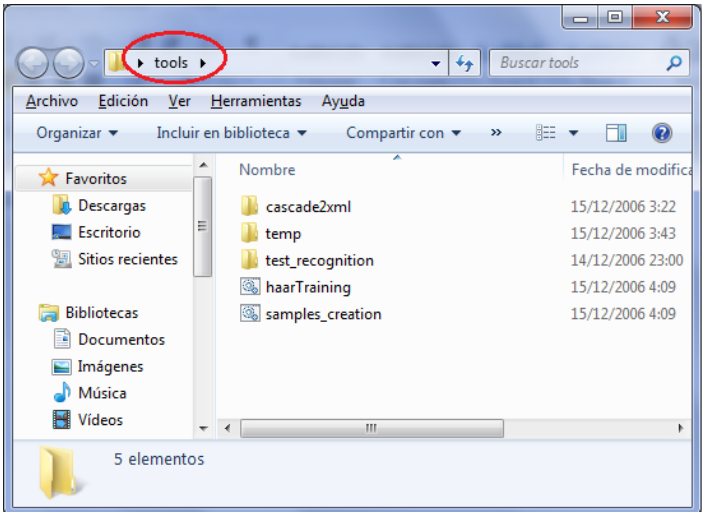


Figura 63: Herramienta tools descomprimida.

Una vez recolectado las imágenes las ubicamos de manera correspondiente en los siguientes directorios **tools\temp\negative** para las negativas y **tools\temp\positive\rawdata** para las positivas.

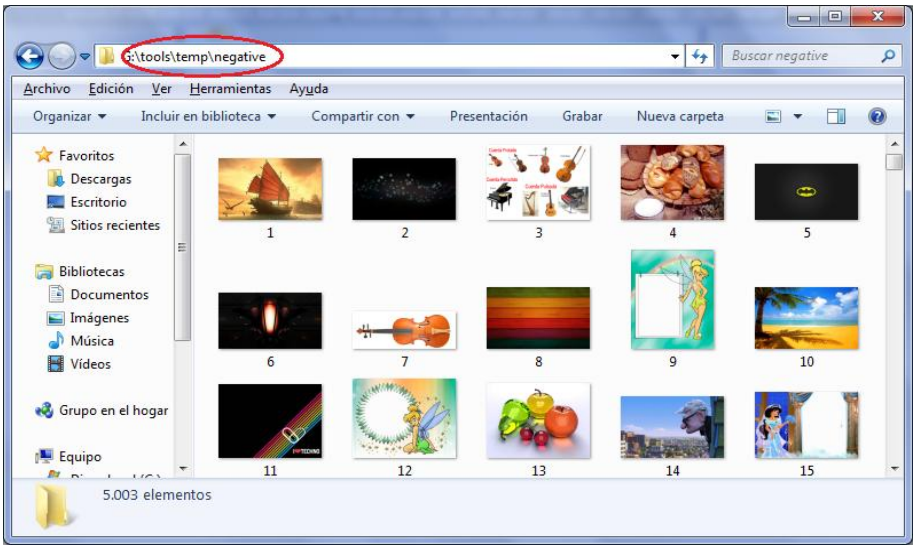


Figura 64: Directorio de las imágenes negativas.

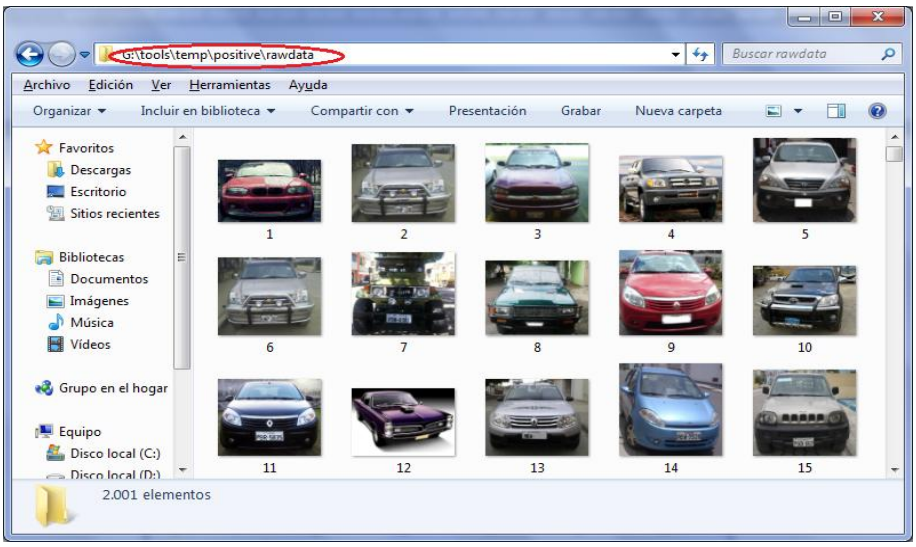


Figura 65 . Directorio de las imágenes positivas.

En el directorio donde se ubicaron las imágenes negativas, se encuentra el archivo **create_list.bat**, lo ejecutamos y generará un archivo create_**list.txt.**

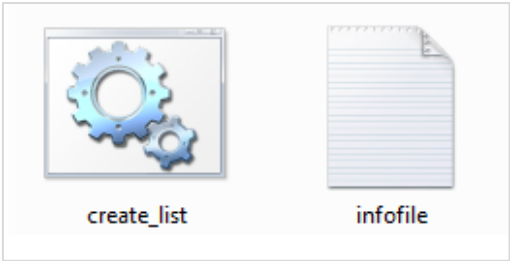


Figura 66: Archivo .bat y .txt generado.

El archivo infolile.txt contiene el nombre de las 5000 imágenes negativas con su respectiva extensión.

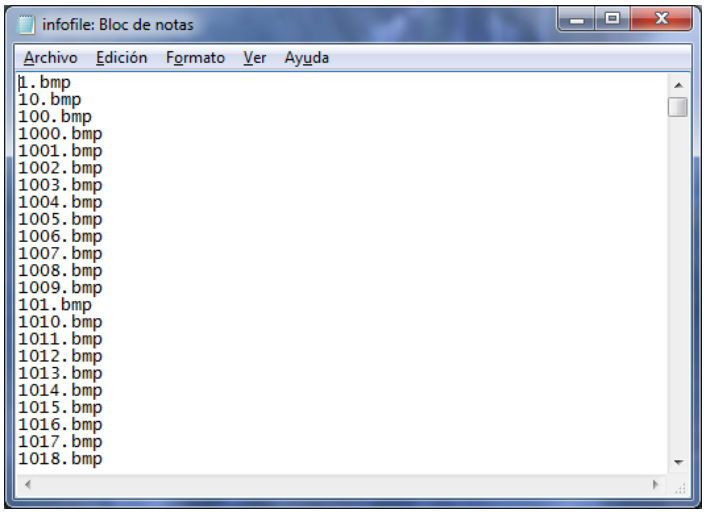


Figura 67: Contenido del archivo infofile.txt.

En el directorio de las imágenes positivas se encuentra el archivo **objectmarker.exe**, lo ejecutamos y presentará las siguientes interfaces. La interfaz de la parte superior indica la imagen que contiene el objeto a reconocer. En la interfaz de la parte inferior indica la información adicional de la imagen que se generará al señalar el objeto, el cual se encierra en un rectángulo de color fucsia (Seleccionar con el mouse, con la barra espaciadora para generar los datos y presionar **enter** para pasar a la siguiente imagen). Se debe realizar todo este proceso para todas las imágenes positivas, es decir señalar el objeto en todas las imágenes.

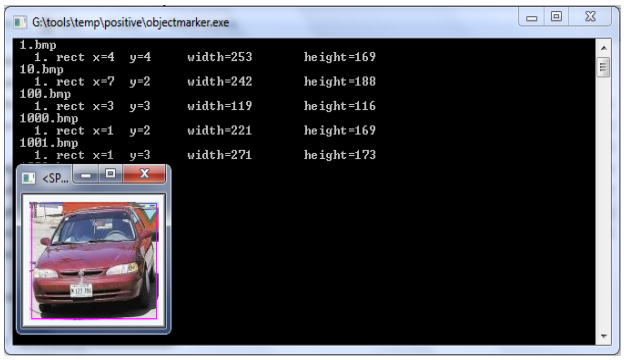


Figura 68: Objeto seleccionado e información generada.

La información adicional generada representa las coordenadas de la ubicación del objeto y las dimensiones de la parte seleccionada. Por ejemplo en la imagen 1.bmp al seleccionar el objeto, genera las coordenadas x=4 y=4 y las dimensiones 253x169.

Se debe realizar este proceso con todas las imágenes positivas, en este caso las 2000 imágenes. Cuando se finalice con todas las imágenes en el archivo info.txt se genera la información adicional especificada anteriormente.

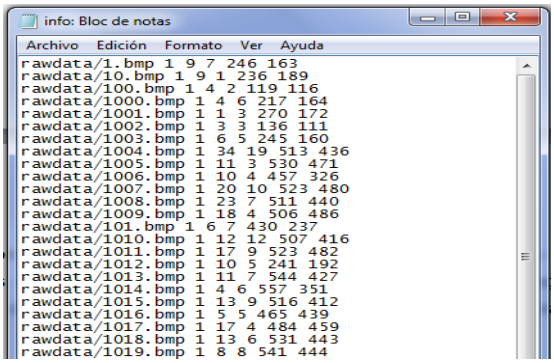


Figura 69: Información adicional generada.

Ahora procedemos a generar el archivo **.vec** el cual contendrá las muestras de cada una de las imágenes para ello utilizaremos la herramienta ***createsamples.exe***. En Windows debemos ejecutar lo siguiente:

createsamples.exe -info positive/info.txt -vec data/vector.vec -num 1800 -w 24 -h 24

Donde:

- **createsamples:** es el nombre de la herramienta
- **info:** es la ubicación del archivo con el índice las imágenes positivas
- **vec:** es el nombre del archivo de salida con la muestra generada
- **num:** cantidad de imágenes positivas
- **w:** ancho de la muestra de salida
- **h:** alto de la muestra de salida.

Para ello abrimos la consola de Windows **cmd** y le damos la ruta de la carpeta **temp** ubicada en **tools** en este caso está en el escritorio:

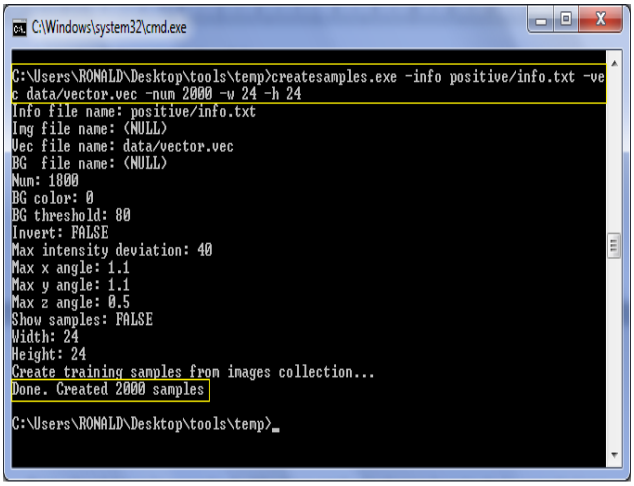


Figura 70: Creación del archivo .vec.

En el directorio ***\temp\data*** la carpeta **tools** se genera el archivo ***vector.vec***

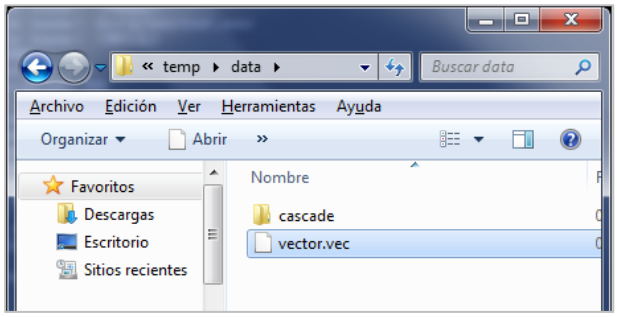


Figura 71: Archivo vector.vec creado.

Finalmente procedemos a realizar el entrenamiento. Para este proceso nos valdremos de otra herramienta llamada ***haartraining.exe*** ubicado en el directorio ***tools\temp*** y el archivo índice generado con las imágenes negativas y el archivo de muestras generado en la etapa anterior.

Ejecutamos lo siguiente:

haartraining.exe -data data/cascade -vec data/vector.vec -bg negative/infofile.txt -npos 2000 -nneg 5000 -nstages 20 -mem 1024 -mode ALL -w 24 -h 24 -nonsym

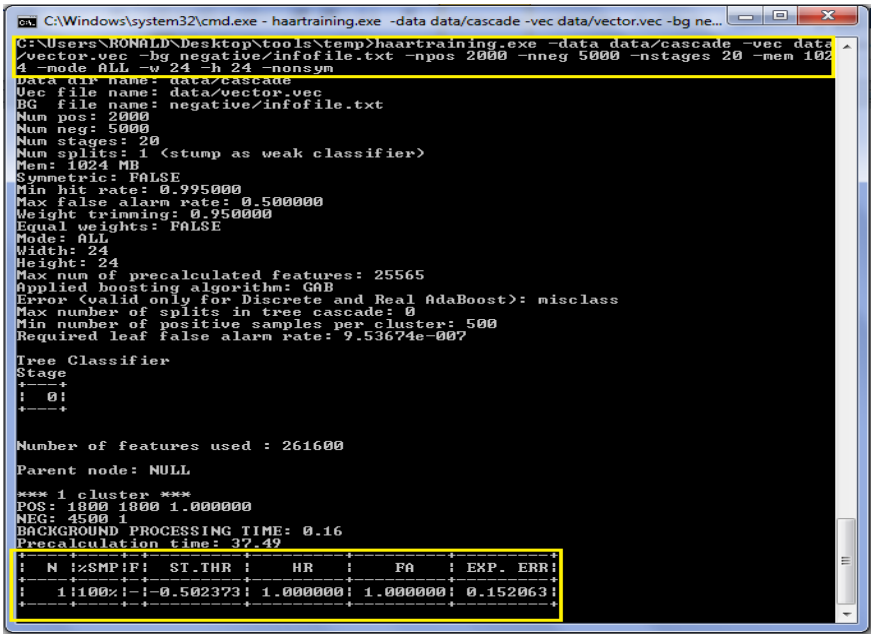


Figura 72: Inicio del entrenamiento generando la primera tabla.

Dónde:

- ***data data/cascade:*** nombre y dirección que le damos al clasificador
- **vec data/vector.vec:** nombre y dirección del archivo .vec creado anteriormente
- **bg negative/infofile.txt:** lista de imágenes negativas
- **npos 2000:** número de imágenes positivas
- **nneg 5000:** número de imágenes negativas
- **nstages 20:** número de fases o estados (A más fases, mejor clasificador).
- **mem 1024:** espacio en memoria asignado para el entrenamiento

- **mode ALL:** método recomendado para el entrenamiento
- **w 24 -h 24:** tamaño exacto utilizado en la creación de las muestras
- **nonsym:** Simetría, si el objeto a buscar es simétrico verticalmente será un entrenamiento más rápido.

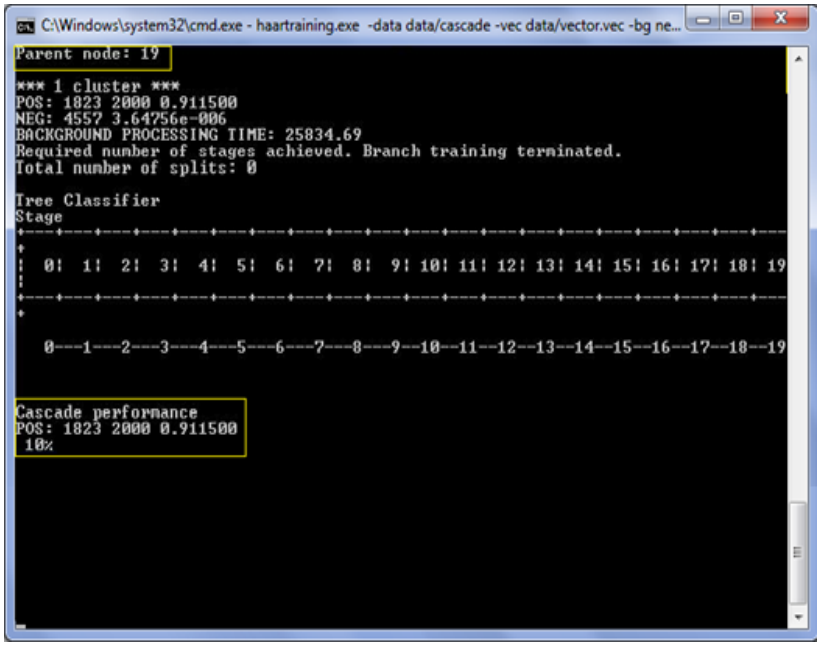


Figura 73: Finalizando el entrenamiento generando la tabla 19.

Al finalizar el proceso de entrenamiento, en el directorio **cascade2xml\data** de la carpeta **tools**, se generan 20 carpetas numeradas desde 0 a 19, cada una contiene una fase o estado del proceso de entrenamiento representado en archivos .**txt.**

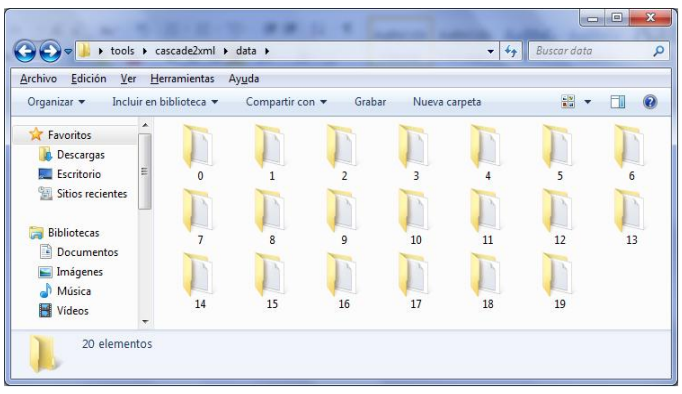


Figura 74: Carpetas que contienen los estados generados.

Ahora generamos el archivo con extensión **.xml**, el cual será nuestro clasificador final para ello utilizaremos la herramienta convert.bat del directorio ***tools\cascade2xml***, lo ejecutamos y se crea un archivo con el nombre ***output.xml*** que es el clasificador final que se utilizará para el reconocimiento de los vehículos.

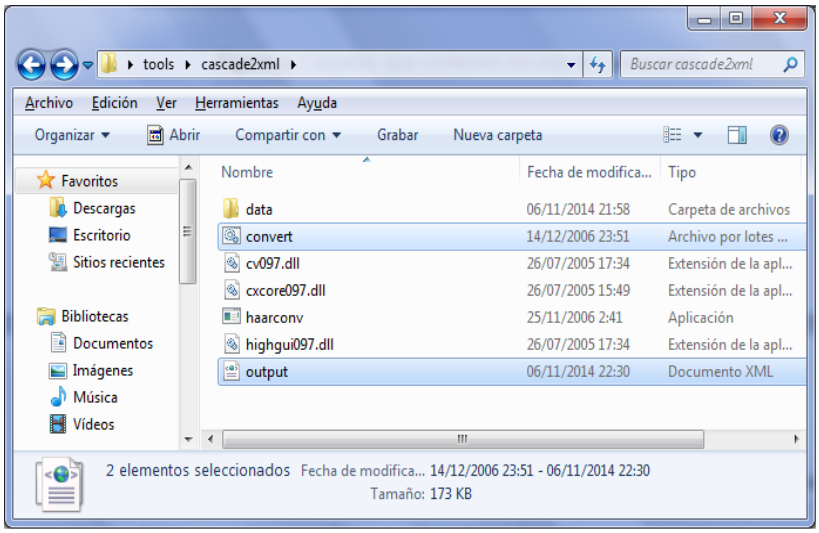


Figura 75: Conversión a xml del archivo final (clasificador)

El proceso de entrenamiento dura varios días dependiendo del número de imágenes utilizadas, mientras más imágenes se utilicen mejor será el reconocimiento y la detección de los objetos. Este proceso duró alrededor de 4 días en un computador core i5 con memoria RAM 4 GB.

El entrenamiento, y por consiguiente el clasificador está estructurado en forma de árbol, para construir un clasificador robusto debería tener por lo menos dos nodos por fase (nsplits), pero a consecuencia de dicha estructura el tiempo de computación crecerá exponencialmente según avancen las fases. Parámetros muy altos harán bloquearse al PC, nunca debemos usar toda la memoria del ordenador. La cantidad de variables que existen independientes de estos parámetros (tamaño de las imágenes, procesador, tamaño del patrón, numero de imágenes positivas y negativas...) en este proceso hace imposible hacerse una idea de lo que va a tardar, por eso es bueno empezar con valores modestos e ir incrementándolos, teniendo en cuenta que una buena estimación para que el clasificador sea robusto es más o menos una semana de entrenamiento, siendo menos importante cuales fueron los parámetros que se

mejoraron. Podemos empezar por ejemplo con 2 splits, 0.998 de minhitrate, y 14 stages (muy orientativo) [28].

No está preparado para procesadores de más de un núcleo, este problema hace que el entrenamiento vaya más rápido en una portátil, es más, mientras se ejecuta se puede comprobar en el administrador de tareas está usando exactamente un 25%.

3.1.1.2 Clasificador en cascada para la detección de matrículas vehiculares

Se realizó en la plataforma Ubuntu 14.04, para su construcción se emplearon 1000 imágenes positivas, es decir, imágenes que contienen placas vehiculares (objeto a detectar) y 2000 imágenes negativas que no contienen el objeto a detectar.
Cabe recalcar que para la construcción de este tipo de clasificadores, se debe tener previamente instalado y configurado Opencv (Ver Anexo 2).

Primeramente se debe crear una carpeta para almacenar las imágenes positivas y negativas que formarán parte del clasificador, en este caso se creó la carpeta **Imágenes** en el escritorio de Ubuntu. Allí se almacenaron dos carpetas más: una con las 1000 imágenes positivas y otra con las 2000 imágenes negativas, esto con el fin de facilitar la creación de nuestro clasificador.

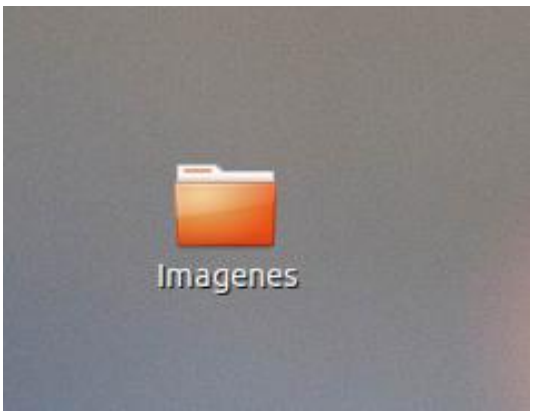


Figura 76: Creación de la carpeta contenedora Imágenes.

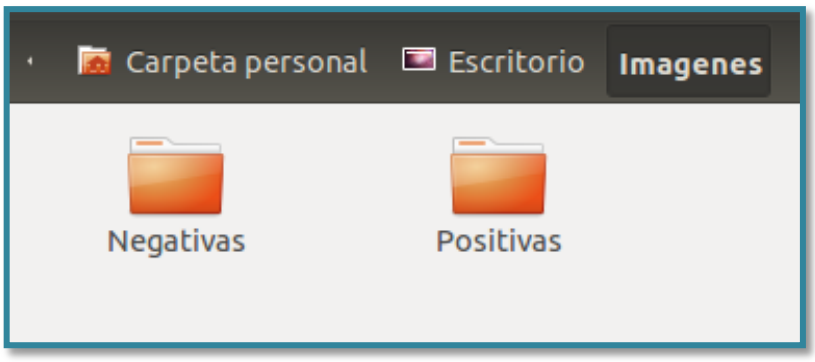


Figura 77: Imágenes positivas y negativas dentro de la carpeta Imágenes.

Figura 78: Imágenes positivas de tamaño 80 x 40 píxeles y formato .jpg.

Figura 79: Imágenes negativas de cualquier tamaño pero con formato .jpg.

Luego debemos abrir el directorio donde se halla la carpeta **Imágenes:**

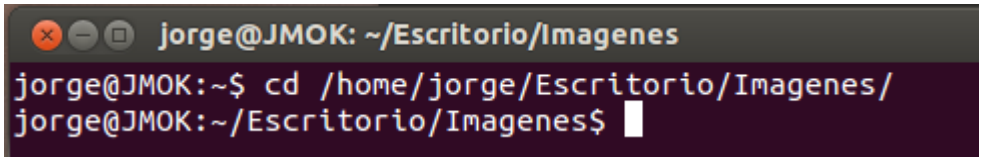


Figura 80: Directorio de la carpeta Imágenes.

En este directorio debemos crear dos archivos, un archivo denominado **positivas. Info** en base a las imágenes positivas y otro denominado **negativas.txt** sobre las imágenes negativas:

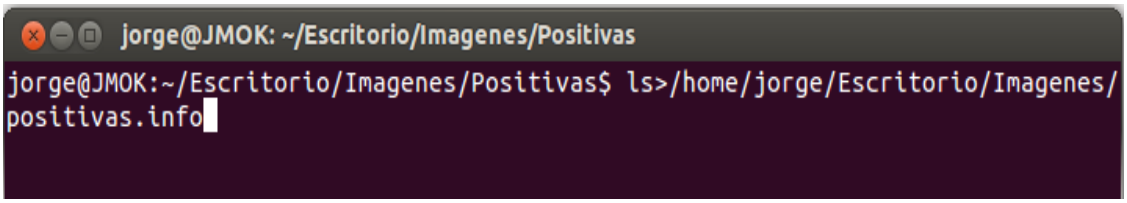


Figura 81: Creación del archivo positivas.info.

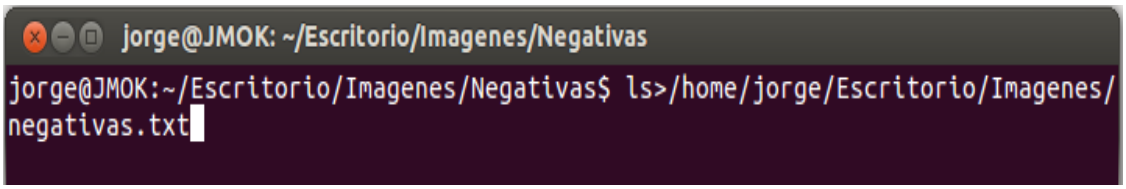


Figura 82: Creación del archivo negativas.txt.

Ahora debemos editar los archivos creados anteriormente. Para el caso del archivo **positivas.info,** abrimos el archivo, luego seleccionamos todo su contenido y presionamos el tabulador (\t) y escogemos la opción **Reemplazar** de la pestaña **Buscar** y sustituimos sus valores, como se muestra en la siguiente figura:

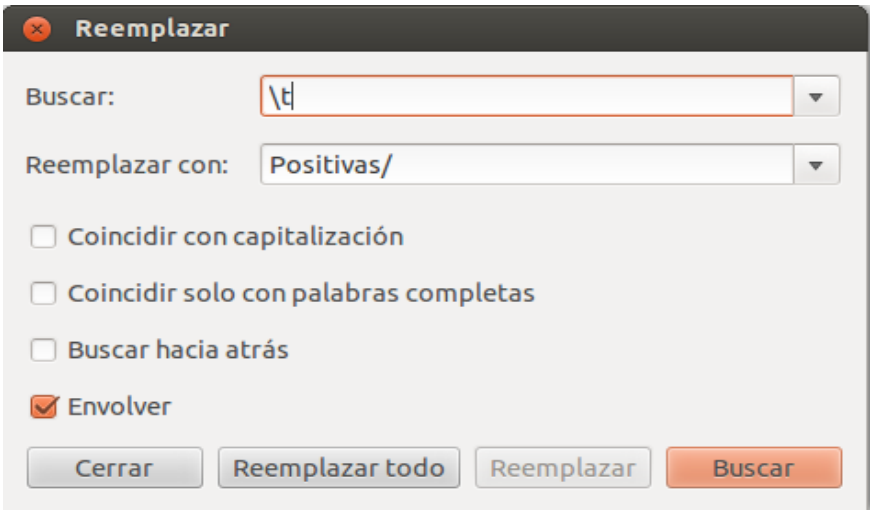


Figura 83: Proceso de edición del archivo positivas.info, parte del tabulador.

También debemos editar la siguiente parte del archivo:

Figura 84: Proceso de edición del archivo positivas.info, parte de la extensión.

Y nos quedaría un archivo más o menos de la siguiente manera:

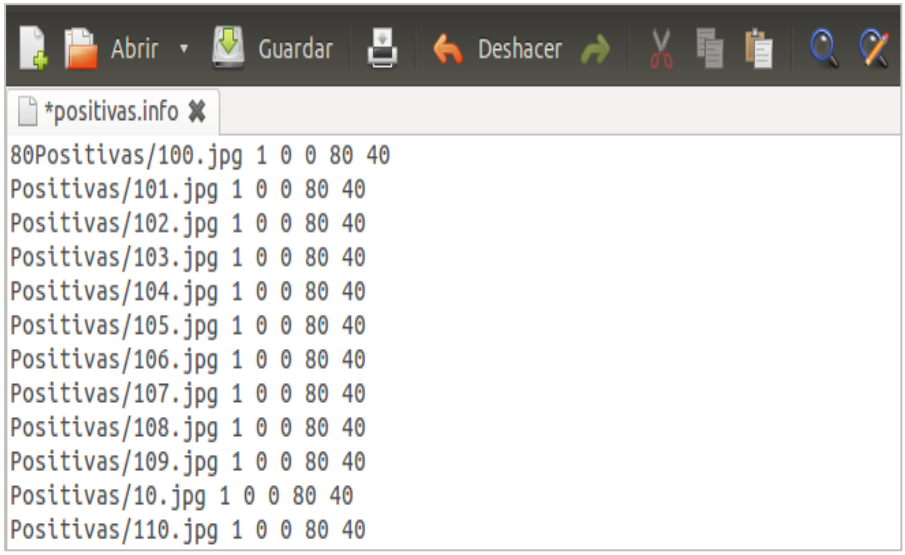


Figura 85: 84. Archivo positivas.info editado.

Ya hemos editado el archivo **positivas.info,** ahora nos queda editar el archivo **negativas.txt.** Para esto hacemos el mismo proceso de edición que para el archivo anterior, pero únicamente editando la parte del tabulador.

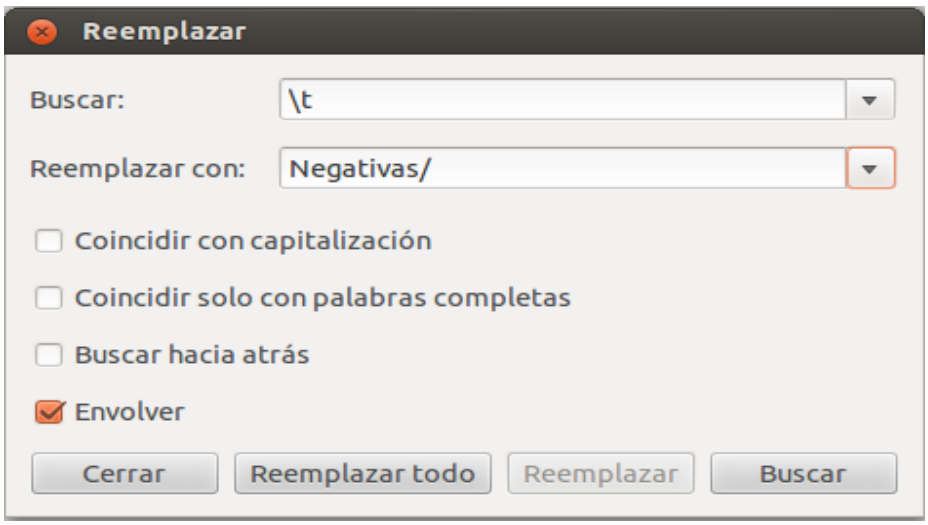


Figura 86: Proceso de edición del archivo negativas.txt.

El siguiente paso consiste en crear el archivo .vec que en este caso se denomina **placas.vec,** para lo cual debemos ubicar el directorio de la carpeta **Imágenes** y hacer uso de los archivos creados anteriormente.

Figura 87: Creación del archivo placas.vec dentro de la carpeta Imágenes.

En la figura anterior se puede observar que se está especificando el número 1000, pues este número corresponde al número de imágenes positivas mencionadas anteriormente. Además se está haciendo uso de los archivos **positivas.info** y **negativas.txt.**

Para comprobar que el archivo **placas.vec** se creó correctamente, al ejecutar el comando anterior debemos obtener algo similar a lo siguiente:

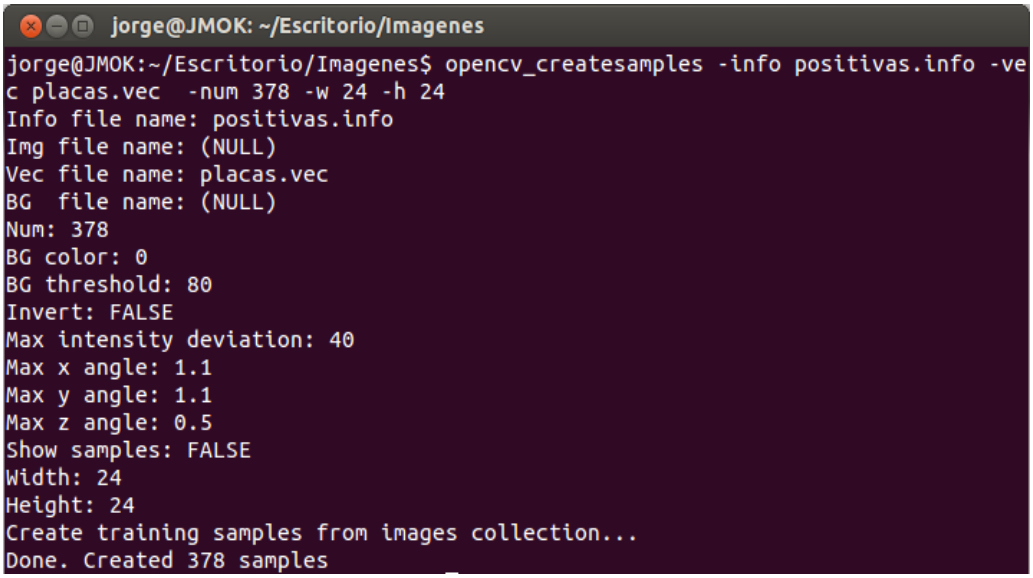


Figura 88: Proceso de creación del archivo placas.vec.

Figura 89: Verificación del archivo placas.vec.

Para finalizar el proceso de construcción del clasificador se debe crear una carpeta para almacenar los archivos .xml que se nos generen durante el proceso de entrenamiento, en este caso se creó la carpeta **Clasificador** dentro de la carpeta **Imágenes:**

Figura 90: Creación de la carpeta Clasificador.

Regresamos nuevamente a la consola de comandos y digitamos el siguiente comando:

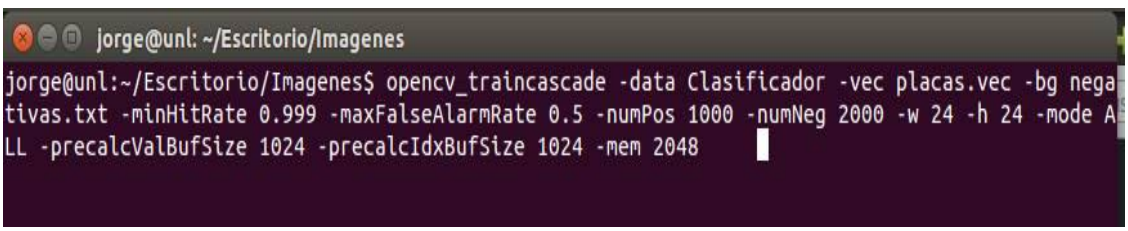


Figura 91: Comando para generar los archivos .xml del clasificador.

En el comando **–data** debemos especificar la carpeta donde se alojarán los archivos .xml, en nuestro caso es en la carpeta **Clasificador.** En el comando **–vec** el archivo .vec que se creó anteriormente, de igual manera en **–bg** el archivo **negativas.txt** que se creó previamente.

En la siguiente figura se muestra el proceso de generación de archivos .xml que serán los que se utilicen más adelante en la sección de implementación.

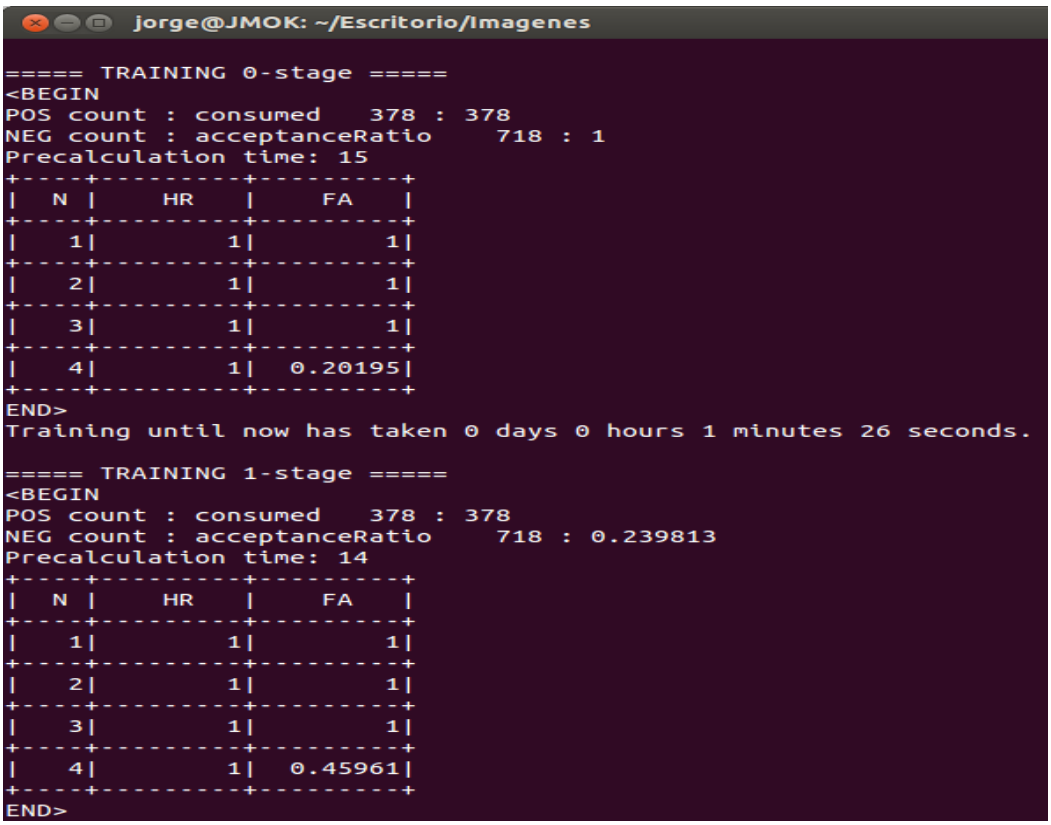


Figura 92: Proceso de generación de archivos .xml del clasificador.

3.1.2. Implementación del clasificador en cascada en el algoritmo para la detección de vehículos

Luego de haber construido el clasificador en cascada para la detección de vehículos, el siguiente paso consiste en implementarlo en el Sistema.

En la siguiente figura se puede observar tal implementación.

```
CascadeClassifier clasificadorVehiculos;
cap.read(a1);
//Clasificador para la detección de Vehículos
clasificadorVehiculos.load("/home/jorge/TesisCis/VehicularUNL/P1/clasificadores/output.xml");
vector<Rect> vdetectados;
clasificadorVehiculos.detectMultiScale(a1,vdetectados,1.2,3,0,cvSize(30,30));
for (size_t var = 0; var < vdetectados.size(); ++var) {
   rectangle(a1,Point(vdetectados[var].x,vdetectados[var].y), Point(vdetectados[var].x+vdetectados[var].width,
                     vdetectados[var].y+ vdetectados[var].height),cvScalar(0,255,0));
   putText(a1,"Vehiculo",cvPoint(vdetectados[var].x,vdetectados[var].y),CV_FONT_NORMAL,0.6,cvScalar(170,0,0),2,8);
   s1= a1(vdetectados[var]);
```

Figura 93: Implementación del clasificador para la detección de vehículos

3.1.3. Implementación del clasificador en cascada en el algoritmo para la detección de matrículas vehiculares

La implementación de este algoritmo es muy similar al de detección de vehículos, debido a que ambos se construyeron de manera similar pero con diferente propósito y diferentes tipos de imágenes.

A continuación se puede observar la implementación del clasificador empleado en la detección de matrículas vehiculares.

```
CascadeClassifier clasificadorPlacas;
//Clasificador para la detección de Placas
clasificadorPlacas.load("/home/jorge/TesisCis/VehicularUNL/P1/clasificadores/cascade.xml");
vector<Rect> pdetectadas;
clasificadorPlacas.detectMultiScale(a1,pdetectadas,1.2,3,0,cvSize(10,10));

for (unsigned int variable = 0; variable <pdetectadas.size(); ++variable) {
   rectangle(a1,Point(pdetectadas[variable].x,pdetectadas[variable].y), Point(pdetectadas[variable].x+pdetectadas[variable].
            width,pdetectadas[variable].y+ pdetectadas[variable].height),cvScalar(170,0,0));
   putText(a1,"Placa",cvPoint(pdetectadas[variable].x,pdetectadas[variable].y),CV_FONT_NORMAL,0.6,cvScalar(0,170,127),2,8);
   p1= a1(pdetectadas[variable]);
```

Figura 94: Implementación del clasificador para la detección de placas.

4. Fase 4. Verificación del Sistema

Luego de haber construido e implementado los clasificadores en los algoritmos correspondientes, es momento de comprobar la precisión de los mismos, a continuación se describen los resultados obtenidos.

4.1. Detección de vehículos

Inicialmente para la construcción del clasificador en cascada para la detección de vehículos se emplearon 500 imágenes positivas y 1200 imágenes negativas, obteniendo resultados no muy efectivos.

Para conseguir mejores resultados se optó por emplear 2000 muestras, es decir, imágenes positivas y 5000 imágenes negativas en la construcción del clasificador en cascada para la detección de vehículos.

De 50 vehiculos que pasaron frente a la cámara para ser detectados a una distancia aproximada de 4 a 10 metros y con una iluminación adecuada, se obtuvieron los siguientes resultados:

Tabla 10: PORCENTAJE DE PREDICCIÓN DEL CLASIFICADOR PARA LA DETECCIÓN DE VEHÍCULOS.

Muestras	Imágenes	Detectadas	Porcentaje
500	Vehículos	29	58%
2000	Vehículos	49	98%

En la siguiente figura se puede observar de manera gráfica en términos estadísticos lo anteriormente detallado:

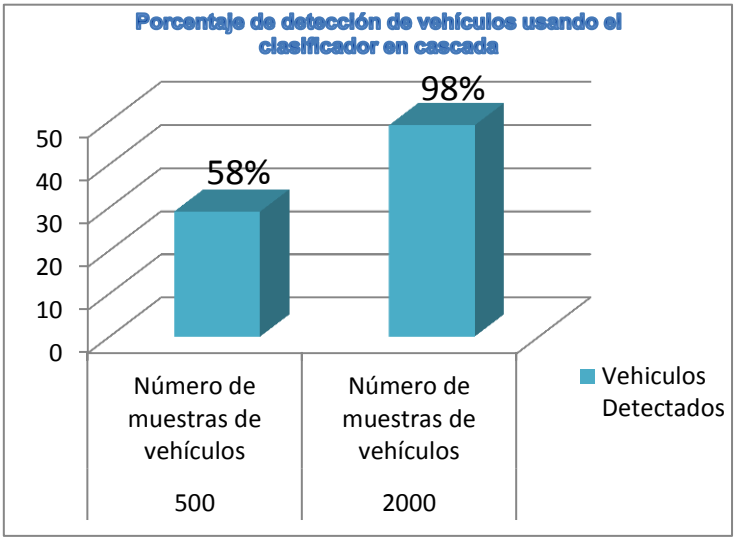


Figura 95: Resultados estadísticos de la detección de vehículos.

Es evidente que mientras más muestras se utilicen, mejores serán los resultados que se obtengan. Al utilizar únicamente 500 imágenes para la construcción del clasificador que se encargará de realizar la detección de vehículos (autos). De 50 vehículos analizados, sólo fueron detectados por el clasificador 29 vehículos, es decir, el 58%, incluso llegando a detectar otros objetos que no son autos como por ejemplo motos u otro tipo de objetos en particular.

En la siguiente figura se muestra la detección de vehículos usando el clasificador inicial, que consistió en usar 500 imágenes positivas y 1200 negativas.

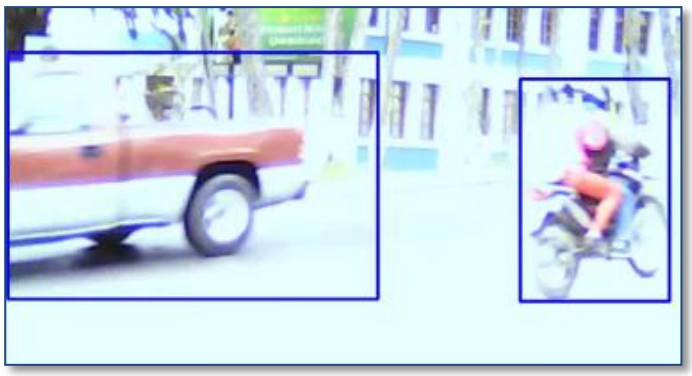


Figura 96: Detección de vehículos usando pocas muestras.

También se realizaron pruebas a escala empleando vehículos de juguete, de igual manera no se consiguieron resultados factibles.

Figura 97: Detección de vehículos a escala usando pocas muestras.

En cambio al utilizar 2000 imágenes positivas como muestras y 5000 imágenes negativas para la construcción del clasificador que realizará la detección de vehículos se consiguieron grandes resultados, los mismos que se muestran a continuación:

Figura 98: Detección de vehículos usando gran cantidad de muestras.

4.2. Detección de matrículas vehiculares

Como ya se mencionó anteriormente, para realizar la detección de matrículas vehiculares se tuvo que construir un clasificador conocido como clasificador en cascada, empleando 1000 imágenes como muestra (positivas) y 2000 imágenes negativas.

Los resultados obtenidos fueron excelentes, pero asumiendo ciertas restricciones, por ejemplo que el vehículo que contiene la matrícula que queremos detectar debe estar a una distancia aproximada de 2 a 4 metros en frente de la cámara y no debe existir contraluz.

Por cada 50 vehículos que pasaron en frente de la cámara se obtuvieron los siguientes resultados:

Tabla 11: PORCENTAJE DE PREDICCIÓN DEL CLASIFICADOR PARA LA DETECCIÓN DE MATRÍCULAS VEHICULARES.

Muestras	Imágenes	Detectadas	Porcentaje
1000	Placas	47	94%

En la siguiente figura se puede apreciar en términos estadísticos los resultados obtenidos en la detección de matrículas vehiculares:

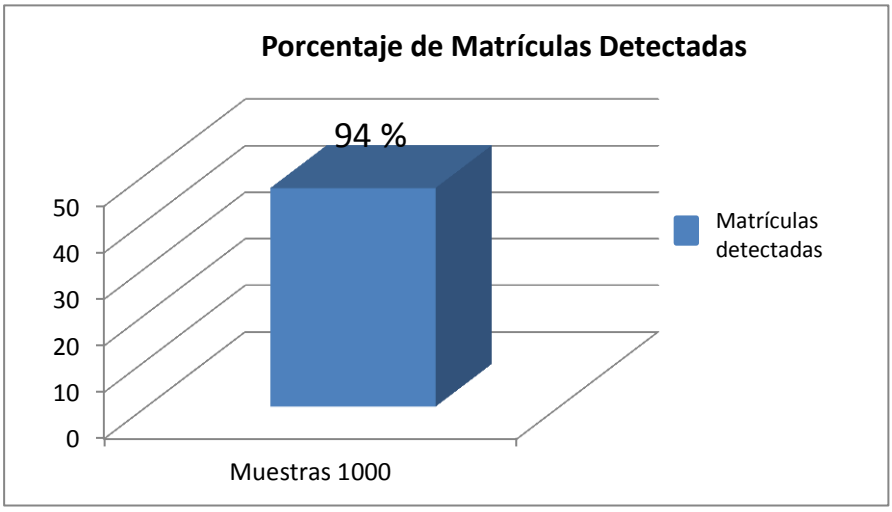


Figura 99: Resultados estadísticos de la detección de matrículas

En la siguiente figura se puede apreciar la detección de matrículas vehiculares haciendo uso del clasificador en cascada.

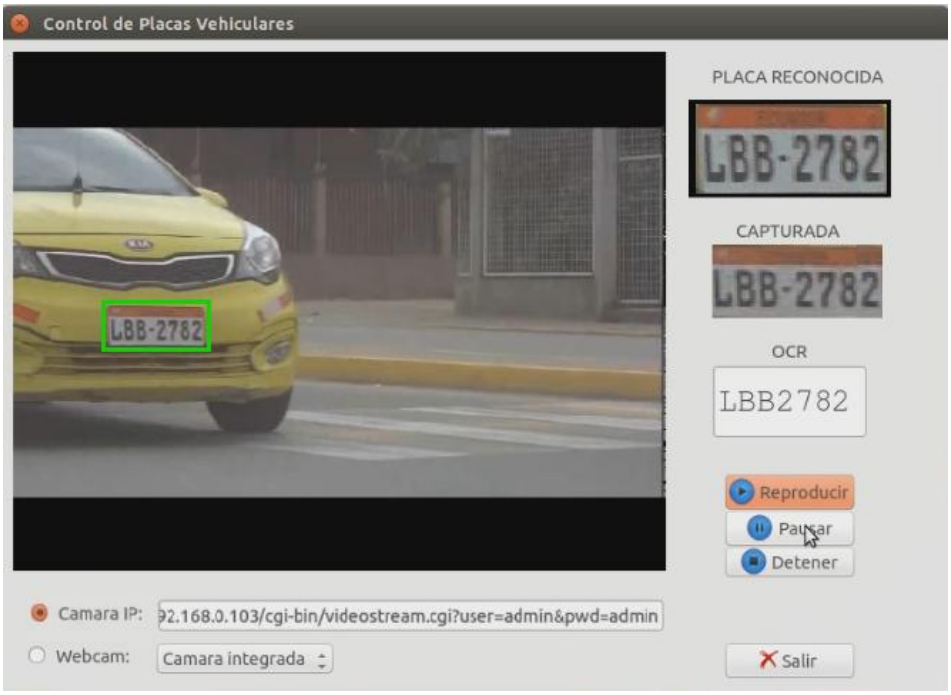


Figura 100: Detección de matrículas usando el clasificador en cascada

En la figura anterior se puede ver claramente que una vez detectada la matrícula, se la muestra en la parte superior derecha de la ventana, para luego ser procesada y reconocida por el algoritmo encargado de efectuar el reconocimiento óptico de caracteres.

4.3. Detección y captura de matrículas vehiculares

Luego de haber visto cómo se realiza la detección de vehículos y matrículas, vamos a presentar la captura de estas matrículas que serán almacenadas en una base de datos y se obtendrán reportes de los mismos.

Figura 101: Detección y captura de matrículas usando el clasificador en cascada

4.4. Análisis del Reconocimiento de matrículas vehiculares usando OCR

La detección de matrículas vehiculares nos abre un diálogo que nos permite realizar un análisis acerca del reconocimiento de las matrículas vehiculares. Para ello hay que tomar en cuenta algunos puntos clave, los cuales se detallan a continuación:

4.4.1. Capacidad de la cámara y calidad del video

La capacidad de la cámara debe ser considerada muy estrictamente. La cámara utilizada en este tipo de proyectos debe tener una excelente resolución de video, es decir, para que la detección sea muy buena, la cámara debe tener por lo menos un Pixelaje de 16-20 MP.

En el presente proyecto se usó una cámara IP marca APEXIS cuya resolución es de 3 MP, en lo que se refiere a la detección de matrículas vehiculares no existe ningún problema, pero sí en el reconocimiento de caracteres (OCR), esto se complica debido a que la imagen de la matrícula vehicular que es detectada y capturada esta borrosa, lo cual implica que el reconocimiento no sea el correcto como se muestra en la siguiente imagen.

Figura 102: Aplicación del OCR a las placas capturadas de baja calidad

Podemos evidenciar claramente que al aplicar el OCR a la imagen capturada existen caracteres que no corresponden a la matrícula vehicular. En la imagen de la matrícula pueden ser apreciados por el ojo humano los caracteres PNS 043 y al usar el OCR se reconocen los caracteres 3F4X5ML, todo esto se debe a lo anteriormente mencionado.

4.4.2. Distancia entre el Vehículo y la cámara

La cámara debe estar cerca del lugar donde se pueda ver con claridad la matrícula del vehículo para que pueda ser detectada claramente. La distancia debe ser de 2-4 metros de longitud ya que lo que más se complica es el reconocimiento (OCR) de los caracteres.

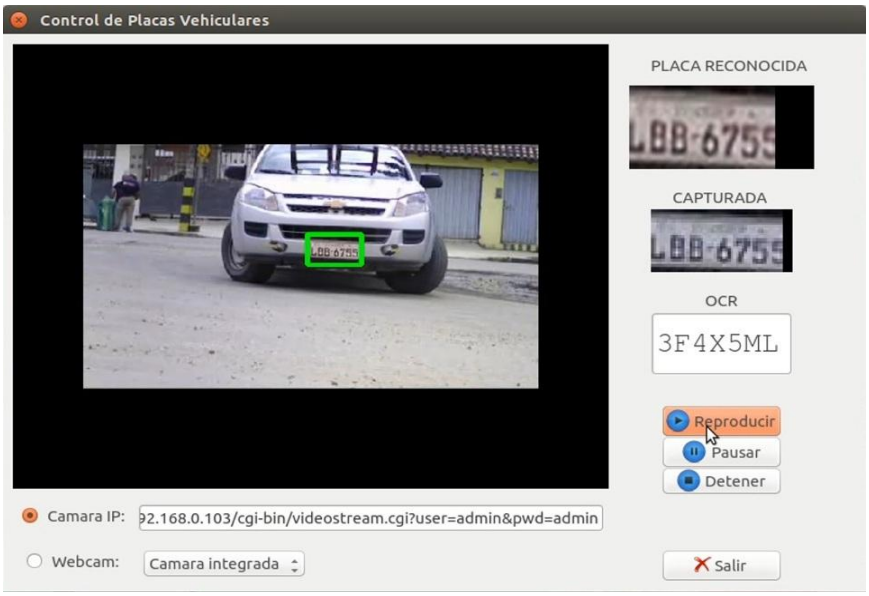


Figura 103: Reconocimiento de caracteres considerando la distancia.

4.4.3. Ubicación de la cámara

La cámara debe estar perfectamente ubicada en un ángulo donde la matrícula vehicular pueda ser detectada y capturada en su totalidad. De lo contrario existirían problemas en la detección, así como una captura de una matrícula incompleta, es decir, faltándole letras o números, como lo muestra la siguiente figura.

Figura 104: Placas que no son detectadas correctamente.

4.4.4. Estado de la placa vehicular

Sin duda es un punto muy importante a considerar, ya que se convierte en un problema o en una desventaja en lo que respecta a la detección. Si la placa se encuentra muy deteriorada el reconocimiento de caracteres de dicha placa vehicular no se realizará de manera adecuada, podrían aparecer caracteres que no pertenecen la placa o simplemente no sería detectada.

Figura 105: Placas que no son detectadas.

4.4.5. Perspectiva del vehículo

Algo complicado de controlar ya que dependiendo de la anchura de la vía por donde circula el vehículo, este puede tomar varias perspectivas que dificultan la detección de la matrícula.

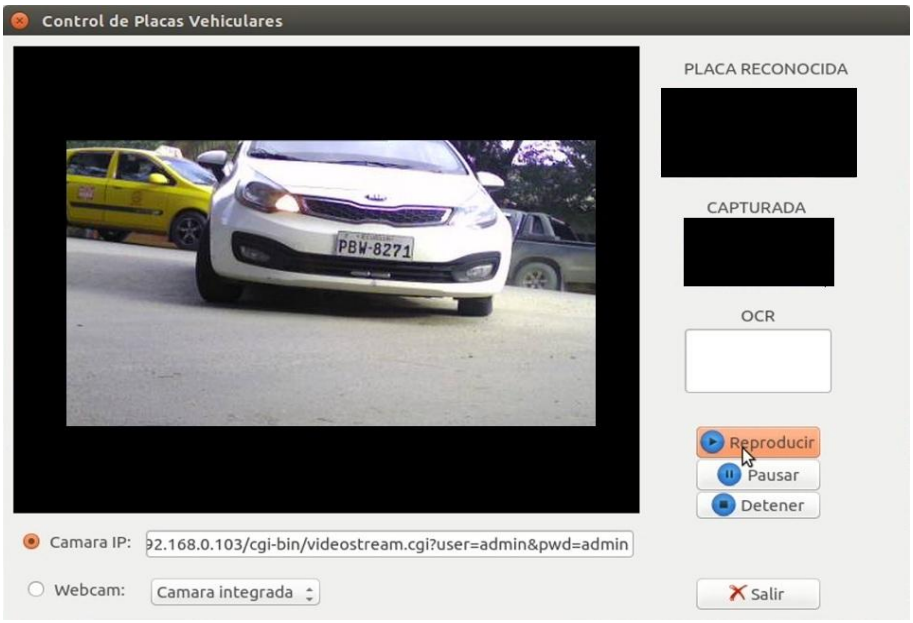


Figura 106: Placas que no son detectadas en diferentes perspectivas.

4.4.6. Resultados del OCR en una correcta ubicación de la cámara y con una excelente calidad de video

Estas tomas son de una cámara digital cuya resolución de video es de 16.1 MP, la ubicación está a un costado de la vía por donde circulan los vehículos a una distancia de 4 metros aproximadamente.

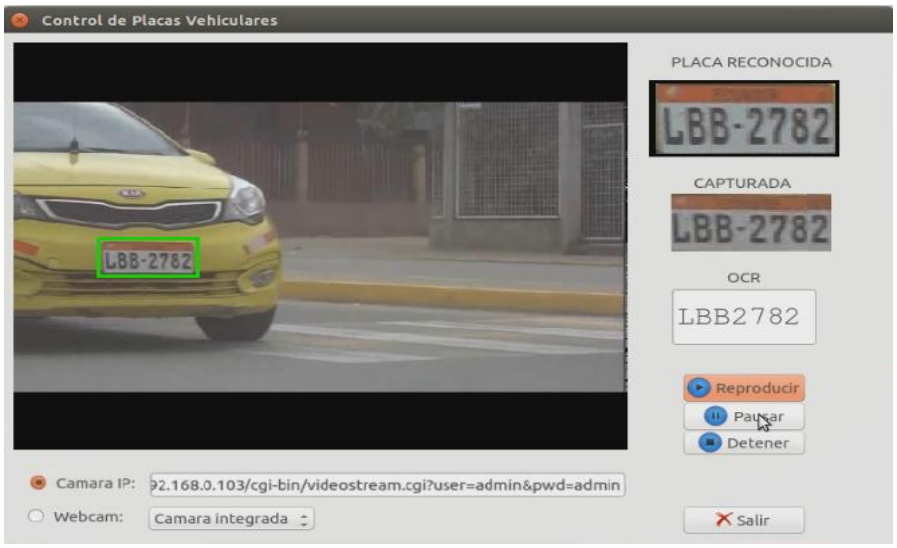


Figura 107: .Reconocimiento efectivo de caracteres.

4.4.7 Detección de placas vehiculares cuando la cámara está en movimiento y reproducción lenta respecto del vehículo

Además de las consideraciones resaltadas anteriormente, es necesario tener en cuenta que cuando la cámara que se está utilizando para realizar la detección de las matriculas vehiculares está en movimiento, esto con respecto al vehículo, los resultados esperados no son buenos, ya que el movimiento es un factor que incide directamente en la detección y reconocimiento de la matrícula del vehículo, podría detectar imágenes que no tienen nada que ver con el objetivo de la aplicación, además el OCR tampoco sería eficiente en el reconocimiento. También cabe recalcar que la reproducción del video ser normal, es decir no debe ser lenta, esto depende mucho de la capacidad de la máquina.

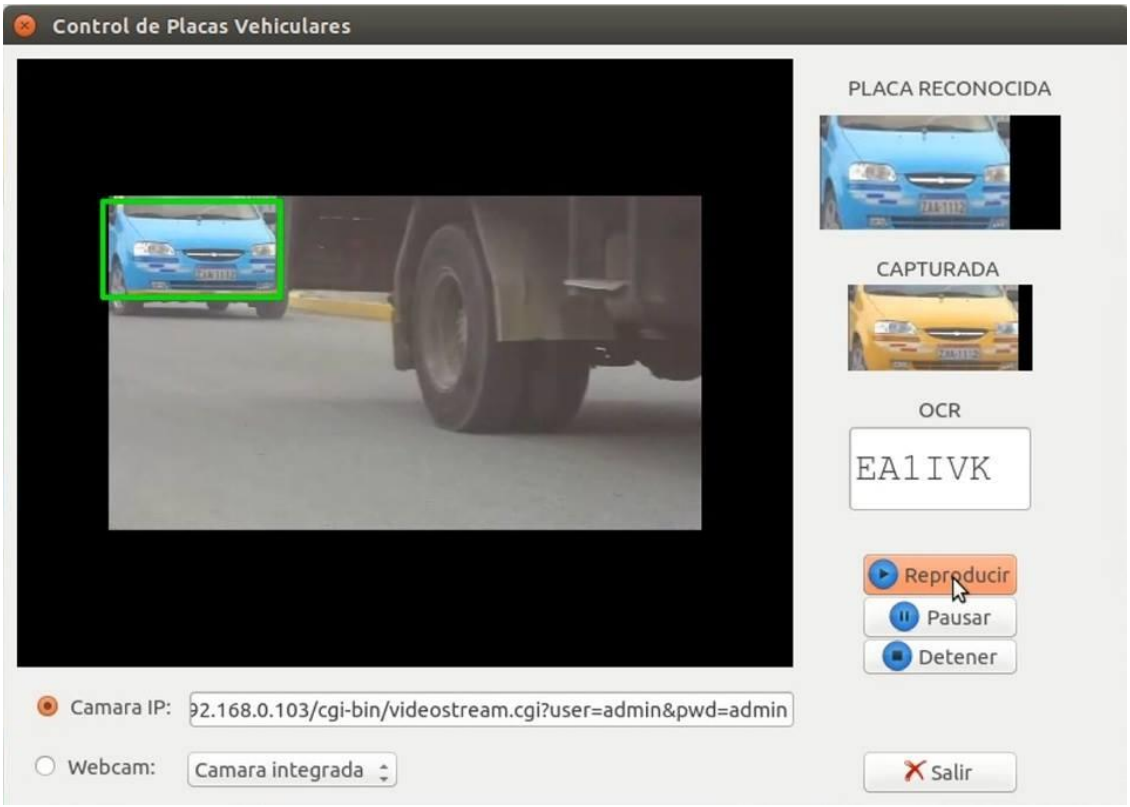


Figura 108: Detección de objetos cuando la cámara está en movimiento.

4.5 Generación de reportes

En la **Fig. 106** claramente se puede observar que una placa luego de ser detectada es capturada y presentada, para luego aplicarle el reconocimiento óptico de caracteres, y posteriormente almacenada en una base de datos MySQL, de la cual se generan los reportes, tanto de las fotos de las placas que son detectadas como los caracteres de las mismas. En la **Figura 107** se puede ver como se generan los reportes de las placas mediante fotos y en la **Figura 108** las placas vehiculares en caracteres.

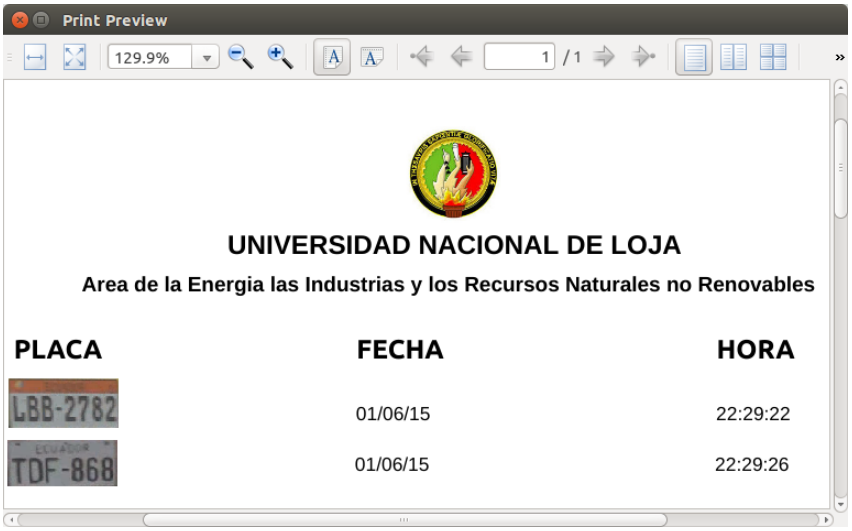


Figura 109: Reportes de Fotos de las Placas

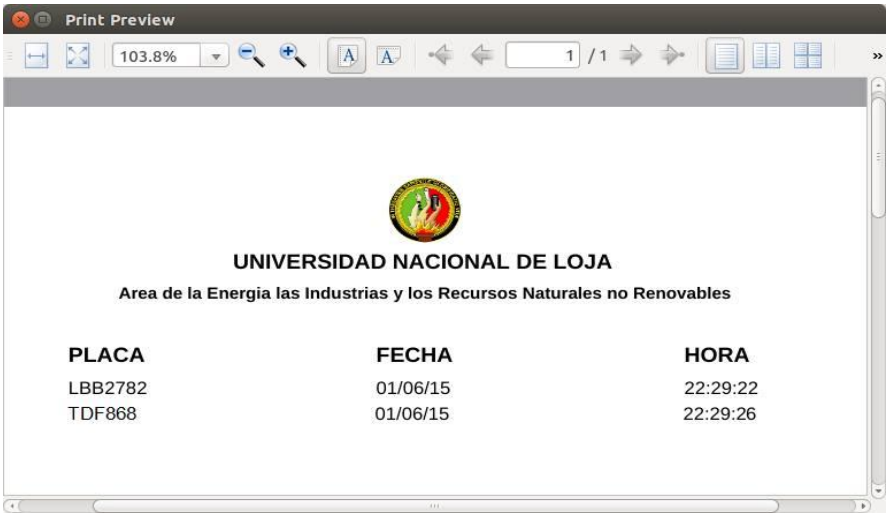


Figura 110: Reportes de caracteres de las Placas

g. Discusión

Luego de haber concluido con el desarrollo del proyecto de fin de carrera titulado "**Desarrollo de un sistema basado en Visión Artificial para el reconocimiento de matrículas vehiculares en el AEIRNNR de la UNL**" es momento de realizar un análisis acerca de los objetivos establecidos para el desarrollo del presente proyecto, con el fin de evaluar su cumplimiento, los mismos que a continuación se describen:

- Identificar que los objetos que ingresan al AEIRNNR son vehículos a través de la Visión Artificial

Pues este logro es evidente ya que para conseguirlo se tuvo que utilizar un algoritmo para realizar la detección de vehículos, para lo cual se construyó un clasificador denominado clasificador en cascada. Gracias a este algoritmo se consiguieron excelentes resultados en lo que respecta a identificar que los objetos que ingresan al AEIRNNR son vehículos para posteriormente realizar la detección de las placas vehiculares. Aunque si existe una restricción referente a la iluminación, porque cuando hay excesiva luz o contraluz, o en la noche los resultados que se obtienen no son los adecuados.

✓ Reconocer y capturar una foto de las matrículas de los vehículos

Luego de que se haya detectado un vehículo, el siguiente paso es realizar la detección de matrículas para lo cual se empleó otro algoritmo encargado de realizar esa tarea, para luego cumplir con el objetivo establecido que consiste en realizar el reconocimiento y captura de las matrículas, evidentemente para realizar este reconocimiento y captura se emplea un clasificador en cascada el cual se encarga en realizar una clasificación de objetos de interés, mediante la función haartraining proporcionada por Opencv.

Además en esta parte hay que destacar que existen algunos problemas, las placas son detectadas únicamente cuando la cámara se encuentra en frente del vehículo y a una distancia de 2 a 4 metros y además hay que tener en cuenta que la iluminación sea la adecuada, porque de lo contrario el Sistema no realizará ninguna detección sea de vehículos o de placas. Y eso no es todo también hay que tener en cuenta que las

placas estén en buen estado, preferiblemente sean placas modernas y estén limpias para realizar un buen reconocimiento.

- ✓ Desarrollar el proceso de almacenamiento y generación de reportes digital, con los datos obtenidos de la Visión Artificial, para la toma de decisiones

Finalmente, luego de que se haya realizado el reconocimiento de placas o matrículas vehiculares el Sistema se encarga de almacenar los datos referentes a las mismas en una base de datos, la cual fue creada usando el gestor de bases de datos MYSQL y para los reportes se utilizó la herramienta NCReport V2.13.0 que trabaja con el IDE Qt V 5.2.1, generándonos un archivo en formato .PDF en donde constan los datos de las matrículas, constando la foto de la matrícula, la hora y fecha que fue registrada en el Sistema.

h. Conclusiones

Luego de haber concluido con el desarrollo del presente proyecto de fin de carrera, es necesario dar a conocer algunas conclusiones, las mismas que a continuación se detallan:

- El uso de clasificadores en cascada ha permitido realizar la detección de vehículos y de sus respectivas matrículas, con cierto margen de error en algunos casos debido a que no se asumieron ciertas restricciones.

- La metodología con enfoque en cascada es un buen camino a seguir cuando se trata de desarrollar sistemas, especialmente si están relacionados con el tema de Visión Artificial, ya que cuenta con etapas o fases que son comunes para cualquier sistema.

- Las librerías Opencv representan una ayuda excepcional dentro del campo de la Visión Artificial, puesto que existe gran cantidad de información y la posibilidad de integrarse con varios lenguajes de programación como C++ y entornos de desarrollo como QT Creator, los mismos que han sido aplicados en el presente proyecto.

- Para la creación de los clasificadores se utilizaron tanto imágenes negativas como positivas, para de esta manera poder diferenciar si existe o no el objeto a detectar, motivo por el cuál mientras más imágenes se tengan mejores serán los resultados que se obtengan.

i. Recomendaciones

Además de las conclusiones planteadas anteriormente, también es necesario realizar algunas recomendaciones a considerar al momento de desarrollar un Sistema basado en Visión Artificial. A continuación se detallan estas recomendaciones:

- Al desarrollar un Sistema basado en Visión Artificial es conveniente iniciar seleccionando cuidadosamente las herramientas a utilizar, tomando en cuenta el soporte, la utilidad y sobre todo la flexibilidad de las mismas.

- Utilizar herramientas libres en el desarrollo de Sistemas que hagan uso de la Visión Artificial para de esta manera lograr eficacia y eficiencia durante el proceso de desarrollo.

- Para el desarrollo de un proyecto enfocado en Visión Artificial es recomendable utilizar una metodología que sea común para cualquier otro tipo de Sistema como es la metodología con enfoque en cascada.

- Al desarrollar un Sistema con Visión Artificial hay que considerar los cambios climáticos en el medio ambiente ya que dependiendo de esto existirá poca o excesiva cantidad de luz solar, lo cual impide obtener buenos resultados, por lo tanto la luz es el principal elemento a considerar. También hay que tener en cuenta el lugar y la distancia a la que va a estar ubicado el sensor (cámara) del objeto a detectar.

- Además se recomienda que para obtener mejores resultados en lo que respecta a la detección de vehículos, matrículas o reconocimiento óptico de caracteres, la cámara debe estar ubicada en una posición fija para evitar que el sistema detecte o reconozca otras cosas que no son parte del objeto de interés.

Trabajos futuros

El desarrollo de este tipo de sistemas da lugar a nuevas líneas de investigación, sean por ejemplo la mejora del mismo o sirviendo de base para el desarrollo de otros Sistemas mucho más robustos. Siendo algunos de los trabajos futuros los siguientes:

- Desarrollo de un sistema con Visión Artificial para el cobro de peajes a través del reconocimiento de placas vehiculares.
- La implementación de este tipo de sistemas dentro de las gasolineras.
- Para los parqueaderos de vehículos
- Sistema base para un macro proyecto como el de Semáforos inteligentes.
- De ser posible también se podría migrar la aplicación a dispositivos móviles, por ejemplo a aquellos que usan Android como Sistema Operativo.

j. Bibliografía

[1] Platero Dueñas, Carlos. Introducción a la Visión Artificial. Dpto. Electrónica, Automática e Informática Industrial. Apuntes de Visión Artificial. [En línea]: http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf

[2] Mejía Vilet, José Ramón. Procesamiento digital de imágenes. 2005. [En línea]. http://read.pudn.com/downloads159/ebook/711796/Procesamiento_Digital_de_Imagenes.pdf

[3] Visión Artificial - EcuRed. (s. f.). Recuperado 21 de octubre de 2014, a partir de http://www.ecured.cu/index.php/Visi%C3%B3n_Artificial

[4] García Santillán, Elías. "Detección y clasificación de objetos dentro de un salón de clases empleando técnicas de procesamiento digital de imágenes". Universidad Autónoma Metropolitana. Tesis 2008.

[5] Ruiz Cabeza, Héctor. Infaimon su asesor en visión artificial. Disponible en: http://www.jcee.upc.edu/JCEE2010/pdf_ponencies/PDFs/25_11_10/INFAIMON-Vision%20artificial.pdf.

[6] Software de imagen | Sistemas de visión Artificial para INDUSTRIA | INFAIMON. [En línea]. Disponible en: http://www.infaimon.com/es/tecnologia-software. [Accedido: 21-oct-2014].

[7] Sobrado Malpartida, Eddie Ángel. "Sistema de Visión Artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot". Pontificia Universidad Católica del Perú. Tesis 2003.

[8] National Instrumenst. Una Plataforma, Infinidad de Posibilidades. [En línea]. Disponible en :http://www.ni.com/labview/why/esa/

[9] Image Processing Toolbox - MATLAB. [En línea]. Disponible en: http://www.mathworks.com/products/image/.

[10] Machine vision and imaging software - Matrox Imaging Library (MIL). [En línea]. Disponible en:

http://matrox.com/imaging/en/products/software/mil/?ref=imve728x90web. [Accedido: 21-oct-2014].

[11] NI Vision - Visión Artificial para Cualquier Aplicación - National Instruments. [En línea]. Disponible en: http://www.ni.com/vision/esa/. [Accedido: 21-oct-2014].

[12] Aplicaciones de la Visión Artificial. [En línea] http://dmi.uib.es/~ygonzalez/VI/Material_del_Curso/Teoria/Aplicaciones_VC.PDF

[13] Bercovich, Eduardo. Septiembre 2011. [En línea]. Disponible en: http://dreduardobercovich.blogspot.com/2011_09_01_archive.html. [Accedido: 21-oct-2014].

[14] Visión Artificial interacción sin mandos. Asignatura de Gráficos en Computación - 12/2010. Consultado en Línea en: http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/VisionArtificial/aplicaciones.html

[15] ROBINDUSTRIA - Sistemas de Visión Industrial. [En línea]. Disponible en: http://formacion.faico.org/Proyectos/ROBINDUSTRIA/SistemasDeVisionArtificial.htm. [Accedido: 23-oct-2014].

[16] Marqués Rossana, Acuña Carlos Javier. Barbeira María Pilar. Fundamentos de la imagen digital. [En línea]. http://valijas.ceibal.edu.uy/files/imagen_digital_distribucion.pdf

[17] LAURA T. Cámara digital vs ojo humano. Publicado 11-05-2012. Consultado en línea en: http://fotografosenpotencia.wordpress.com/2012/05/11/ojo-digital-vs-ojo-humano/

[18] Tipos de imágenes y formatos - APRENDE TIC. [En línea]. Disponible en: https://sites.google.com/site/ticvalcarcel/optimizacion-de-imagenes-para-internet/tipos-de-imagenes-y-formatos. [Accedido: 21-oct-2014].

[19] Cárdenas Hidalgo, Paul; Flores Vargas, José Alfredo; López Zavaleta, Jaime; Martínez Moreno, Pablo. “Diseño de Sistema de reconocimiento de placas utilizando MatLab”. Instituto Politécnico Nacional. Tesis 2009.

[20] Reconocimiento de Caracteres Ópticos (OCR) usando MatLab. [En línea]. http://www.matpic.com/esp/matlab/ocr.html

[21] Delgado Montiel, José Luis. “Reconocimiento de placas vehiculares”. Instituto Politécnico Nacional. Tesis 2010.

[22] Lara Rodríguez, Gustavo Adolfo. Técnicas de reconocimiento de imágenes para la creación de fotomosaicos. Guatemala. 2003. [En línea]. http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r25669.PDF

[23] Gutiérrez, Richard; Frydson, Ma., Fernanda; Vintimilla, Phd., Boris. Aplicación de Visión por Computador para el reconocimiento automático de Placa Vehiculares utilizando OCR’s Convencionales. [En línea]. http://www.dspace.espol.edu.ec/bitstream/123456789/19074/1/Paper_Gutierrez_Frydson_Vintimilla.pdf

[24] La mirada del Golem: Librerías de visión artificial. [En línea]. Disponible en: http://miradadelgolem.blogspot.com/2013/04/librerias-de-vision-artificial.html. [Accedido: 23-oct-2014].

[25] Placas de Ecuador (por provincias) - Foros Ecuador. [En línea]. Disponible en: http://www.forosecuador.ec/forum/aficiones/autos-y-motos/212-placas-de-ecuador-por-provincias. [Accedido: 21-oct-2014].

[26] Instalar OpenCV 2.4.2 en Ubuntu 12.04 LTS | desarrollophpsenior. [En línea]. Disponible en: http://desarrollophpsenior.wordpress.com/2012/09/19/instalar-opencv-2-4-2-en-ubuntu-12-04-lts/. [Accedido: 31-oct-2014].

[27] Opencv haartraining | Shehan’s blog. [En línea]. Disponible en: http://www.tectute.com/2011/06/opencv-haartraining.html. [Accedido: 07-nov-2014].

[28] IFNOTISNULL: Reconocimiento de patrones con OpenCV - Parte 1. [En línea]. Disponible en: http://ifnotisnull.blogspot.com/2012/01/reconocimiento-de-patrones-con-opencv.html. [Accedido: 07-nov-2014].

k. Anexos

Anexo 1. Instalación de Opencv 2.4.9 en Ubuntu 14.04 LTS

Para el desarrollo del presente proyecto, que consiste en un Sistema de Visión Artificial, previamente se tuvo que realizar la instalación y configuración de Opencv, para lo cual se utilizó una guía de instalación [26].

A continuación se detallan todos los pasos que se deben seguir para su instalación:

- En primera instancia se debe realizar una actualización del sistema, para lo cual se utilizan los siguientes comandos:

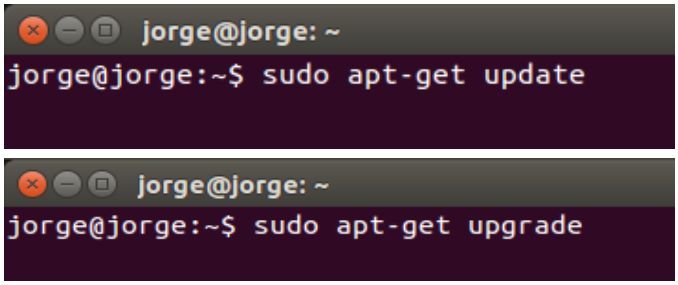


Figura 111: Comandos para la actualización del sistema.

- El siguiente paso consiste en instalar todas las dependencias para asegurar el soporte para leer y escribir imágenes, etc. , para lo cual se utiliza la siguiente línea de comandos:

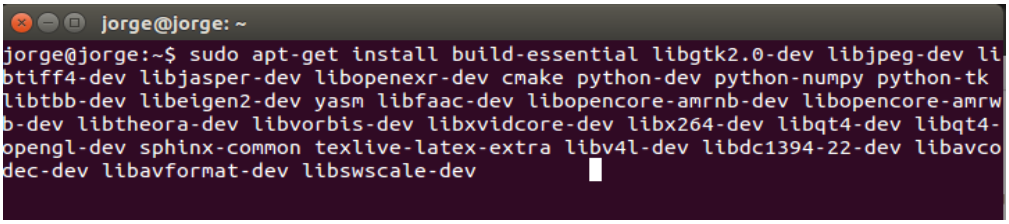


Figura 112: Instalación de las dependencias.

- Ahora debemos obtener Opencv 2.4.9, para lo cual es necesario visitar la página oficial de descargas **opencv.org/downloads** y obtendremos una carpeta como la siguiente:

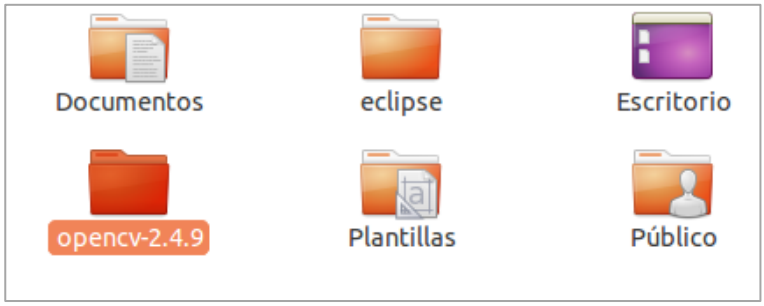


Figura 113: Descarga de Opencv 2.4.9

- A continuación se debe generar el Makefile utilizando el cmake que viene en la carpeta de Opencv 2.4.9, para lo cual primeramente debemos introducirnos en el directorio de Opencv 2.4.9 usando el comando que en la siguiente figura se muestra:

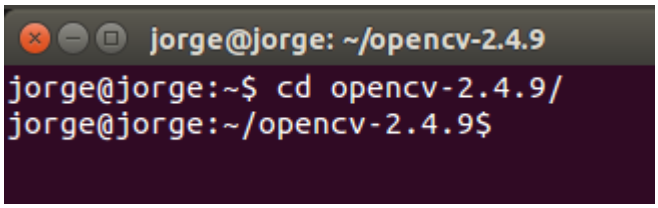


Figura 114: Cambio de directorio.

Luego de que estemos dentro de la carpeta de Opencv 2.4.9 vamos a crear la carpeta **build,** que será la que contenga todos los archivos que se obtengan luego de compilar el **cmake,** para esto usamos:

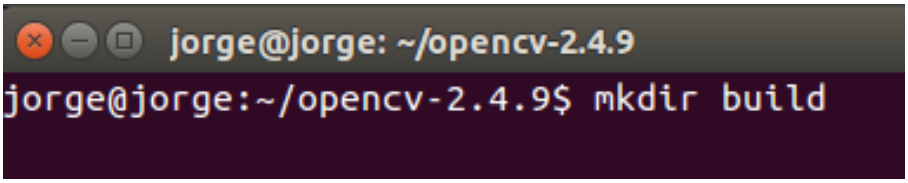


Figura 115: Creación de la carpeta build.

Seguidamente cambiamos de directorio, y nos dirigimos a la carpeta **build:**

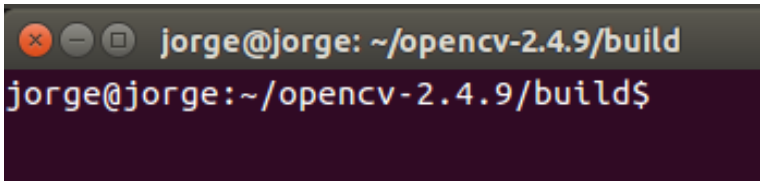


Figura 116: Cambio de directorio a build

Ahora debemos ejecutar la siguiente línea de comandos:

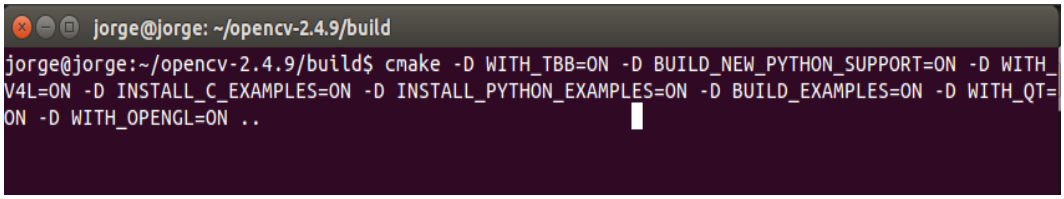


Figura 117: Generación del Makefile usando cmake.

Para comprobar que la línea de comandos anteriormente citada es correcta y no da ningún error debemos fijarnos que lo generado sea parecido a lo siguiente:

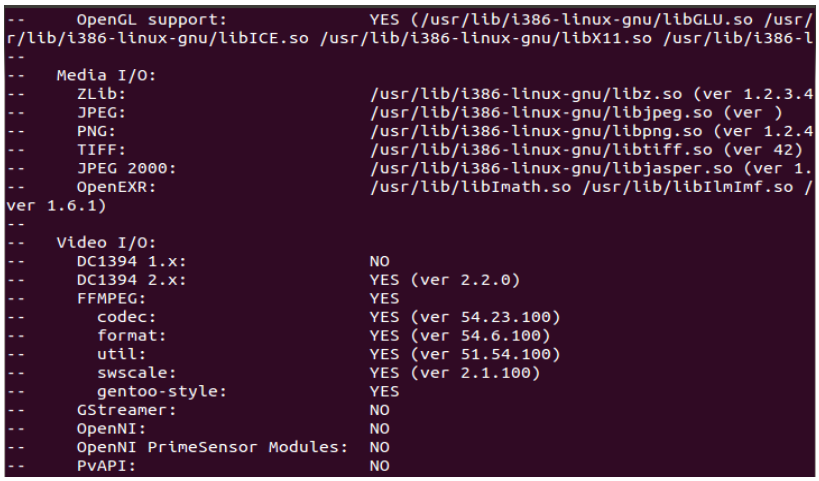


Figura 118: Verificación de la compilación del cmake.

- Luego de que se haya verificado de que no existan errores en la compilación de cmake, podremos instalar Opencv 2.4.9 mediante los siguientes comandos:

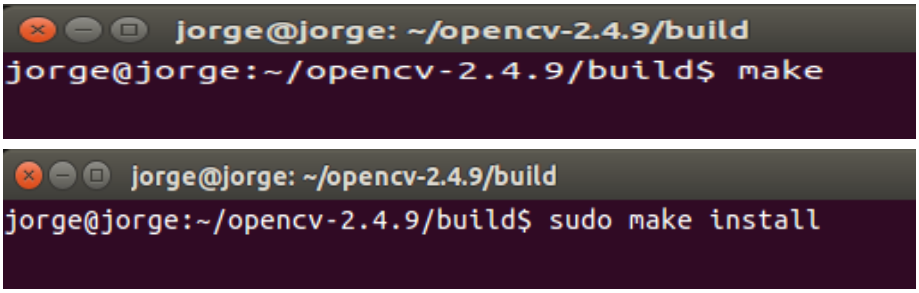


Figura 119: Proceso de instalación de Opencv 2.4.9.

- El siguiente paso consiste en configurar Opencv 2.4.9, para lo cual debemos abrir el fichero **opencv.conf** de la siguiente manera:

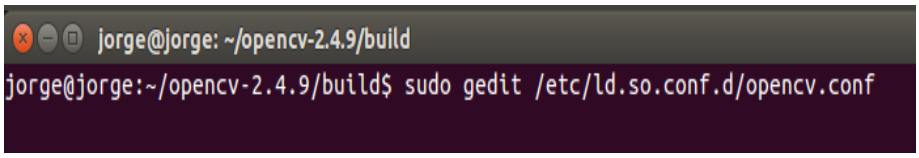


Figura 120: Proceso de configuración de Opencv 2.4.9

En el fichero que se nos abre, que por cierto estará en blanco le ponemos la siguiente línea: **/usr/local/lib** y guardamos el archivo.

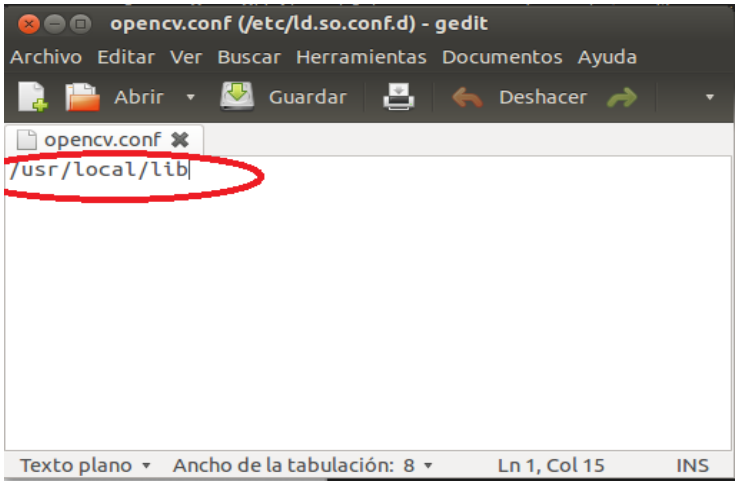


Figura 121: Configuración de Opencv 2.4.9

Seguidamente se procede a configurar las librerías:

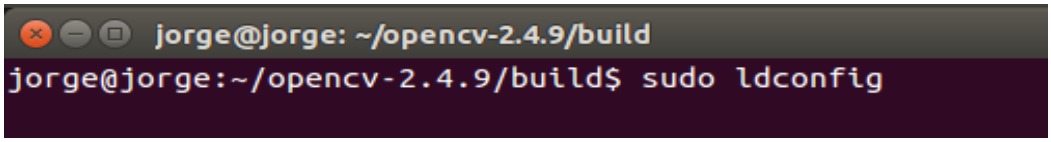


Figura 122: Configuración de las librerías.

Ahora digitamos el siguiente comando para abrir un archivo:

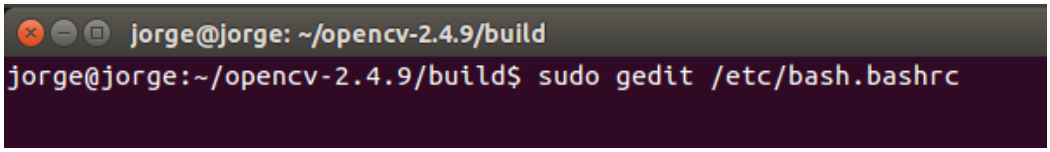


Figura 123: Comando para abrir fichero bash.bashrc.

Al final agregamos las dos siguientes líneas:
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig
export PKG_CONFIG_PATH y guardamos el archivo:

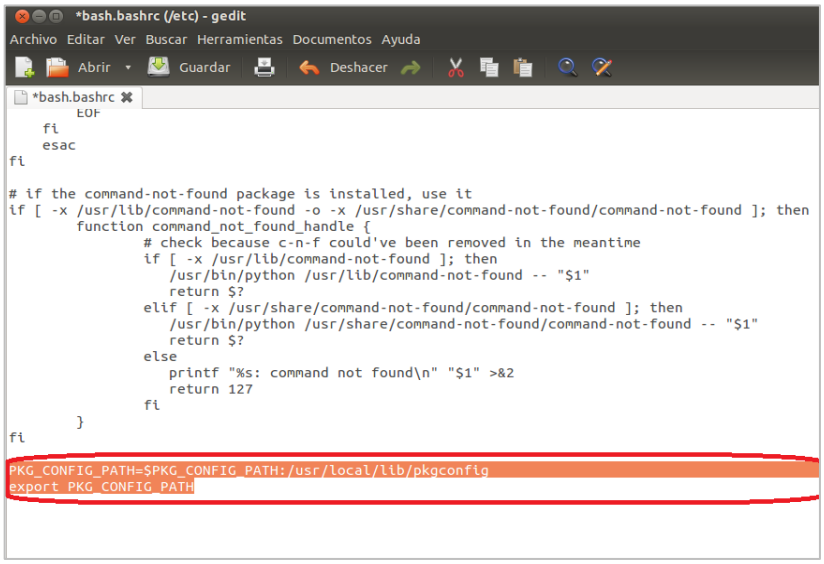


Figura 124: Configuración de las librerías de Opencv 2.4.9.

- Finalmente cerramos la consola y reiniciamos Ubuntu. Y es hora de probar los ejemplos que nos proporciona Opencv, y procedemos a construirlos de la siguiente manera:

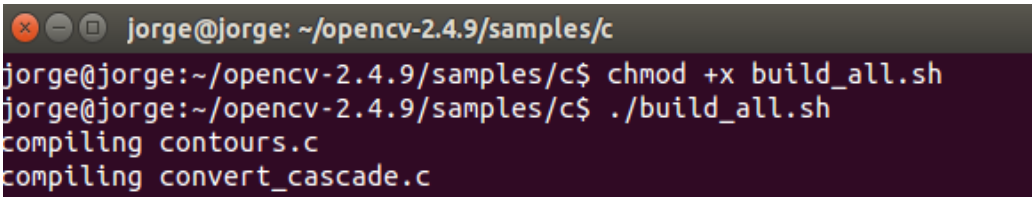


Figura 125: Proceso de construcción de ejemplos de Opencv.

Ahora ejecutamos uno de los ejemplos:

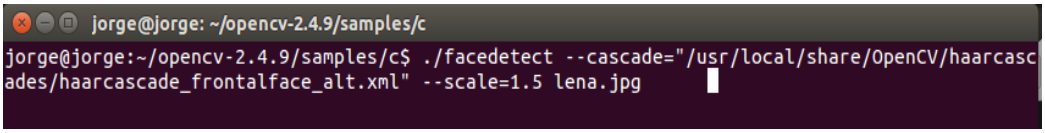


Figura 126: Proceso de construcción de ejemplos de Opencv.

El resultado es el siguiente:

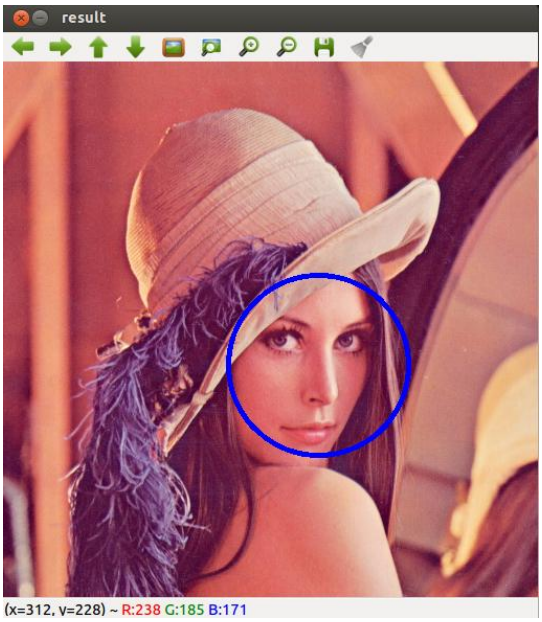


Figura 127: Resultado obtenido al ejecutar uno de los ejemplos.

Anexo 2. Instalación de Opencv en Windows

Primeramente debemos descargar el instalador que poseen las librerías de Opencv de la página oficial **www.opencv.org.**

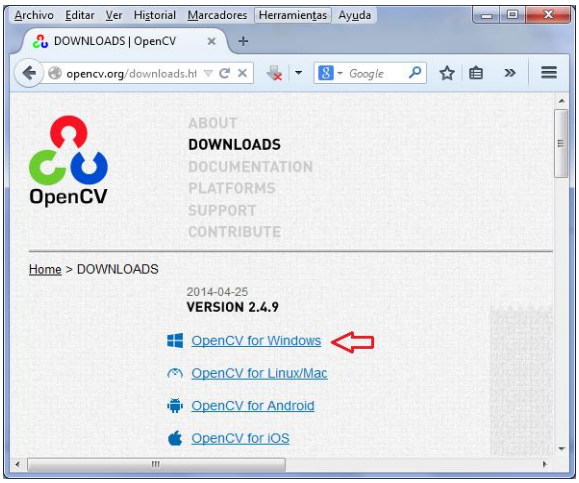


Figura 128: Descarga del instalador de Opencv

Seguidamente debemos ejecutar el instalador descargado de Opencv 2.4.9 para extraer las librerías que permitirán realizar la configuración.

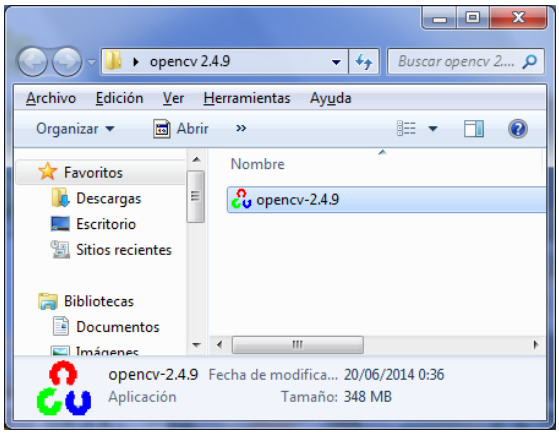


Figura 129: Instalador de opencv2.4.9

El siguiente paso consiste en asignar la ruta o dirección donde se van a descomprimir las librerías, en este caso en la unidad **C:**

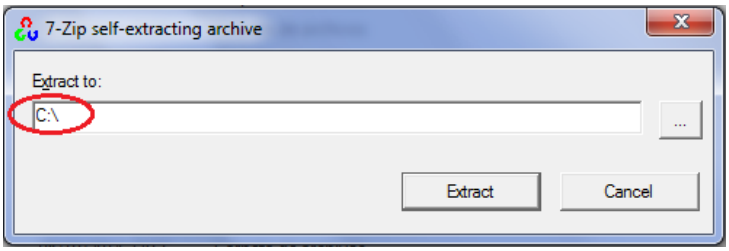


Figura 130: Ruta de extracción de las librerías de Opencv.

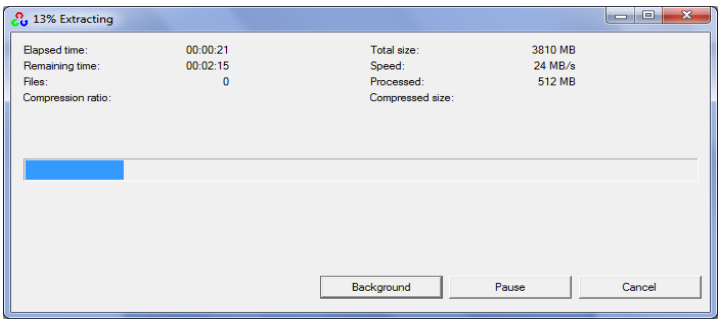


Figura 131: Proceso de extracción de las librerías.

También se debe configurar las variables de entorno (PATH) asignando la ruta de las librerías. Para esto se debe tomar en cuenta la arquitectura del computador (64bits o 32bits).

- ***Para 64bits***

C:\opencv\build\x64\vc10\bin

C:\opencv\build\x64\vc11\bin

- ***Para 32bits***

C:\opencv\build\x86\vc10\bin

C:\opencv\build\x86\vc11\bin

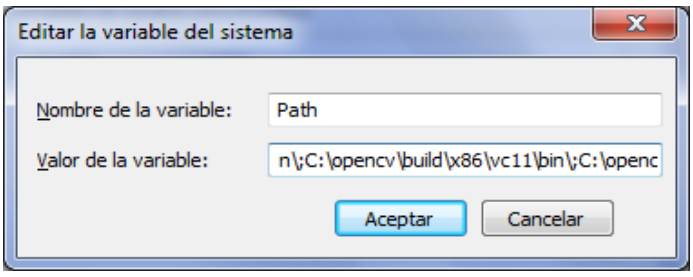


Figura 132: Configuración del PATH.

Anexo 3. Instalación del Entorno de Desarrollo Integrado (IDE) QT

Otro de los elementos necesarios para el desarrollo del Sistema fue la utilización de un IDE (Entorno de Desarrollo Integrado), y se optó por utilizar QT Creator por la facilidad de uso y la fácil integración con el Lenguaje de programación C++ y las librerías de Opencv.

Su instalación no es gran cosa, pero es importante darla a conocer, para lo cual nos dirigimos al **centro de software de Ubuntu 14.04 LTS** y en el campo de texto de búsqueda digitamos **qt creator y** le damos en instalar.

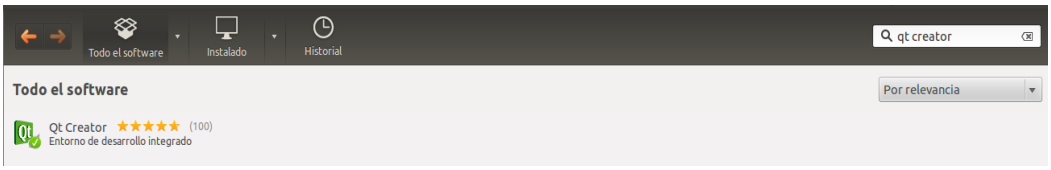


Figura 133: Instalación del IDE QT Creator en Ubuntu 14.04 LTS.

Luego de que hayamos instalado QT Creator vamos a tener una pantalla inicial como la siguiente:

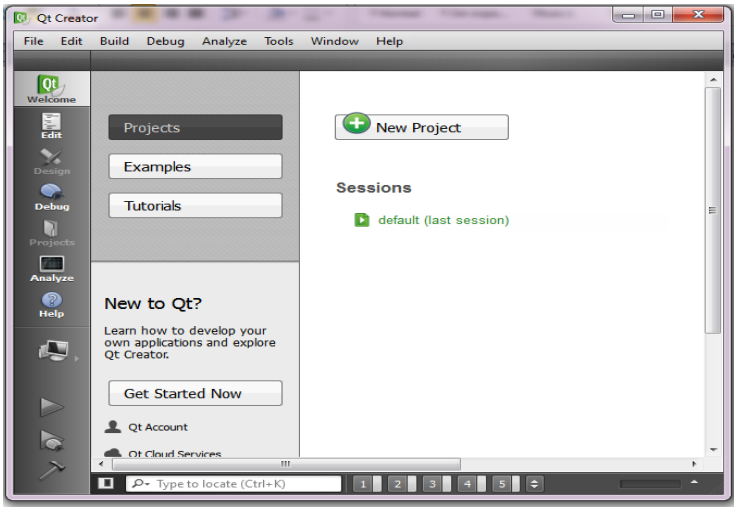


Figura 134: Pantalla principal del IDE QT Creator.

Cabe recalcar que existen otras maneras de instalar este IDE, como por ejemplo descargandolo de la página oficial de QT Creator y seleccionando la versión adecuada.

Anexo 4. Artículo Científico

Desarrollo de un sistema basado en Visión Artificial para el reconocimiento de matrículas vehiculares en el AEIRNNR de la UNL

J. Males, R. Rojas, Tutor: H. Paz

***Abstract*— This article shows how to build a system for license plate detection based on machine vision, using OpenCV libraries, the programming language C ++ and QT Integrated Development Environment Creator. To implement the system use was made of the methodology waterfall approach, because it uses steps that are common to almost any type of system. Between stages highlight is the analysis, design, implementation and verification of the system.**

***Keywords*— Artificial vision, OpenCV, programming language integrated development environment.**

I. INTRODUCCIÓN

EL sentido de la vista permite al ser humano conocer el medio que lo rodea, relacionarse con sus semejantes y contar con los elementos adecuados para captar e interpretar las señales provenientes de ellos. Según Aristóteles la "visión" es recuperar de la información de los sentidos (vista) propiedades válidas del mundo exterior. Marr opina que la "visión" es el proceso que se produce, a partir de las imágenes del mundo exterior, una descripción que es útil para el observador y que no tiene información irrelevante [1]. De hecho se calcula que más de 70% de las tareas del cerebro son empleadas en el análisis de la información visual. Las imágenes visuales obtenidas a través del ojo proporcionan información sobre el color, la forma, la distancia, posición y movimiento de los objetos. En la actualidad todas las disciplinas científicas emplean útiles gráficos para transmitir el conocimiento. Por ejemplo en el desarrollo de software los ingenieros utilizan los modelos Lenguaje de Modelo Unificado (UML) para describir la arquitectura de un sistema informático, en ingeniería electrónica se emplean esquemas de circuitos, a modo gráfico para describirlos. Las grandes empresas utilizan imágenes en alta resolución para realizar publicidad y hacer conocer sus productos. Se podría hacerlo mediante texto, pero para la especie humana resulta mucho más eficiente procesar imágenes que texto, dando una mayor interpretación al conocido refrán popular de "Una imagen vale más que mil palabras" que tiene mucho que ver con los aspectos cognitivos de la especie humana. Hoy en día, se carece de una teoría que explique cómo los humanos perciben el exterior a través de la vista.

Es así que han aparecido términos tecnológicos como visión artificial o visión por computador [2]. Uno de los grandes eventos de las últimas décadas fue la invasión de los medios digitales dentro de todos los aspectos de la vida cotidiana, la computadora ha tomado una gran importancia en el procesamiento de datos, entre ellos las imágenes.

II. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DEL SISTEMA

Actualmente existen una variedad de herramientas disponibles en el mercado de manera comercial y de libre distribución y uso, que facilitan el desarrollo de sistemas basados en Visión Artificial. Al momento de desarrollar aplicaciones de este tipo es muy importante saber elegir una librería como herramienta de desarrollo, aunque no sería muy recomendable utilizar solo una. Es probable que las deficiencias de una puedan ser resueltas por otra, o que para un problema concreto resulte aconsejable usar una librería específica como es el caso de Opencv [3].

Para el desarrollo del Sistema se necesita tener conocimientos de programación, en este caso conocer sobre el lenguaje de programación C++ [4], además tener clara la idea de lo que es un Entorno de Desarrollo(IDE) como QT Creator y saber cómo utilizar las librerías que nos proporciona Opencv en lo que respecta la Visión Artificial. Además conocer sobre el gestor de bases de datos MYSQL.

El lenguaje de programación C++ con la ayuda de las librerías de Opencv permiten realizar excelentes trabajos relacionados con la Visión Artificial, motivo por el cual es muy utilizado en la actualidad para el reconocimiento de objetos en tiempo real, aunque también existe otra herramienta muy similar que es el Toolbox de Matlab, pero a diferencia de Opencv es mucho más lenta al momento de trabajar en tiempo real, razón principal por la cual no se emplea Matlab en lo que respecta a Visión Artificial.

Cabe recalcar que para obtener buenos resultados en el desarrollo de Sistemas de este tipo, referentes al tiempo de ejecución y procesamiento es muy importante, o mejor dicho necesario en muchos casos utilizar diferentes herramientas, lenguajes de programación, para luego integrarlos y hacerlos trabajar en conjunto y así lograr mayores resultados, pero siempre y cuando se los aplique de la manera correcta.

J. Males, Universidad Nacional de Loja, Loja, Ecuador, jamalesc@unl.edu.ec
R. Rojas, Universidad Nacional de Loja, Loja, Ecuador, rfrojasl@unl.edu.ec
H. Paz, Universidad Nacional de Loja, Loja, Ecuador, hpaz@unl.edu.ec

En la Fig. 1 se muestra la apariencia que tiene el entorno de desarrollo integrado QT Creator [5], empleado para desarrollar el sistema para reconocimiento de placas vehiculares, en la cual también se presenta una ventana de loggin para el acceso al sistema empleando una base de datos con la ayuda de MYSQL.

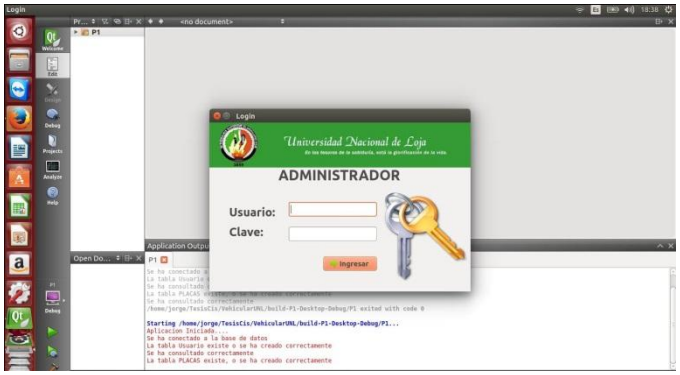


Figura 1. Ejecución del Sistema en el Entorno de Desarrollo Integrado (IDE) QT Creator.

La configuración de las librerías de Opencv se muestra en la Fig. 2, para lo cual se debe tener previamente instalado Opencv, en este caso en el Sistema Operativo Ubuntu 14.04 LTS [6]. Cuando se crea un proyecto en QT Creator se crea un archivo .pro en el cual debemos establecer el directorio de las librerías de Opencv.

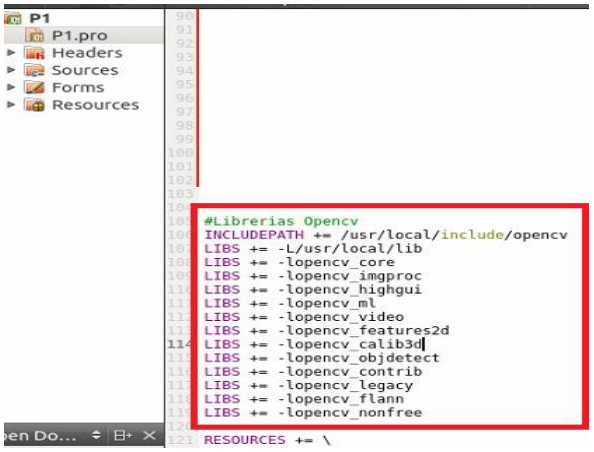


Figura 2. Configuración de las librerías de Opencv en el archivo .pro del IDE QT Creator.

Para poder utilizar el gestor de bases de datos MYSQL hay que realizar la instalación en la plataforma Ubuntu 14.04 (que es la que se usó para desarrollar el sistema).

III. IMPLEMENTACIÓN DEL SISTEMA.

Como ya se mencionó de manera anticipada que para el desarrollo del sistema se utilizó la metodología con enfoque en cascada [7], debido a que es común para casi cualquier tipo de sistema, a continuación se detallan cada una de las etapas correspondientes a la metodología:

a) Análisis de requerimientos: Al tratarse de un sistema para la detección de placas vehiculares en el AEIRNNR de la UNL, en esta fase se tuvo que realizar el análisis acerca de cómo se llevan los procesos para realizar el control y registro de los vehículos pertenecientes a la institución, para lo cual se tiene que recolectar información, y de esta manera poder determinar los requerimientos del Sistema. Esto fue posible gracias a la ayuda de la técnica de observación directa y la entrevista.

Entre los requerimientos funcionales y no funcionales están los siguientes:

- ✓ **Requerimientos funcionales**
  - Utilizar una cámara para realizar la detección de objetos.
  - Realizar la detección de placas y la detección de vehículos.
  - Mostrar fotos de las placas detectadas.
  - Almacenar en una base de datos las placas detectadas.
- ✓ **Requerimientos no funcionales**
  - Poseer una interfaz amigable para el usuario.
  - Facilitar su uso de manera apropiada.

b) Diseño del sistema: Luego de haber realizado el análisis y determinación de los requerimientos para el desarrollo del Sistema, el siguiente paso es el diseño del mismo, para lo cual se ha desarrollado un algoritmo, que a continuación se detalla.

1. Algoritmo para la detección de matrículas vehiculares. En la Fig. 3 se muestra el algoritmo que permite realizar: primeramente la detección de los vehículos en tiempo real, y luego la detección de las matrículas de los vehículos.

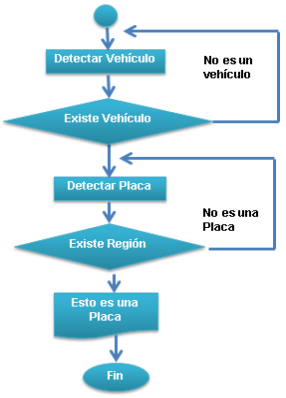


Figura 3. Diagrama de flujo del algoritmo empleado para la detección de matrículas vehiculares.

En esta fase hay algo muy interesante que recalcar, que tanto para la detección de vehículos como matrículas se empleó una técnica muy conocida en la actualidad, que consiste en construir un clasificador en cascada. Para construir este clasificador se utilizan muestras de objetos a detectar, es decir, se necesitan imágenes que contengan el objeto a detectar, para este caso se utilizaron 2000 imágenes positivas que contengan vehículos y 5000 imágenes negativas para poder diferenciar el objeto, esto en cuanto a la detección de vehículos. Y para la detección de matrículas se emplearon

como muestras 1000 imágenes positivas y 2000 imágenes negativas.

2. Reconocimiento Óptico de caracteres (OCR)

El Reconocimiento Óptico de caracteres (OCR) tiene como objetivo recuperar los datos de la matrícula, es decir, obtener los números y letras de las matrículas. Para cada matrícula detectada, se procede a segmentar cada uno de los números y letras de la misma. En la **Figura 4** se puede apreciar el diagrama de flujo del OCR.

Figura 4. Diagrama de flujo del algoritmo empleado para el Reconocimiento Óptico de Caracteres.

2. Modelo del dominio del Sistema

Gracias a la ayuda del conjunto de entidades que se emplearon para desarrollar el Sistema se realizó el modelo del dominio, el cual se muestra en la **Figura 5.**

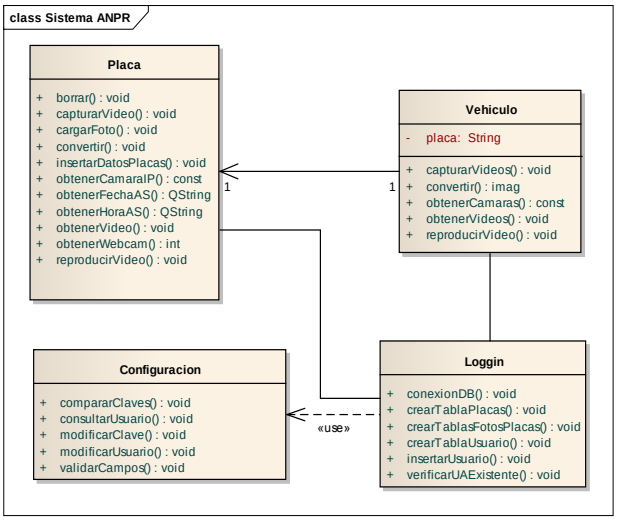


Figura 5. Diagrama de flujo del algoritmo empleado para el Reconocimiento Óptico de Caracteres.

c) Implementación del sistema: En esta fase se debe realizar la codificación del sistema, para lo cual se utilizan los algoritmos correspondientes y los clasificadores que se construyeron para realizar la detección de vehículos y placas vehiculares. Estos clasificadores (archivos) tienen la extensión .xml. En la Fig. 6 se pude apreciar el código fuente, donde se realiza la implementación del algoritmo para la detección de vehículos y en la Fig. 7 para la detección de matrículas vehiculares.

```
VideoCapture c(0);
Mat s;
Mat a;
void placas::capturar(){

    CascadeClassifier clasificador;
    QString resu;
    c.read(a);
    clasificador.load("/home/jorge/TESISCIS/CTVehiculo/ControlVehicular/clasificadores/output.xml");
    vector<Rect> reconocidos;
    clasificador.detectMultiScale(a,reconocidos,1.2,3,0,cvSize(20,20));
    for (size_t var = 0; var < reconocidos.size(); ++var) {
       rectangle(a,Point(reconocidos[var].x,reconocidos[var].y), Point(reconocidos[var].x+reconocidos[var].width,reconocidos[var].y+ reconocidos[var].height),cvScalar(0,255,0));
       s= a(reconocidos[var]);
       imshow("Vehiculo Detectado",s);
```

Figura 6. Implementación del algoritmo para la detección de vehículos.

```
VideoCapture c(0);
Mat s;
Mat a;
void placas::capturar(){

    CascadeClassifier clasificador;
    QString resu;
    c.read(a);
    clasificador.load("/home/jorge/TESISCIS/CTVehiculo/ControlVehicular/clasificadores/placas.xml");
    vector<Rect> reconocidos;
    clasificador.detectMultiScale(a,reconocidos,1.2,3,0,cvSize(20,20));
    for (size_t var = 0; var < reconocidos.size(); ++var) {
       rectangle(a,Point(reconocidos[var].x,reconocidos[var].y), Point(reconocidos[var].x+reconocidos[var].width,reconocidos[var].y+ reconocidos[var].height),cvScalar(0,255,0));
       s= a(reconocidos[var]);
       imshow("Placa Detectada",s);
```

Figura 7. Implementación del algoritmo para la detección de matrículas vehiculares.

d) Resultados: Luego de haber concluido la parte del diseño e implementación del Sistema, la siguiente parte corresponde a la verificación del mismo, obteniéndose los siguientes resultados:

1. Detección de vehículos. Es evidente la potencialidad que poseen las librerías Opencv al momento de desarrollar sistemas basados en Visión Artificial y más aún cuando se tratan de sistemas en tiempo real, y también haciendo uso de los clasificadores en cascada para realizar la detección de objetos, que en este caso los objetos a detectar fueron los vehículos y las placas. Cabe destacar que el clasificador en cascada .xml es la pieza fundamental para realizar la detección de vehículos, como se puede evidenciar en la Fig. 6 no es mucho el código fuente empleado, el mayor trabajo fue construir dicho clasificador, lo cual permitió obtener grandes resultados frente a otras técnicas aplicadas para la detección de objetos dentro de la Visión Artificial. Para ser exactos en términos estadísticos podemos decir que se logró realizar entre un 90 y 95% la detección de vehículos en tiempo real, pero con la desventaja de que al existir mucha luz se alteran los resultados en lo referente a la detección. Para lo cual es conveniente tener muy en cuenta este tipo de casos que se suscitan cuando se desarrolla un Sistema de este tipo.

En la Fig. 8 se muestra la ventana principal del sistema, en la cual al seleccionar la opción control vehicular vamos a poder realizar la detección de vehículos en tiempo real, tal y como se muestra en la Fig. 9, encerrando cada vehículo detectado en un recuadro.

Figura 8. Pantalla principal del sistema.

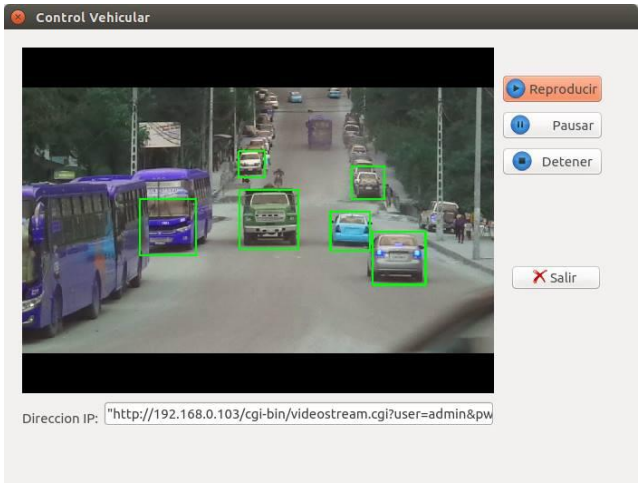


Figura 9. Detección de vehículos en tiempo real.

En la Fig. 10 también podemos ver como el sistema realiza la detección de vehículos ubicando la cámara a una altura considerable de 2 a 3 metros en el AEIRNNR de la UNL.

Figura 10. Detección de vehículos en tiempo real en el AEIRNNR de la UNL.

2. Detección de matrículas vehiculares: Con la ayuda del clasificador en cascada se logró realizar una excelente detección de matrículas vehiculares, básicamente el algoritmo empleado es el mismo que se empleó para la detección de vehículos, lo que cambia es el clasificador.

La detección de placas vehiculares corresponde a un 90% y 96%, siendo este un gran resultado frente a otras técnicas de detección de objetos que la mayoría de veces lo hacen con imágenes y no en tiempo real.

En la Fig. 11 podemos ver como realiza el sistema la detección de matrículas vehiculares en tiempo real, encerrando cada placa detectada en un recuadro y presentando la foto de la misma, para posteriormente ser almacenada en una base de datos, con la fecha y hora en la que fue detectada.

Además, luego de que se realiza la captura de la matrícula vehicular, esta es procesada para realizar el reconocimiento óptico de caracteres, gracias a la ayuda de una herramienta libre denominada **tessceract.**

En la Fig. 12 podemos ver como se realiza el reconocimiento óptico de caracteres correspondientes a las matrículas vehiculares.

Figura 11. Detección de matrículas vehiculares en tiempo real.

Figura 12. Reconocimiento Óptico de matrículas vehiculares.

Luego de ser detectadas las placas (ver Fig. 11) las matrículas son capturadas y almacenadas en una base de datos (MYSQL), para luego ser presentadas a través de reportes, para lo cual se utilizó la herramienta NCReport. En la Fig. 13 se puede apreciar una muestra de cómo genera el sistema los reportes.

Figura 13. Generación de reportes sobre las placas vehiculares capturadas.

De igual manera también se almacenan los caracteres de las matrículas que son reconocidas por el OCR. En la Fig. 14 se pueden ver los reportes.

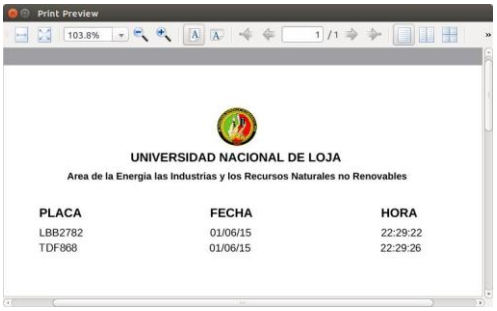


Figura 14. Generación de reportes sobre las placas vehiculares reconocidas.

IV. MÉTODOS UTILIZADOS.

Para el desarrollo del Sistema es necesario seguir un esquema metodológico, el cual está basado en la utilización de diferentes métodos y técnicas que facilitan la recolección de datos, y que se utiliza para estructurar, planificar y controlar un proyecto de desarrollo y es de gran utilidad para obtener grandes posibilidades de éxito en el desarrollo de Sistemas.

1. Método científico. Este método permite obtener los conocimientos necesarios sobre la Visión Artificial, y que además ayuda a plantear soluciones mediante procedimientos, y de alguna manera a comprender ciertos conceptos que se utilizaron en el desarrollo del sistema.

2. Método inductivo. Como este tipo de Sistemas puede ser implantado en instituciones públicas o privadas con el fin de realizar un mejor registro y control de vehículos, este método permitió verificar como iba a repercutir dentro de la institución el manejo de un Sistema para el control y registro de placas vehiculares, y a través de esto se pudo establecer conclusiones sobre las futuras mejoras que tendrá la institución.

3. Método deductivo. Este método permite partir de los aspectos generales entre temas como es el desarrollo de un Sistema basado en Visión Artificial, obteniendo resultados relacionados con la detección de placas vehiculares.

4. Método descriptivo. Este método ayuda a describir paso a paso el proceso de desarrollo del sistema.

V. TRABAJOS RELACIONADOS.

Los Sistemas de computadoras cada vez son más potentes y por ende menos costoso, lo que permite crear nuevas formas de arte que antes no era posible, y algunas otras formas de arte antiguas pueden ahora verse beneficiadas con novedosas técnicas asistidas por computadora [8].

El reconocimiento de imágenes ha evolucionado a medida que mejora la tecnología. Puede encontrarse en numerosos campos. Actualmente con la ayuda de la Visión Artificial se han logrado implementar un gran número de herramientas que han sido de gran utilidad dentro del campo de la industria y fuera de ella. A continuación se presentan algunos de los trabajos relacionados con este tipo de Sistemas, descrito en el presente artículo técnico.

1. Identificación de personas para investigaciones policiacas. Aunque las técnicas aún están en desarrollo en este campo, y aún no existe una aplicación totalmente confiable, es evidente la importancia del reconocimiento de imágenes para la identificación de personas en investigaciones policiacas. Muchas veces en investigaciones de crímenes un testigo puede describir con mucho detalle el rostro de un criminal. Un dibujante profesional convierte la descripción verbal del testigo en un dibujo sobre papel. El trabajo de la computadora consiste en buscar el rostro del criminal en una base de datos de imágenes. En las investigaciones policíacas también se utiliza la búsqueda de huellas dactilares en una base de datos [8].

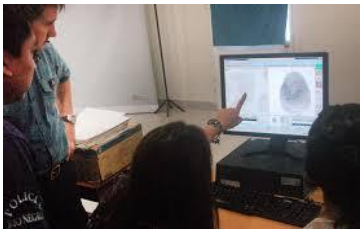


Figura 15. Simulacro de identificación de personas.

2. Información inteligente sobre tráfico vehicular. Actualmente se están desarrollando los denominados semáforos inteligentes que permiten tomar decisiones dependiendo de una serie de parámetros de entrada, como el flujo de vehículos, velocidad media, entre otros, para tratar de evitar el congestionamiento vehicular, tiempos excesivos de viajes, esperas innecesarias, mayor contaminación en el medio ambiente [8].

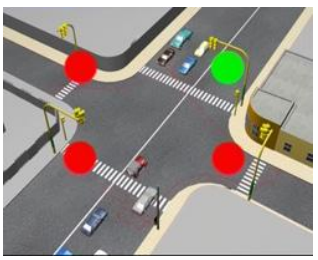


Figura 16. Simulacro de semáforos inteligentes.

3. Biometría. Es el reconocimiento del cuerpo humano a través de ciertas características físicas, como el tamaño de los dedos de la mano, las huellas dactilares o los patrones en las retinas de los ojos. Los sistemas de computadoras actuales permiten tener mejores niveles de seguridad utilizando la biometría. Por ejemplo, una persona puede tener acceso a un área restringida, por medio del reconocimiento de las características físicas de su mano en un dispositivo especial. Si en el proceso de validación se verifica que la persona tiene permiso para entrar al área, entonces le permitirá el acceso. Este tipo de sistemas se está volviendo cada vez más utilizado, desplazando los sistemas antiguos de identificación [8].

Figura 17. Registro e identificación de personas a través de la Biometría.

VI. CONCLUSIONES.

Luego de haber concluido con el desarrollo del presente artículo técnico, es importante puntualizar algunas conclusiones, las mismas que a continuación se detallan:

El uso de clasificadores en cascada ha permitido realizar la detección de vehículos y de sus respectivas matrículas, con cierto margen de error en algunos casos debido a que no se asumieron ciertas restricciones.

La metodología con enfoque en cascada es un buen camino a seguir cuando se trata de desarrollar cualquier tipo de sistemas, especialmente si están relacionados con la Visión Artificial, debido a que contempla etapas o fases que son comunes para cualquier sistema. Las librerías de Opencv representan una ayuda excepcional dentro del campo de la Visión Artificial, debido a que existe gran cantidad de información y la posibilidad de integrarse con varios lenguajes de programación como C++ y entornos de desarrollo como QT Creator y el gestor de base de datos MYSQL. Para la creación de los clasificadores se utilizaron tanto imágenes negativas como positivas, para de esta manera poder diferenciar si existe o no el objeto a detectar, motivo por el cuál mientras más imágenes se tengan mejores serán los resultados que se obtengan.

VII. TRABAJOS FUTUROS.

El desarrollo de este tipo de sistemas da lugar a nuevas líneas de investigación, sean por ejemplo la mejora del mismo o sirviendo de base para el desarrollo de otros Sistemas mucho más robustos. Siendo algunos de los trabajos futuros los siguientes:

- Desarrollo de un sistema con Visión Artificial para el cobro de peajes a través del reconocimiento de placas vehiculares.
- La implementación de este tipo de sistemas dentro de las gasolineras y para los parqueaderos de vehículos.
- Sistema base para un macro proyecto como el de Semáforos inteligentes.
- De ser posible también se podría migrar la aplicación a dispositivos móviles, por ejemplo a aquellos que usan Android como Sistema Operativo.

REFERENCIAS

[1] Graficación Por Computadora: 5.2 VISION POR COMPUTADORA. [En línea]. Disponible en: http://graficacionporcomputadora.blogspot.com/2013/05/52-vision-por-computadora.html.

[2] Visión Artificial - Máster de Ingeniería de la Producción. [En línea]. Disponible en: http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/indexMIP_VA.htm.

[3] Bradski, G. Kaebler, A. Learning Opencv Computer Vision with the OpenCV Library. [En línea]. Disponible en: http://www.cs.haifa.ac.il/~dkeren/ip/OReilly-LearningOpenCV.pdf.

[4] García de Jalón, J. Rodríguez, J. I. Sarriegui, J. M. Brazález, A. (1998). Aprenda C++ como si estuviera en primero. [En línea]. Disponible en: http://mat21.etsii.upm.es/ayudainf/aprendainf/Cpp/manualcpp.pdf.

[5] Qt Creator, desarrollando aplicaciones rápidamente. | Geeks & Linux Atelier! [En línea]. Disponible en: http://glatelier.org/2009/05/15/qt-creator-desarrollando-aplicaciones-rapidamente/.

[6] Instalar OpenCV 2.4.2 en Ubuntu 12.04 LTS | desarrollophpsenior. [En línea]. Disponible en: http://desarrollophpsenior.wordpress.com/2012/09/19/instalar-opencv-2-4-2-en-ubuntu-12-04-lts/.

[7] Soloriio, M. (2013).Metodología en cascada. [En línea]. Disponible en: http://metodologiaencascada.blogspot.com/.

[8] Lara Rodríguez, Gustavo Adolfo. Técnicas de reconocimiento de imágenes para la creación de fotomosaicos. Guatemala. 2003. [En línea]. http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r25669.PDF

Jorge Males Nació en Quito el 25 de febrero de 1991. Sus estudios primarios los realizó en la escuela José Toro y Guzmán de la provincia de Loja, cantón Saraguro y sus estudios secundarios los cumplió en el Instituto Tecnológico Superior Celina Vivar Espinosa"del mismo cantón. Actualmente trabaja como técnico de sistemas del GAD Municipal del cantón Yacuambi provincia de Zamora Chinchipe.

Ronald Rojas Nació en la ciudad de Yanzatza. Sus estudios primarios los realizó en la escuela "General Rumiñahui" de la misma ciudad. Sus estudios secundarios los realizó en el Instituto "Primero de Mayo" de Yanzatza. Ha participado en proyectos de desarrollo de software en los lenguajes Java, Matlab, HTML, entre otros.

Anexo 5: Certificación Sumary

Yacuambi, 09 de Junio del 2015

Lic. Pastora Isabel Sarango Lozano
DOCENTE DE LA ASIGNATURA DE IDIOMA INGLES

A petición verbal de la parte interesada:

CERTIFICA:

Haber revisado el resumen en ingles de la tesis de los egresados Jorge Anibal Males Chalàn y Ronald Fabricio Rojas Livisaca cuyo tema es ***"Desarrollo de un sistema basado en Visión Artificial para el reconocimiento de matrículas vehiculares en el AEIRNNR de la Universidad Nacional de Loja"***

Es todo cuanto puedo certificar facultando así a los interesados hacer uso del presente certificado en lo que estimen conveniente.

Atentamente;

Lic. Isabel Sarango
DOCENTE DE INGLES