

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**TEMA:
DESARROLLO DE UN SISTEMA DE INSPECCIÓN AUTOMÁTICO DE
PCB'S MEDIANTE VISIÓN ARTIFICIAL**

**AUTORES:
WILSON RENE AYO MASABANDA
HILDA MARIZOL MORENO ANDRADE**

**TUTOR:
HAMILTON LEONARDO NÚÑEZ VERDEZOTO**

Quito, marzo del 2021

DECLARACIÓN DE DERECHOS DE AUTOR

Nosotros, Wilson Rene Ayo Masabanda, con documento de identificación N° 172266290-3 e Hilda Marizol Moreno Andrade con documento de identificación N° 060363840-4, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: DESARROLLO DE UN SISTEMA DE INSPECCIÓN AUTOMÁTICO DE PCB'S MEDIANTE VISIÓN ARTIFICIAL, mismo que ha sido desarrollado para optar por el título de: INGENIERO ELECTRÓNICO, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Wilson Rene Ayo Masabanda
Cédula: 1722662903



Hilda Marizol Moreno Andrade
Cédula: 0603638404

Quito, marzo 2021

DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, DESARROLLO DE UN SISTEMA DE INSPECCIÓN AUTOMÁTICO DE PCB'S MEDIANTE VISIÓN ARTIFICIAL, realizado por Wilson Rene Ayo Masabanda e Hilda Marizol Moreno Andrade, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, marzo 2021

A handwritten signature in blue ink, appearing to be 'H. Verdezoto', written in a cursive style.

Hamilton Leonardo Núñez Verdezoto
Cédula de identidad: 1712981487

DEDICATORIA

Este proyecto de titulación es el esfuerzo y la perseverancia en el proceso de aprendizaje para cumplir un sueño y alcanzar una meta más en mi vida, junto a mi compañera de tesis, docentes que compartieron sus enseñanzas para formar un profesional de excelencia con sólidos conocimientos y valores para afrontar los retos de la vida.

Este proyecto lo dedico a mis padres, Wilson y Gladys por brindarme su apoyo incondicional, estar siempre conmigo e inculcarme el valor de la responsabilidad, esfuerzo y perseverancia en cada paso que doy para ser cada vez una mejor persona tanto en lo espiritual como profesional.

Wilson

Este proyecto le dedico a mis padres Leonor y Francisco que, con su amor, y dedicación han logrado impulsarme y me dan fortaleza para seguir adelante.

A mi querido amigo Alejandro por apoyarme en todo momento.

Hilda

AGRADECIMIENTO

Mi más sincero agradecimiento a Hilda por ser parte de este proyecto, por brindarme su confianza y darme todo el apoyo necesario para que esta meta se cumpla, a todos los docentes que fueron parte de las enseñanzas tanto éticas, como profesionales y compartir su conocimiento conmigo, al Ingeniero Hamilton Núñez por ser la guía para culminar con éxito este trabajo de titulación.

Agradezco también a mis padres por apoyarme siempre en todo momento y a mis hermanos que estuvieron siempre conmigo.

Wilson

Agradezco a todas las personas que son parte de mi vida, tanto en lo profesional, espiritual y familiar.

A esos pocos, pero fieles amigos, mi agradecimiento infinito por el apoyo aquellas personas que me han brindado una mano en los momentos difíciles, los llevo en mi corazón.

A mis hermanos que son los pilares en donde me sostengo, y son también mi motivación.

A mi compañero de tesis Wilson y nuestro tutor Ing. Hamilton Núñez por ayudarnos y guiarnos en este camino difícil pero reconfortante.

Hilda

ÍNDICE DE CONTENIDO

DECLARACIÓN DE DERECHOS DE AUTOR.....	ii
DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
RESUMEN.....	xiii
ABSTRACT.....	xiv
INTRODUCCIÓN	xv
CAPÍTULO 1	1
ANTECEDENTES.....	1
1.1 PROBLEMA DE ESTUDIO.....	1
1.2 JUSTIFICACIÓN	1
1.3 OBJETIVOS	2
1.3.1 Objetivo General	2
1.3.2 Objetivos Específicos.....	2
1.4 METODOLOGÍA	2
CAPÍTULO 2	3
FUNDAMENTACIÓN TEÓRICA.....	3
2.1 CONTROL DE CALIDAD	3
2.2 PRODUCTIVIDAD.....	3
2.2.1 Recursos Humanos.....	3
2.2.2 Maquinaria y equipo	4
2.2.3 Organización del trabajo	4
2.3 VISIÓN ARTIFICIAL.....	4
2.4 PROCESAMIENTO DIGITAL DE IMÁGENES.....	4
2.4.1 Imagen.....	5
2.5 FILTRADO DE UNA IMAGEN	5

2.5.1 Filtrado en el dominio de la frecuencia.....	6
2.5.2 Filtrado en el dominio del espacio	7
2.6 FABRICACIÓN DE PLACAS DE CIRCUITO IMPRESO PRINTED CIRCUIT BOARD (PCB).....	8
2.7 MÉTODOS DE FABRICACIÓN DE PCBs	8
2.7.1 Impresión serigráfica.....	8
2.7.2 Fotograbado	9
2.7.3 Fresado de circuitos impresos	9
2.7.4 Impresión en material termosensible	9
2.8 TIPO DE ERRORES EN PCBs	9
2.9 INSPECCIÓN AUTOMÁTICA DE DEFECTOS.....	10
2.10 APLICACIONES DE INSPECCIÓN DE PCB POR VISIÓN ARTIFICIAL	10
2.10.1 Aplicación 1	10
2.10.2 Aplicación 2	12
2.10.3 Aplicación 3	13
2.10.4 Aplicación 4	15
2.10.5 Aplicación 5	16
CAPÍTULO 3	18
DISEÑO E IMPLEMENTACIÓN.....	18
3.1 DIAGRAMA GENERAL DE FUNCIONAMIENTO	18
3.2 DESCRIPCIÓN HARDWARE	19
3.2.1 Escáner	19
3.2.2 PC.....	20
3.2.3 Arduino Uno	21
3.2.4 Motor a pasos	21
3.2.5 Módulo A4988	22
3.2.6 Fuente de alimentación	22

3.3 ESTRUCTURA METÁLICA Y COMPONENTES	23
3.3.1 Acondicionamiento escáner	23
3.3.2 Sistema de desplazamiento	24
3.3.3 Diagrama de bloques del control del motor a pasos	26
3.3.4 Sistema de Iluminación	26
3.3.5 Diagrama cableado eléctrico	28
3.3.6 Hardware sistema de inspección por visión artificial	28
3.4 DESCRIPCIÓN DEL SOFTWARE	29
3.5 DIAGRAMA DE BLOQUES	29
3.6 ADQUISICIÓN DE LA IMAGEN PCB	30
3.7 PREPROCESAMIENTO DIGITAL IMAGEN PCB	31
3.7.1 Importación módulo cv2	31
3.7.2 Lectura de la imagen PCB.....	32
3.7.3 Conversión a escala de grises.....	32
3.7.4 Transformación a imagen binaria.....	32
3.7.5 Atenuación del ruido	33
3.8 SELECCIÓN IMAGEN REFERENCIA	33
3.9 ALINEACIÓN DE IMÁGENES	34
3.10 ALGORITMO DETECCIÓN DE FALLOS EN PCB.....	36
3.11 RECONOCIMIENTO DE ERRORES	38
3.11.1 Daño en el pad.....	38
3.11.2 Espurio	41
3.11.3 Circuito abierto y cortocircuito	43
3.12 INTERFAZ GRÁFICA.....	46
3.12.1 Implementación.....	47
CAPÍTULO 4	49
PRUEBAS Y FUNCIONAMIENTO.....	49

4.1 FABRICACIÓN PLACAS PCB.....	49
4.2 PRUEBA DE INSPECCIÓN AUTOMÁTICA POR VISIÓN ARTIFICIAL. 50	
4.3 PRUEBA DE INSPECCIÓN POR VISIÓN HUMANA.....	53
4.4 RESULTADOS.....	54
CAPÍTULO 5.....	58
5.1 CONCLUSIONES.....	58
5.2 RECOMENDACIONES.....	59
REFERENCIAS.....	60

ÍNDICE DE FIGURAS

Figura 2.1	Píxeles de una porción de la imagen.....	5
Figura 2.2	Filtrado de ruido al 75% de píxeles ruidosos.....	6
Figura 2.3	Detección de los bordes de imagen.	6
Figura 2.4	Filtros en el dominio del espacio.	7
Figura 2.5	Capas de la placa para fabricación de PCBs.....	8
Figura 2.6	Tipos de errores en PCBs.	10
Figura 2.7	Diagrama de bloques aplicación 1.....	11
Figura 2.8	GUI del sistema de inspección de PCBs por visión artificial.....	11
Figura 2.9	(a) Segmentación de área, (b) Segmentación de borde.....	12
Figura 2.10	Hardware del sistema de inspección óptica automatizada.....	12
Figura 2.11	Diagrama de flujo de la aplicación 2.	13
Figura 2.12	Diagrama de flujo de la aplicación 3.	14
Figura 2.13	GUI de la aplicación 3.	15
Figura 2.14	Diagrama de bloques de la aplicación 4.	15
Figura 2.15	Detección automática de defectos en PCB.	16
Figura 2.16	Diagrama de flujo de la aplicación 5.	17
Figura 2.17	Detección de defectos de PCB mediante procesamiento de imágenes...	17
Figura 3.1	Diagrama general de funcionamiento.....	18
Figura 3.2	Hardware sistema de inspección por visión artificial.	19
Figura 3.3	Estructura armario metálico.....	23
Figura 3.4	Ubicación y acondicionamiento del escáner.....	24
Figura 3.5	Acople del motor al sistema deslizante.	25
Figura 3.6	Obertura armario metálico.....	25
Figura 3.7	Mecanismo de soporte del PCB.....	26
Figura 3.8	Diagrama de bloques del control del motor a pasos.	26
Figura 3.9	Método colocación y especificaciones flujo luminaria.	27
Figura 3.10	Sistema de iluminación.....	27
Figura 3.11	Diagrama eléctrico unifilar.	28
Figura 3.12	Hardware del sistema de inspección por visión artificial.	28
Figura 3.13	Diagrama de bloques estructura de software.	29
Figura 3.14	(a) Imagen escaneada, (b) Imagen escaneada con sistema de iluminación.	30
Figura 3.15	Diagrama de flujo preprocesamiento de la imagen PCB.....	31

Figura 3.16 (a) Imagen del PCB binaria, (b) Imagen del PCB binaria invertida.....	32
Figura 3.17 (a) Imagen del PCB binaria, (b) Imagen del PCB binaria filtrada.....	33
Figura 3.18 Diagrama de flujo alineación de las imágenes ITest e IRef.....	34
Figura 3.19 Alineación de las imágenes, (a) IRef , (b) ITest.....	35
Figura 3.20 Diagrama de flujo algoritmo detección de errores.	36
Figura 3.21 IRef XOR ITest.	37
Figura 3.22 (a) IRef XOR ITest , (b) Defectos en la imagen PCB.....	38
Figura 3.23 Diagrama de flujo reconocimiento del error pad.....	39
Figura 3.24 (a) Imagen drills de referencia, (b) Imagen drills de test.....	40
Figura 3.25 Imagen IDE.....	40
Figura 3.26 (a) Contornos pads, (b) Error daño en pads.....	41
Figura 3.27 Diagrama de flujo, reconocimiento del error espurio.....	42
Figura 3.28 Contorno espurio canny.....	43
Figura 3.29 (a) Espurio PCB, (b) Espurio imagen ITest.	43
Figura 3.30 Diagrama de flujo reconocimiento de cortocircuito y circuito abierto...	44
Figura 3.31 Análisis matricial de contornos hiearchy.....	45
Figura 3.32 Circuito abierto.	46
Figura 3.33 Ventana de trabajo.	47
Figura 3.34 Menú bar del sistema de inspección.	48
Figura 4.1 Algoritmo detección de errores en PCB 7.	51
Figura 4.2 Funcionamiento aplicación para PCB 7.	52
Figura 4.3 Funcionamiento algoritmo en PCB 8.	52
Figura 4.4 Funcionamiento aplicación para PCB 8.	53
Figura 4.5 Visión Artificial VS Visión Humana.....	55
Figura 4.6 Visión Artificial VS Visión Humana.....	57

ÍNDICE DE TABLAS

Tabla 3.2 Especificaciones técnicas PC.	21
Tabla 3.3 Características microcontrolador Atmega 328.....	21
Tabla 3.4 Especificaciones técnicas motor a pasos.....	22
Tabla 3.5 Especificaciones técnicas driver.	22
Tabla 3.6 Especificaciones técnicas Fuente DC.	23
Tabla 3.7 Partes del sistema deslizante.	24
Tabla 3.8 Matriz contornos hierarchy.	45
Tabla 4.1 PCBs 7, 8, 2 a ser examinados.	49
Tabla 4.2 PCBs 5, 6 a ser examinados	50
Tabla 4.3 Inspección por visión artificial (I.V.A.).	53
Tabla 4.4 Inspección por visión humana con y sin experiencia.....	54
Tabla 4.5 Visión artificial vs visión humana (número de errores).....	54
Tabla 4.6 Visión artificial vs visión humana(tiempo).....	56
Tabla 4.7 Repeticiones del sistema de inspección mediante visión artificial.	57

RESUMEN

El desarrollo del sistema de inspección automática de PCBs mediante visión artificial facilita, simplifica y evita problemas de cansancio, a las personas encargadas de realizar la inspección, de la empresa DINNOVASOFT. En la etapa inicial se implementa el hardware, que integra el sistema de adquisición a través de un sensor óptico (escáner), y para el procesamiento digital de imágenes se realiza a través de un ordenador; para el alojamiento y funcionamiento del hardware del sistema de adquisición, se ha construido una estructura metálica que contiene, un sistema de iluminación, encargado de intensificar el flujo lumínico sobre el área del PCB, un sistema que desplaza el PCB hacia la ventana del escáner, mediante el giro de un motor a pasos controlado por un Arduino, y por último un escáner encargado de la captura y digitalización de imagen del PCB; la siguiente etapa es el desarrollo del software, que se realiza en un PC, en lenguaje programación Python, OpenCV y PYQT5, el método es detectar el defecto en función de la imagen digital del PCB, utilizando tecnologías de procesamiento computacional de imágenes, el algoritmo se basa en la entrada de dos imágenes, una a ser examinada en busca de errores y la otra es la imagen de referencia sin errores, estas dos imágenes al compararse y mediante operaciones de sustracción, análisis, y reconocimiento, proporcionan como resultado los errores del PCB, finalmente estos defectos pueden ser visualizados por una interfaz gráfica.

Palabras clave: escáner, visión artificial, inspección de PCB, GUI.

ABSTRACT

The development of the automatic inspection system for PCBs by means of artificial vision makes it easy, simplifies and avoids fatigue problems for the people in charge of carrying out the inspection, of the DINNOVASOFT company. In the initial stage, the hardware is implemented, which integrates the acquisition system through an optical sensor (scanner), and for the digital image processing it is carried out through a computer; For the housing and operation of the hardware of the acquisition system, a metallic structure has been built that contains, a lighting system, in charge of intensifying the light flux over the PCB area, a system that moves the PCB towards the scanner window, by turning a stepper motor controlled by an Arduino, and finally a scanner in charge of capturing and digitizing the PCB image; The next stage is the development of the software, which is carried out on a PC, in Python, OpenCV and PYQT5 programming language, the method is to detect the defect based on the digital image of the PCB, using computational image processing technologies, the algorithm is based on the input of two images, one to be examined for errors and the other is the reference image without errors, these two images when compared and through subtraction, analysis, and recognition operations, resulting in the errors of the PCB, finally these defects can be visualized by a graphical interface.

Keywords: scanner, machine vision, PCB inspection, GUI.

INTRODUCCIÓN

En el proyecto se detalla el desarrollo de un sistema de inspección automático de PCBs, mediante visión artificial para la empresa DINNOVASOFT en la ciudad de Quito-Ecuador.

En el capítulo 1, se expone los antecedentes y la problemática de la utilización de la inspección manual o humana en PCBs, y se plantea los objetivos del trabajo de titulación.

En el capítulo 2, se sustenta la fundamentación de la teoría sobre conceptos como: control de calidad, productividad, visión artificial, PCBs y se recopila información de 5 aplicaciones de proyectos similares, esta investigación contiene diagramas de flujo y/o bloques del funcionamiento y sus respectivas GUI.

En el capítulo 3, se muestra la estructura del sistema de inspección por visión artificial, que incluye planos, el hardware del sistema de adquisición de imágenes como el hardware del procesamiento computacional de imágenes con sus respectivos diagramas de bloques, se plantea el software utilizado, que contiene diagramas de flujo del sistema de inspección, así como los diagramas de flujo individuales de los algoritmos de clasificación de los errores: pistas, pads, espurio, creación de la interfaz de usuario.

En el capítulo 4, se realizan pruebas de funcionamiento y verificación del sistema de inspección en PCBs de diferente tamaño y densidad de pistas.

En el capítulo 5, se concreta las conclusiones y recomendaciones de los resultados alcanzados en las pruebas de funcionamiento del sistema.

CAPÍTULO 1

ANTECEDENTES

1.1 PROBLEMA DE ESTUDIO

DINNOVASOFT es una empresa dedicada al desarrollo de la tecnología y fabricación manual de PCBs, durante la fabricación se producen errores, tales como: daños en pistas, daños en pads, espurio, entre otros, que pueden causar graves daños a corto o largo plazo en el funcionamiento del circuito.

La detección de estos defectos puede convertirse en una actividad difícil para los inspectores que lo realizan de forma manual, ya que la visibilidad de un objeto es un proceso activo y repetitivo en el cual el observador requiere de concentración, durante un periodo prolongado de tiempo, pudiendo ocasionar errores u omisiones, por otro lado las nuevas tecnologías electrónicas necesitan nuevos diseños de PCB, con dimensiones más pequeñas, nuevos componentes y nuevas funcionalidades, lo que hace que la inspección humana se torne más compleja de ejecutar.

1.2 JUSTIFICACIÓN

La automatización de los procesos de inspección visual se ha vuelto esencial para mejorar la eficiencia en la industria manufacturera de PCBs, la automatización de estos sistemas, tiene ventaja sobre la inspección humana, poseen técnicas innovadoras y contribuyen al avance tecnológico de la industria electrónica.

Dentro de las principales tareas de este sistema son: detección, ubicación y clasificación de defectos en PCBs desnudos de forma automática, simplificar la tarea y evitar problemas relacionados con la fatiga, a las personas encargadas de realizar la inspección, además de ser una alternativa que busca maximizar la productividad. Al tiempo que ayuda a reducir costos, tiempo y mejorar la gestión de riesgos.

El proceso de inspección se puede automatizar mediante la sustitución de la visión humana por la visión artificial, mediante la selección de técnicas adecuadas, procesamiento de imágenes y algoritmos informáticos.

1.3 OBJETIVOS

1.3.1 Objetivo General

Desarrollar un sistema de inspección automático para la detección de fallos en PCBs de simple cara, de hasta 21x30cm, mediante visión artificial para la empresa DINNOVASOFT.

1.3.2 Objetivos Específicos

- Recopilar información de prototipos de sistemas de inspección de PCBs por visión artificial existentes, para la determinación de componentes tanto de hardware y software utilizables.
- Implementar el hardware para el sistema de inspección automático, con visión artificial, para la detección de fallos en placas de PCBs de hasta 21x30cm.
- Diseñar un sistema de captura y adquisición de imágenes, que permita el reconocimiento de errores, mediante técnicas de procesamiento digital de imágenes.
- Desarrollar una interfaz gráfica que facilite al usuario, el acceso al sistema de inspección automático de PCBs.
- Realizar un análisis comparativo de detección de errores, entre inspección por visión artificial e inspección por visión humana, para la evaluación de los resultados del sistema final.

1.4 METODOLOGÍA

Dado que el proyecto se basa principalmente en la visión artificial para el control de calidad se utilizará la investigación experimental, que es “un tipo de método de investigación en el que el investigador controla deliberadamente las variables para delimitar relaciones entre ellas y está basado en la metodología científica” y que implica la observación, manipulación, registro de las variables que afectan nuestro objeto de estudio; como son la adquisición de imágenes, tipos de errores a detectar.

El método comparativo “procedimiento sistemático de contrastación de uno o más fenómenos, a través del cual se buscan establecer similitudes y diferencias entre ellos” para realizar un análisis comparativo de detección de errores entre inspección por visión artificial e inspección por visión humana.

CAPÍTULO 2

FUNDAMENTACIÓN TEÓRICA

2.1 CONTROL DE CALIDAD

La palabra calidad se refiere a un producto que supera el 100% de manera satisfactoria, dentro de las etapas que intervienen en su fabricación. Los diferentes departamentos que aportaron para la elaboración de un producto de calidad como el de ingeniería, producción, compras, personal; llevan a cabo distintas labores encaminadas a la terminación del producto que demanda el consumidor, hasta terminar el proceso en la inspección de calidad que evalúa el producto y determina las modificaciones que considere necesario hasta hacerlo satisfactorio para el cliente. El control de calidad es el conjunto de procesos que utiliza la empresa para la elaboración de un producto de calidad, a su vez la empresa invierte, produce y arroja ganancias, para todos los miembros de una empresa. (Saúl, 2014)

2.2 PRODUCTIVIDAD

“La productividad es la relación insumos-productos en cierto periodo con especial consideración a la calidad.” (Weihrich, 2012)

Productividad es la relación entre la cantidad de bienes y servicios producidos y la cantidad de recursos utilizados. En el proceso de fabricación, la productividad evalúa el rendimiento de las máquinas, equipos de trabajo y las personas que operan en el desarrollo de un producto. Los factores importantes que afectan la productividad en empresas se determinan por:

2.2.1 Recursos Humanos

Es un factor determinante de la productividad, está compuesto por personas que dan vida a una organización. Las personas se relacionan e interactúan para obtener metas en una empresa, este hecho genera procesos psicológicos y psicosociales que afectan los resultados de una organización, debido a que influye directamente en un proceso y de éste dependen los demás factores.

2.2.2 Maquinaria y equipo

La maquinaria y los equipos ayudan a la producción, por lo tanto, es primordial tomar en consideración: calidad, estado de la maquinaria, mantenimientos y el correcto manejo de los equipos.

2.2.3 Organización del trabajo

Este factor hace referencia a la estructuración y/u organización, conformación, distribución y diseño de los espacios de trabajo, que se distribuyen de acuerdo al tipo de maquinaria, equipos y actividad que se ejecuta en cualquier proceso. (Silvia, 2012)

2.3 VISIÓN ARTIFICIAL

La Asociación de Imágenes Automatizadas (AIA), expone que la visión artificial abarca la mayoría de las aplicaciones industriales en las que se combinan hardware y software proporcionando así una alineación operativa en la ejecución de sus funciones a dispositivos, fundamentadas en la captación y procesamiento de imágenes. Por lo tanto, la visión artificial aplicada a la industria posee características como: bajo costo, precisión, confiabilidad, resistencia, estabilidad mecánica y de temperatura.

Los sistemas de visión artificial se basan en sensores digitales, cámaras industriales o escáneres con ópticas especializadas para adquirir imágenes, de modo que el hardware y el software del procesador que se utilice puedan procesar, analizar y medir diversas características para la toma de decisiones o cumplir funciones específicas para lo cual fueron creados. (Chauhan, 2011)

2.4 PROCESAMIENTO DIGITAL DE IMÁGENES

El procesamiento digital de imágenes está presente en muchas aplicaciones industriales entre las cuales podemos nombrar: ciencias médicas, biometría e identificación, agricultura, ganadería, satélites de exploración terrestre, así como también se encuentran en varias tareas de distinta índole en la sociedad. Para tener más claro el concepto de procesamiento de una imagen es necesario conocer la definición de algunos términos que son parte del procesamiento digital: (Doniwar, 2017)

2.4.1 Imagen

Una imagen se considera como la unión de varios puntos de colores, es decir, una sucesión de puntos relacionados entre sí, que forman una matriz de información para el uso digital; estos puntos se llaman píxeles (picture element) y se definen como la menor unidad homogénea en color y que es parte esencial de una imagen digital, tal cual indica la Figura 2.1.

Figura 2.1 Píxeles de una porción de la imagen.



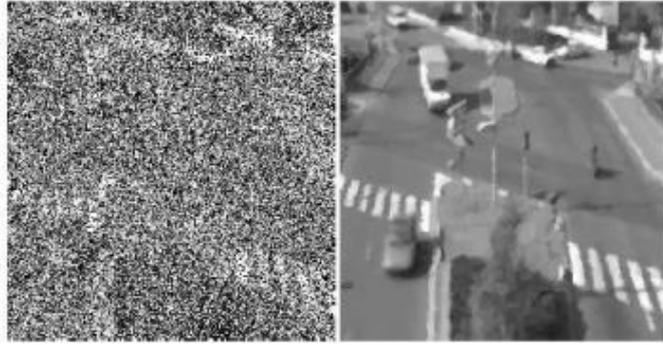
Fuente: (Querejeta Simbeni, 2017)

2.5 FILTRADO DE UNA IMAGEN

La presencia de ruido puede afectar significativamente la información de las características de la imagen, por lo tanto, se puede encontrar, dentro de preprocesamiento de las imágenes, filtros para la restauración de las señales que han sido distorsionados. El filtrado de una imagen digital es la operación que se realiza sobre los píxeles para mejorarla, el objetivo de un filtro puede ser remarcar cierta información o conseguir un efecto especial dentro de una imagen. Las tareas principales en el filtrado de una imagen son:

- Suavizado de la imagen: consiste en disminuir las variaciones de intensidad entre píxeles vecinos.
- Realzado de bordes: consiste en acentuar los bordes en una imagen.
- Eliminación del ruido: consiste en eliminar píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos, típicamente el ruido es generado en la adquisición de la imagen o en el proceso de transmisión.

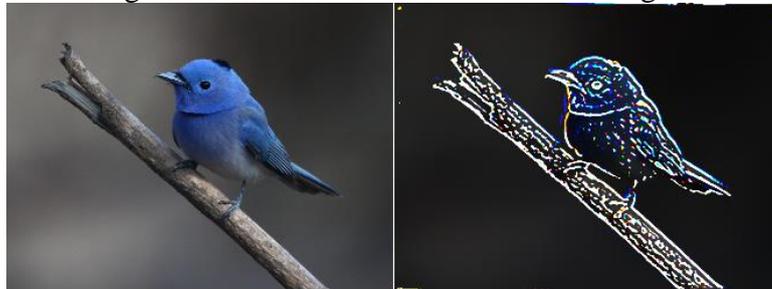
Figura 2.2 Filtrado de ruido al 75% de píxeles ruidosos.



Fuente: (Querejeta Simbeni, 2017)

- Detección de bordes: consiste en descubrir los píxeles donde se produce un cambio brusco en la función intensidad. (Querejeta Simbeni, 2017)

Figura 2.3 Detección de los bordes de imagen.



Fuente: (Querejeta Simbeni, 2017)

2.5.1 Filtrado en el dominio de la frecuencia

Trabaja en la transformada de Fourier de la imagen, consiste, en modificar el teorema de convolución de señales, de modo que lo primero que se efectúa es la transformada de Fourier de la imagen, luego se emplea el filtro, finalmente se realiza transformada inversa de Fourier, para devolverla al dominio espacial, los filtros más comunes son:

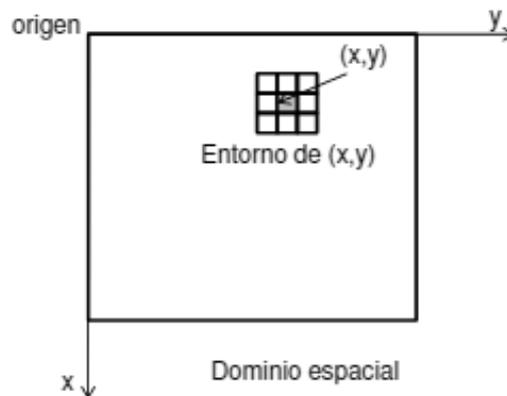
- Filtros paso bajo: En este filtro pasan los elementos de frecuencia baja y por el contrario impiden el paso de elementos de frecuencia alta, las aplicaciones de este filtro incluyen: eliminación de ruido, suavizado y reducción de la nitidez.
- Filtros paso alto: En este filtro pasan los elementos de frecuencia alta y, por el contrario, impiden el paso de elementos de frecuencia baja; la aplicación de este filtro es la detección de zonas de contraste elevado como bordes.
- Filtros Paso Banda: En este filtro pasan elementos de cierto rango de frecuencia; las aplicaciones de este filtro es aislar los efectos de ciertas bandas de frecuencias seleccionadas sobre una imagen y eliminar ruido, pero se

aplican menos, debido a que elimina contenido importante de las imágenes.
(Viera-Maza, 2017)

2.5.2 Filtrado en el dominio del espacio

Trabaja sobre el plano de la imagen, directamente en los píxeles de la imagen, mediante la convolución de máscaras cuadradas, las operaciones de filtrado espaciales se precisan en un entorno de vecindad del punto a transformar (x, y) Figura 2.4.

Figura 2.4 Filtros en el dominio del espacio.



Fuente: (Sergio Sada Pezonaga, 2015)

- Obtención de contornos: Se fundamenta en las variaciones de intensidad que se dan píxel a píxel. Demandan menor coste computacional, por que trabajan desde el operador gradiente en una máscara, se consigue el contorno de la imagen y clasifica las formas existentes en ella, filtros de Sobel, Prewitt, Roberts y contornos Canny son los más utilizados.
- Realce de bordes por desplazamiento y diferencia: Quita de la imagen original una copia desplazada de la misma, como resultado se obtiene el realce de bordes existentes, según la máscara de convolución utilizada sea horizontal, vertical o diagonal (horizontal/vertical).
- Laplaciano: Este filtro exalta los bordes en todas las orientaciones, ya que se emplea el operador laplaciano, y que permite mejorar resultados con un aumento de ruido en la imagen.
- Filtrado de realce de bordes por gradiente direccional: Se emplea para destacar y resaltar con mayor precisión los bordes que se localizan en una dirección determinada. (González, 2012)

2.6 FABRICACIÓN DE PLACAS DE CIRCUITO IMPRESO PRINTED CIRCUIT BOARD (PCB)

Una placa de circuito impreso es una plancha conocida como baquelita que está compuesta de un material rígido aislante, esta a su vez posee una capa de cobre por una de sus caras o por ambas, que va a servir para conectar distintos componentes eléctricos y electrónicos que se montarán sobre y por debajo de ella; dependiendo si es una placa de una sola capa o multicapa.

El material aislante se compone generalmente de fibra de vidrio, que es un material incombustible y excelente aislante incluso con poco espesor, mientras que la capa conductora es una lámina de cobre, según sea el tipo de placa, la capa de cobre puede estar protegida por una capa de resina fotosensible. (Ulloa javier, 2017)

La Figura 2.5 se observa las capas de las que está compuesta una placa para realizar una baquelita.

Figura 2.5 Capas de la placa para fabricación de PCBs.



Fuente: (Ulloa javier, 2017)

2.7 MÉTODOS DE FABRICACIÓN DE PCBs

Existen varios métodos para la fabricación de placas de circuitos electrónicos PCBs, entre los métodos más utilizados tenemos:

2.7.1 Impresión serigráfica

Utiliza tintas que resisten el grabado, para proteger la capa de cobre que son las líneas del circuito. Los procesos de grabado posteriores eliminan el cobre no deseado.

2.7.2 Fotograbado

Utiliza una placa especial, esta es una placa de circuito impreso en la que la capa de cobre viene cubierta por una capa de resina fotosensible y se la conoce como placa sensibilizada. Esta resina posee unas propiedades químicas que, al exponerse a la luz ultravioleta, retira la laca protectora sólo en las zonas donde se deba eliminar el cobre; para proteger en este caso las pistas del circuito, se coloca un acetato donde está su esquema.

2.7.3 Fresado de circuitos impresos

Utiliza equipos modernos o maquinas CNC de 2 o 3 ejes para retirar el cobre del sustrato no deseado. Los diseños de circuitos en este tipo de máquinas se los realiza con programas específicos.

2.7.4 Impresión en material termosensible

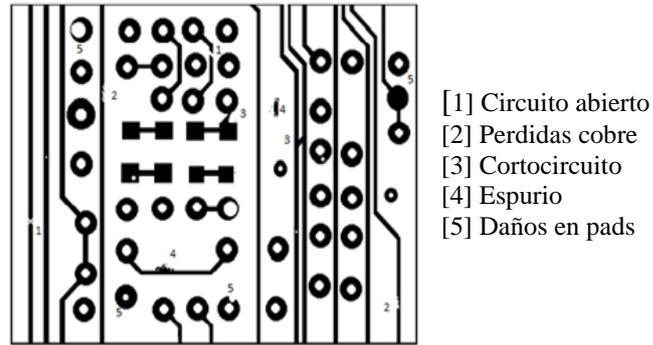
Se imprime el diseñado en papel glossy (fotográfico) o cualquier tipo de papel que tenga cera y se transfiere el diseño a la placa de cobre a través de calor. (Morales Rodrigo, 2018).

2.8 TIPO DE ERRORES EN PCBs

Lograr un diseño sin errores en placas PCBs no es un trabajo sencillo; de acuerdo a la complejidad del circuito, en el diseño y fabricación de las placas PCBs, se puede llegar a producir numerosas fallas, que se describen a continuación:

- Cortes en pistas: Este daño describe un circuito abierto o corte en la pista.
- Perdidas de cobre: Se da cuando parte de la pista está incompleta o tiene una parte de cobre faltante, pero esta no cortada.
- Daños en pads (Hole): Se da en los pads donde debería ir el elemento, este tipo de daño puede describir pads incompletos con su centro faltante, tamaños excedentes o faltantes en los mismos y deformidad de los pads.
- Cortocircuito: Se da cuando las pistas se unen entre sí, ya sea con pads o entre líneas de pistas diferentes.
- Espurio de cobre: Este tipo de daño muestra un excedente de cobre en las pistas o alguna pista extra que no debe estar en el circuito de la placa. (Kunte, 2018)

Figura 2.6 Tipos de errores en PCBs.



Fuente: (Kunte, 2018)

2.9 INSPECCIÓN AUTOMÁTICA DE DEFECTOS

La fabricación de circuitos impresos genera múltiples defectos durante sus procesos químicos, estos deben ser detectados en las primeras etapas de producción para ser corregidos a tiempo.

La inspección automática de defectos de PCBs de imágenes naturales utiliza técnicas de procesamiento de imágenes, mediante software computacional, esta tecnología permite al fabricante corregir en determinadas etapas del proceso de manufactura los defectos, o desechar las tarjetas que no se puedan reparar, previniendo un desgaste innecesario de recursos en circuitos que posiblemente habrá que desechar y garantizando además la calidad de la fabricación de los circuitos impresos, evitando que al usuario final le lleguen defectos que afecten negativamente el funcionamiento de su circuito una vez sea ensamblado. (Doniwar, 2017)

2.10 APLICACIONES DE INSPECCIÓN DE PCB POR VISIÓN ARTIFICIAL

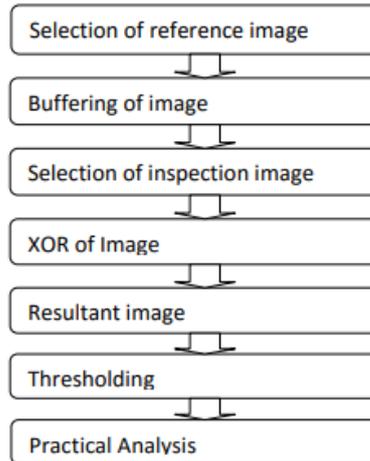
2.10.1 Aplicación 1

Aplicación 1 “Detección de fallas de PCB mediante el método de sustracción de imágenes” desarrollado por Ms. VinitaP y Mr. Shivkant KaushikP, publicado en enero 2016, por la International Journal of Scientific Engineering and Applied Science (IJSEAS); inicialmente las fallas se han detectado manualmente con lupa, dado que las fallas son muy pequeñas y difíciles de encontrar, por lo que se ha desarrollado una técnica de procesamiento de imágenes; primero se compara una imagen estándar de PCB sin errores con una imagen del PCB con errores, utilizando el algoritmo de la operación resta, a través de la cual se puede resaltar las principales regiones con defectos, después y a través de la técnica de análisis de conexión podemos encontrar

los errores fatales y potenciales. El hardware para este sistema se basa en los nuevos avances tecnológicos de computadores, cámaras, y el software en algoritmos de visión por computador.

La Figura 2.7 detalla el diagrama de bloques de la programación de la aplicación 1.

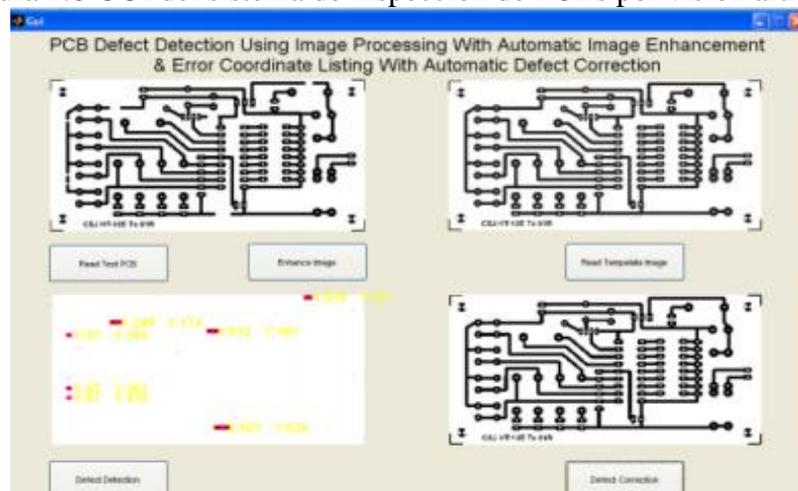
Figura 2.7 Diagrama de bloques aplicación 1.



Fuente: (Ms. VinitaP, 2016)

En este sistema; primero se realiza la selección de la imagen de referencia, se almacena en un búfer para organizar los píxeles, luego se selecciona la imagen a ser inspeccionada, estas dos imágenes se comparan con la operación XOR, y se obtiene la imagen resultante, esta imagen se transforma a binaria, el proceso termina con el análisis y detección de errores Figura 2.8.

Figura 2.8 GUI del sistema de inspección de PCBs por visión artificial.



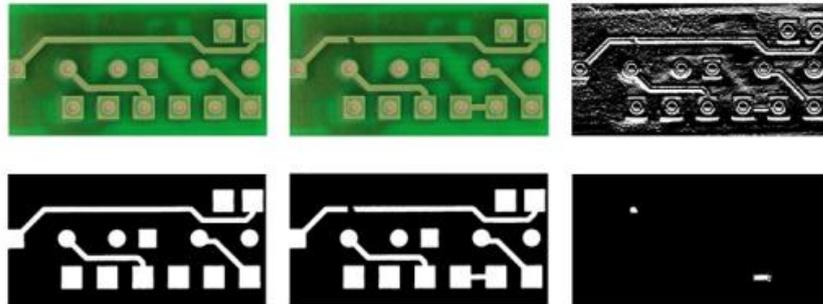
Fuente: (Ms. VinitaP, 2016)

Este sistema detecta fácilmente todas las fallas en corto tiempo, localiza la ubicación exacta de la falla y también muestra el tipo de defecto; con esta información se puede tomar acciones inmediatas para la corrección del PCB.

2.10.2 Aplicación 2

Aplicación 2 “Inspección óptica automatizada de un circuito impreso” por Kunte y Ondřej, enero 2018, en la ciudad de Praga dentro de la República Checa; el método se basa en la resta de las imágenes; la imagen del PCB con errores y la imagen de PCB sin errores, las dos imágenes se comparan mediante algoritmos de sustracción, las diferencias entre las imágenes se procesan a fondo ya que pueden utilizarse como señal de defectos. Los métodos difieren según el tipo de segmentación utilizado; el proceso de segmentación permite identificar las partes significativas de la imagen, por lo tanto, permite definir el área y/o bordes de la imagen, para esta aplicación se utiliza dos métodos; el primer método utiliza una segmentación de áreas y el segundo método utiliza una segmentación de borde, Figura 2.9.

Figura 2.9 (a) Segmentación de área, (b) Segmentación de borde.



Fuente: (Kunte, 2018)

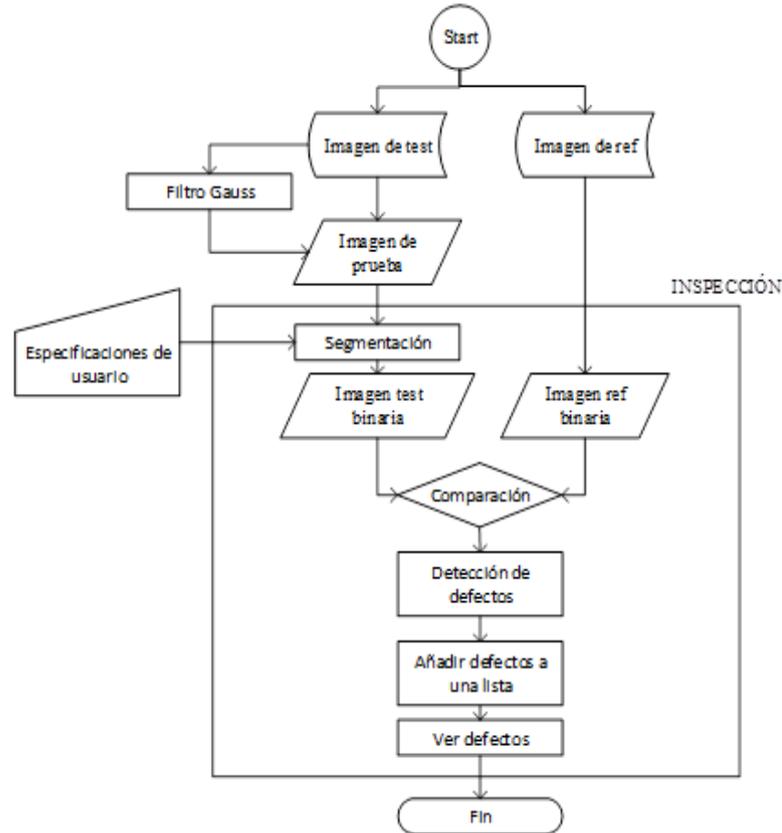
Figura 2.10 Hardware del sistema de inspección óptica automatizada.



Fuente: (Kunte, 2018)

La Figura 2.11, describe la programación de la aplicación 2: Inspección óptica automatizada de un circuito impreso.

Figura 2.11 Diagrama de flujo de la aplicación 2.



Fuente: (Kunte, 2018)

El sistema inicia; cargando la imagen de prueba (placa PCB) con errores y la imagen de referencia sin defectos, la imagen de prueba es calibrada por la cámara para eliminar la distorsión, se dimensiona y se aplica un filtro de suavizado, luego la imagen es segmentada por el método de detección de bordes, luego ambas imágenes son transformadas a binarias, el siguiente paso es la comparación de las dos imágenes binarias, como resultado de la diferencia de estas imágenes, se obtiene la detección de errores, estos son guardados en una lista y visualizados posteriormente en una pantalla.

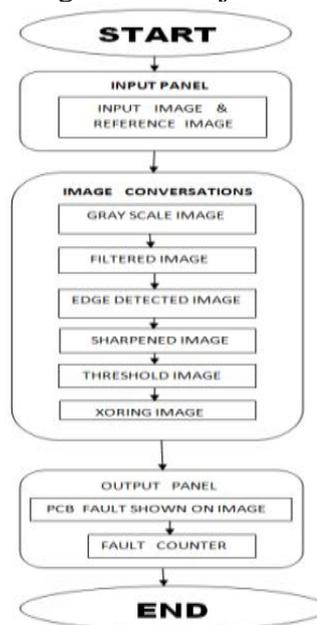
2.10.3 Aplicación 3

Aplicación 3 “Inspección de PCB desnudo para circuito abierto, daños en pads usando operaciones de procesamiento de imagen” desarrollado por Doniwar, Dadhe y Baiswara en el año 2017, publicado en Internacional de Investigación en Ingeniería

Computacional (IJCER) en Maharashtra India; esta aplicación consiste en el procesamiento digital de imágenes para la detección de fallas en el PCB mediante el software MATLAB. El algoritmo consiste en dos imágenes de entrada, una para inspeccionar en busca de errores, es decir, el diseño del circuito PCB y otra es la imagen de referencia o la imagen estándar de la PCB. Después de este proceso, se obtiene las fallas que existen en el PCB, como: agujeros, rupturas, etc., esta aplicación ayuda a detectar la falla en la etapa primaria del diseño.

La Figura 2.12 describe la programación de la aplicación 3: Inspección del PCB desnudo para circuito abierto, daños en pads usando operaciones de procesamiento de imagen.

Figura 2.12 Diagrama de flujo de la aplicación 3.



Fuente: (Doniwar, 2017)

El sistema inicia; con la entrada de las imágenes; la imagen del PCB y la imagen de referencia, el siguiente paso es el preprocesamiento de las imágenes; inicia con la conversión a escala de grises, filtrado de la imagen, detección de bordes, afilado de bordes de la imagen, usado para realzar la línea estructural en una imagen, se transforma las imágenes en formato binario, el siguiente paso es redimensionamiento de ambas imágenes, estas deben tener el mismo tamaño en píxeles. A la salida de sistema podemos observar los defectos en una imagen numerados, tal cual señala la Figura 2.13.

Figura 2.13 GUI de la aplicación 3.



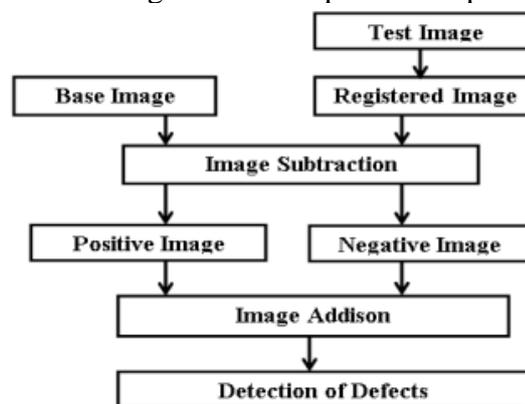
Fuente: (Doniwar, 2017)

2.10.4 Aplicación 4

Aplicación 4 “Detección automática de defectos en PCB” desarrollado por Ashish Singh, Vimal H. Nayak y Mohammed G. Vayada, publicado en (IJIRST) International Journal for Innovative Research in Science & Technology, noviembre 2014; en el artículo se analiza la inspección automática de la máquina utilizando un algoritmo de resta. En este algoritmo compara una imagen ideal con la imagen de prueba en la que el defecto debe ser detectado como agujero faltante, daños en pads, cortocircuito, mordida de ratón, circuito abierto, etc.

En la Figura 2.14 se detalla el diagrama de flujo de la aplicación 4: Detección automática de defectos en PCB.

Figura 2.14 Diagrama de bloques de la aplicación 4.



Fuente: (Ashish Singh, 2014)

Se importa la imagen de referencia, y las imágenes de test; la imagen de test se registra, luego estas dos imágenes se restan, y se obtiene una nueva imagen con las diferencias, de esta imagen resultado de la resta, la imagen positiva se invierte y la suma de las imágenes; la imagen positiva y la imagen negativa dan como resultado una nueva imagen en la que se detectan las fallas, y que es evidenciado en la Figura 2.15.

Figura 2.15 Detección automática de defectos en PCB.

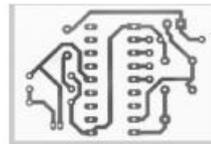


Fig. 6: Reference Image

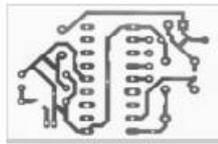


Fig. 7: Test Image

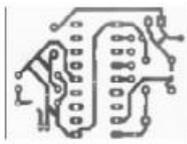


Fig. 8: Registered Image



Fig. 9: Positive Image



Fig. 10: Negative Image



Fig. 11: Etching effect



Fig. 12: Total defect in PCB



Fig. 13: Wrong hole size

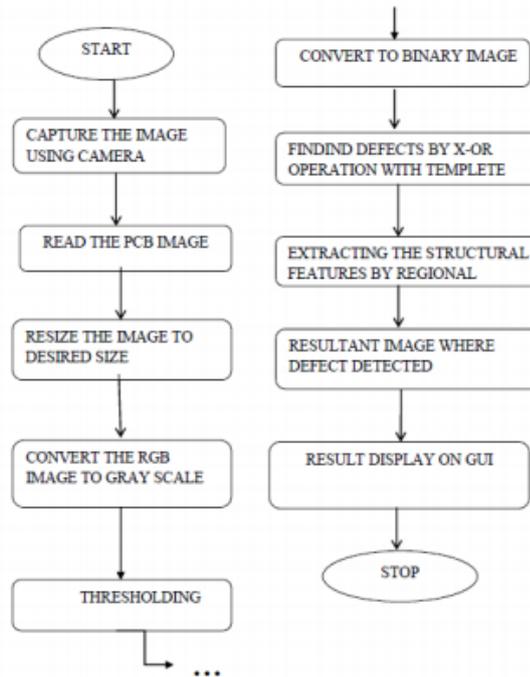
Fuente: (Ashish Singh, 2014)

2.10.5 Aplicación 5

Aplicación 5 “Detección de defectos de PCB mediante procesamiento de imágenes y sistema integrado” desarrollado por Neelum Dave, Vikas Tambade, Balaji Pandhare, Suman Saurav publicado en International Research Journal of Engineering and Technology (IRJET), mayo 2016; la técnica básica del sistema propuesto es detectar el defecto en base a la imagen digital de la PCB mediante técnicas de procesamiento de imágenes, el sistema debe recopilar imágenes de PCB a través de la cámara. Después se realiza el proceso de reconocimiento y análisis de imágenes, las imágenes debe compararse con las plantillas, por lo que los defectos comunes se pueden detectar con precisión, se utiliza un microcontrolador para el control de un motor para el desplazamiento de una cinta transportadora.

La Figura 2.16 describe el diagrama de flujo de la aplicación 5: Detección de defectos de PCB mediante procesamiento de imágenes y sistema integrado.

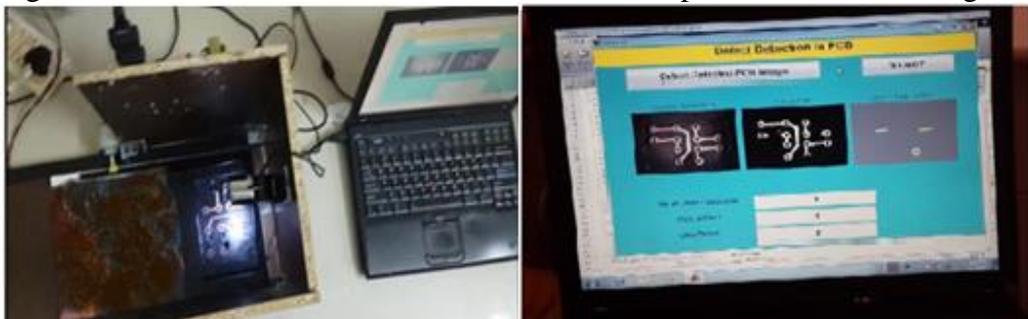
Figura 2.16 Diagrama de flujo de la aplicación 5.



Fuente: (Neelum Dave, 2016)

Para este sistema; en el primer paso se captura la imagen del PC usando una cámara, una vez digitalizada, se lee la imagen, el siguiente proceso es el redimensionamiento, conversión a escala de grises, se establece el umbral en la imagen y se convierte a una imagen binaria, lo siguiente es la utilización del operador XOR, se identifican las diferencias entre las imágenes, arrojando como resultado los defectos en una nueva imagen y por último los resultados se pueden visualizar en una pantalla por una interfaz gráfica Figura 2.17.

Figura 2.17 Detección de defectos de PCB mediante procesamiento de imágenes.



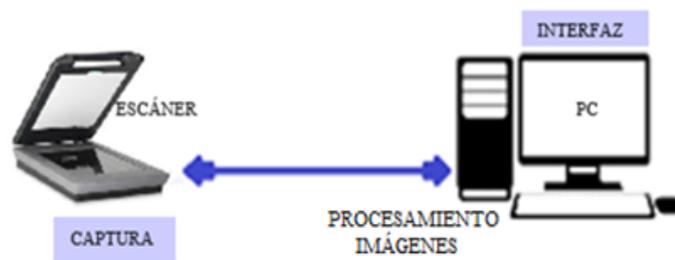
Fuente: (Neelum Dave, 2016)

sistema de desplazamiento lineal, por medio del giro de un motor a pasos controlado por un Arduino, que hace desplazarse a todo el sistema deslizante y que, transporta la placa PCB desde el exterior del armario hacia la ventana del escáner hasta adherirse completamente; luego entra en funcionamiento la segunda parte del hardware que se basa en un computador, este por medio del interfaz gráfico programado en Python y PYQT5 envía una señal, para iniciar la digitalización de la placa PCB y la almacena en un espacio de memoria del computador, la transmisión del archivo se realiza por conexión wifi. Con la imagen de la placa PCB, y la imagen de referencia que son los datos de entrada al sistema, el ordenador realiza el procesamiento computacional, a través del lenguaje de programación Python y OpenCV.

3.2 DESCRIPCIÓN HARDWARE

El hardware para el sistema de inspección Figura 3.2, integra un ordenador para el procesamiento de imágenes y visualización de errores, y una estructura metálica que contiene el hardware para el sistema de adquisición de imágenes que incluye: una fuente auxiliar de luz, un sistema lineal de desplazamiento, implementado para transportar al PCB de forma automática hacia la ventana del escáner y por último un escáner para la captura de la imagen del PCB.

Figura 3.2 Hardware sistema de inspección por visión artificial.



Elaborado por: Hilda Moreno

Para la construcción de la estructura metálica, primero se detalla cada componente utilizado, seguido se describe paso a paso, la implementación.

3.2.1 Escáner

Se ha seleccionado como sensor óptico un escáner del dispositivo multifuncional Epson L355 para la captura de la imagen del PCB, porque representa una solución

simple, que se ajusta al tamaño A4, y forma plana del PCB, puede convertir el circuito impreso a datos de imagen (foto).

Este proyecto no está destinado a la producción en masa, por lo que no se requiere que la digitalización sea en tiempo real, sin embargo, la velocidad del escáner sigue estando dentro de la inspección en línea, con una velocidad; a una resolución de 300 ppp, como referencia, monocromático: 2.4 milisegundos por línea y color: 9.5 milisegundos por línea. Posee una tecnología que se basa en que la luz se refleje hacia atrás y hacia adelante varias veces mientras se desplaza desde la placa a la lente, sin que exista la necesidad de una distancia entre la lente y la placa, lo que hace que no se requiera enfoque, sea más precisa en la resolución y que la distribución de luz desde el centro a los extremos sea igual en todas las zonas del PCB.

En la Tabla 3.1 se amplían las especificaciones técnicas del escáner L355.

Tabla 3.1 Especificaciones técnicas escáner L355.

	Parámetro	Valor
	Voltaje	100-240V
	Frecuencia:	50-60Hz +/- 0,5Hz
	Resolución	1200 x 2400 dpi
	Profundidad color	48 bits
	Área digitalización	21,6 x 29,7 cm
	Velocidad	Color: 9,5 ms /línea
	Consumo	11W / 2,3W

Elaborado por: Wilson Ayo

3.2.2 PC

Se utiliza un ordenador marca DELL como hardware, para el procesamiento digital de imágenes. En general este PC ofrece una capacidad y rendimiento idóneo para este proyecto, debido a su arquitectura interna, con parámetros como la cantidad de memoria y procesador que permite ejecutar operaciones e instrucciones de máquina simultáneamente.

En la Tabla 3.2, se detalla las especificaciones técnicas de la arquitectura interna del computador.

Tabla 3.1 Especificaciones técnicas PC.

	Procesador	
	Procesador	Intel Core i7
	Velocidad	4.60 GHz
	Cache	8 MB
	Memoria RAM	
	Tipo /Capacidad	DDR4/8GB
	Tarjeta Gráfica	
	Serie	GeForce MX150
	Memoria dedicada	GDDR5 2GB
	Consumo	500 vatios

Elaborado por: Wilson Ayo

3.2.3 Arduino Uno

Es un sistema electrónico de prototipado abierto, basado en software y hardware flexibles, que contiene un microcontrolador basado en el ATmega328 de 8 bits, posee 14 pines digitales, 6 de los cuales son PWM, funcionan como entradas y/o salidas, cada pin trabajan a 5V y puede suministrar o recibir 40 mA de corriente, además cuenta con conexión USB para cargar programas y comunicarse con un PC de forma rápida y fácil, está completado con los elementos necesarios, y características que se adaptan al desarrollo del funcionamiento del circuito electrónico para el sistema deslizante.

Tabla 3.2 Características microcontrolador Atmega 328.

	Parámetro	Valor
	Voltaje entrada	5 V
	Pines de E/S digitales	14/6 PWM
	Corriente pin E/S	40 mA
	Memoria flash	32 KB
	Velocidad de reloj	16 MHz

Elaborado por: Wilson Ayo

3.2.4 Motor a pasos

Se utiliza un motor a pasos Nema 17, bipolar, con un ángulo de paso de 1.8 grados y representa 200 pasos por una vuelta. Se ha seleccionado este motor por que se requiere mayor precisión en el arranque y parada del motor, para evitar que la placa PCB pueda desprenderse de la porta placas, así como prevenir daños en el área de digitalización, el peso capaz de cargar este motor a pasos es de 5.2 kg/cm, peso aproximado de porta placas, placa y sistema deslizante.

En la Tabla 3.4 se describe las especificaciones técnicas del motor a pasos Nema 17.

Tabla 3.3 Especificaciones técnicas motor a pasos.

	Parámetro	Valor
	Voltaje nominal	12V
	Voltaje referencia	5V
	Corriente	1.2 A /bobinado
	Temperatura	-10 a 40 °C
	Torque	5.2 kg/cm
	Pasos por vuelta	200 (1. 8°/paso)

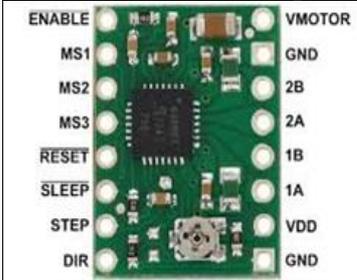
Elaborado por: Wilson Ayo

3.2.5 Módulo A4988

El Módulo A4988 permite controlar específicamente el motor a pasos, tiene incorporado un circuito de control de corriente PWM de tiempo de inactividad fijo, que limita la corriente de carga a un valor deseado y apto para el funcionamiento del motor. Simplifica el trabajo generando señales necesarias para su funcionamiento y además brinda las protecciones necesarias de temperatura y corriente.

En la Tabla 3.5 se amplían las especificaciones técnicas del motor a pasos Nema 17.

Tabla 3.4 Especificaciones técnicas driver.

	Parámetro	Valor
	Voltaje referencia	5V
	Intensidad máxima	2A
	Temperatura	-20 a 85°C
	Tensión máxima	35V

Elaborado por: Wilson Ayo

3.2.6 Fuente de alimentación

Es la encargada de proporcionar un valor de voltaje apropiado para el funcionamiento de los dispositivos electrónicos. Para el ejercicio de este proyecto la fuente de alimentación convierte la entrada de tensión alterna de la red eléctrica 110V, en una tensión continua 5V y 12V.

Tabla 3.5 Especificaciones técnicas Fuente DC.

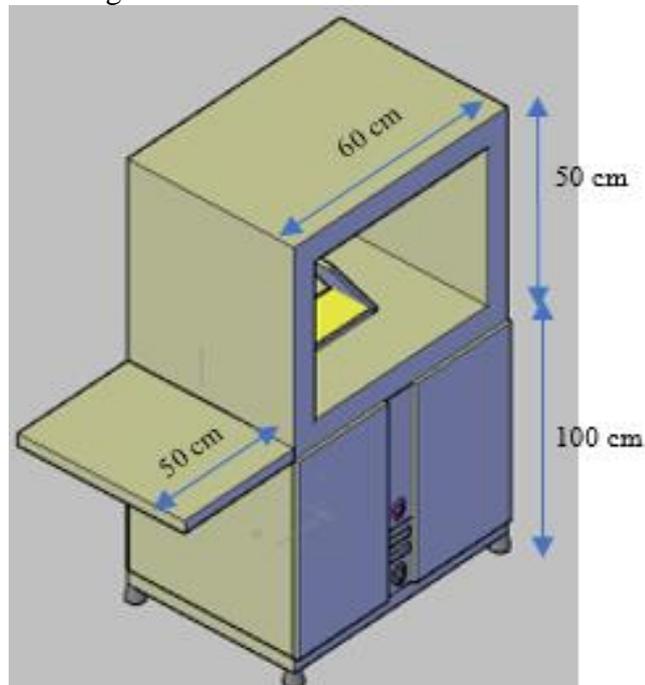
Parámetro	Valor
Voltaje entrada	120-240VAC
Frecuencia	60Hz
Voltaje salida	5V-12V DC
Corriente	5A
Dimensiones	160x98x40mm

Elaborado por: Wilson Ayo

3.3 ESTRUCTURA METÁLICA Y COMPONENTES

Para el alojamiento y funcionamiento del sistema, se ha construido una estructura metálica, Figura 3.3, consta de una cabina en la parte superior, en la que se aloja el hardware para el sistema de adquisición de imágenes, y en la repisa que se encuentra en el lateral izquierdo, se ubica el hardware para el procesamiento digital de imágenes y visualización de errores.

Figura 3.3 Estructura armario metálico.



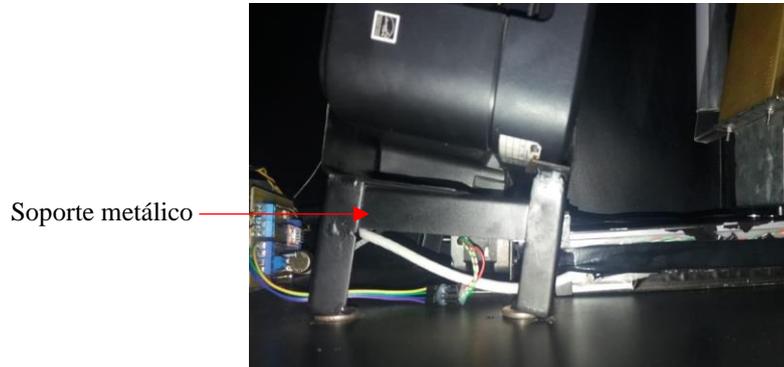
Elaborado por: Wilson Ayo

3.3.1 Acondicionamiento escáner

Para la captura de la imagen del circuito impreso PCB, se utilizó el escáner del dispositivo multifuncional Epson L355, se ha desmontado la impresora y el sistema de

inyección, el dispositivo electrónico se colocó dentro de la cabina del armario metálico de forma vertical, sobre un soporte metálico, Figura 3.4.

Figura 3.4 Ubicación y acondicionamiento del escáner.



Elaborado por: Wilson Ayo

3.3.2 Sistema de desplazamiento

Para desplazar la placa PCB, desde el lateral izquierdo de la estructura metálica hasta la ventana del escáner, primero se realiza el ensamblaje de las partes, que componen el sistema deslizante Tabla 3.7, incluye: placa bracket, tornillo sin fin, perfil de aluminio, eje.

Tabla 3.6 Partes del sistema deslizante.



Elaborado por: Wilson Ayo

El siguiente paso es el acoplamiento del motor al sistema deslizante Figura 3.5, para esto inicialmente; se fija el sistema lineal y motor a la superficie del armario metálico, colocando una base metálica en la parte inferior tanto del sistema lineal como del motor y luego se ajusta con tornillos para que se mantengan estáticos. Finalmente se conecta el eje del motor al eje del sistema lineal deslizante.

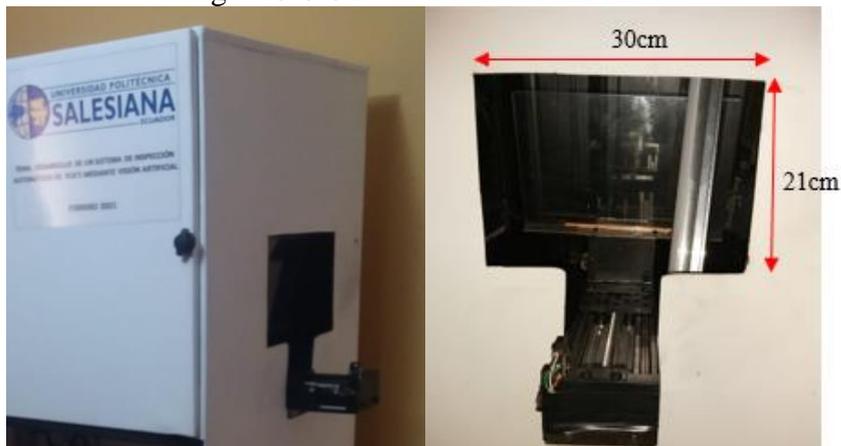
Figura 3.5 Acople del motor al sistema deslizante.



Elaborado por: Wilson Ayo

Para que pueda atravesar la placa PCB, se ha realizado una obertura en el lateral izquierdo del armario metálico, las dimensiones del área son 21x30 cm, tal cual es mostrado en la Figura 3.6.

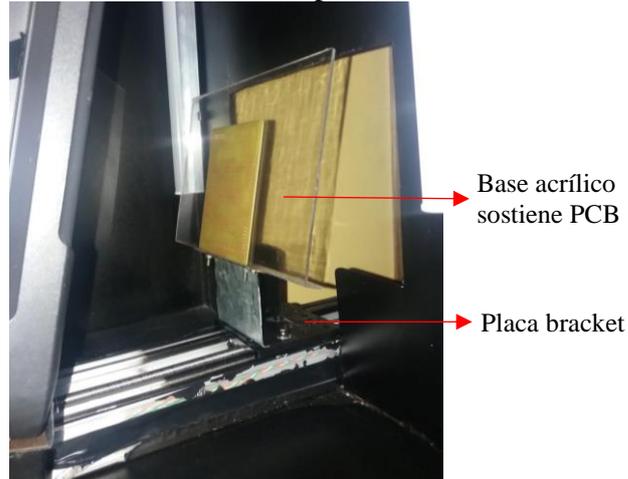
Figura 3.6 Obertura armario metálico.



Elaborado por: Wilson Ayo

Para la sujeción del PCB sobre el sistema deslizante se ha implementado un mecanismo que consiste en fijar con tornillos un soporte metálico sobre la placa bracket, luego sobre esta se ha colocado una base de acrílico de 29x20 cm para sostener el PCB, se ha seleccionado este material para evitar efectos no deseados sobre la placa como se observa en la Figura 3.7.

Figura 3.7 Mecanismo de soporte del PCB.

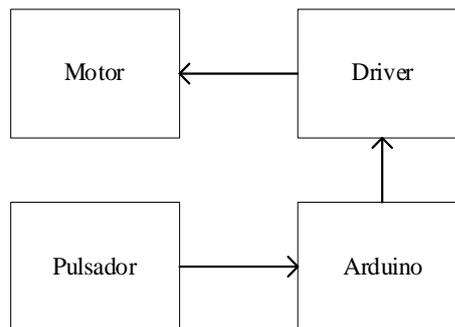


Elaborado por: Wilson Ayo

3.3.3 Diagrama de bloques del control del motor a pasos

La Figura 3.8 detalla el diagrama de bloques del control del motor a pasos.

Figura 3.8 Diagrama de bloques del control del motor a pasos.



Elaborado por: Hilda Moreno

Al presionar un pulsador, se envía una señal de inicio a un pin del Arduino, luego el microprocesador envía la señal de inicio al driver A4988, y el motor se acciona, y gira en sentido horario, efectuando el desplazamiento del sistema lineal y también de la placa PCB hacia adelante, hasta adherirse a la ventana del escáner, al llegar a esta posición y mediante la acción de un sensor fin de carrera detiene el motor y se queda en esa posición hasta que el operador presione el botón de stop, que reinicia el proceso.

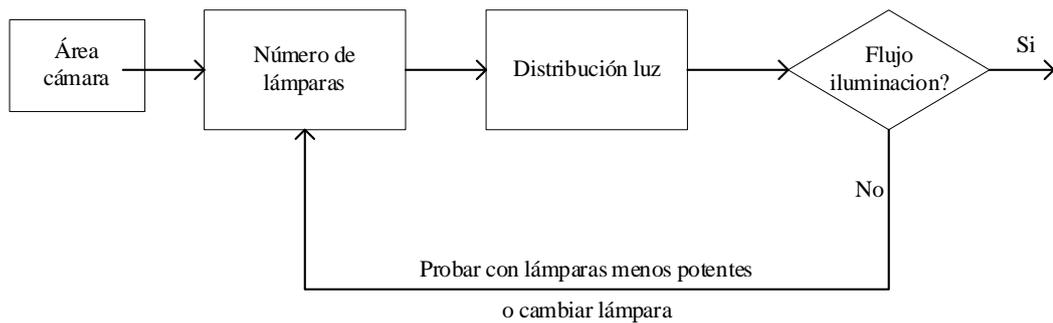
3.3.4 Sistema de Iluminación

El sistema de iluminación, se basó en la construcción de una cámara, pintada de color negro, que contiene una fuente de luz auxiliar, con estos ajustes se busca obtener la

cantidad de luz apropiada, iluminar de manera uniforme en todas las zonas del PCB, minimizar los efectos no deseados sobre el circuito impreso.

Para la selección y ubicación de las luminarias se utilizó el método de lúmenes, que consiste en calcular el valor medio de la iluminancia dentro de la cámara. Este método se describe en el siguiente diagrama de bloques Figura 3.9.

Figura 3.9 Método colocación y especificaciones flujo luminaria.



Elaborado por: Hilda Moreno

Se ingresa los datos área a iluminar, luego se calcula el número y tipo de luminaria a ser colocado, el siguiente paso es distribuir la luz de forma uniforme sobre el área, y finalmente calcular el flujo luminoso total, si el flujo total es el correcto se termina el proceso, de lo contrario vuelve al inicio disminuyendo y/o aumentando la potencia o cambiando de lámpara hasta llegar al nivel de luz que se requiere. En este caso se colocó un tubo led de color blanco, de 30W y 830 lúmenes Figura 3.10.

Figura 3.10 Sistema de iluminación.

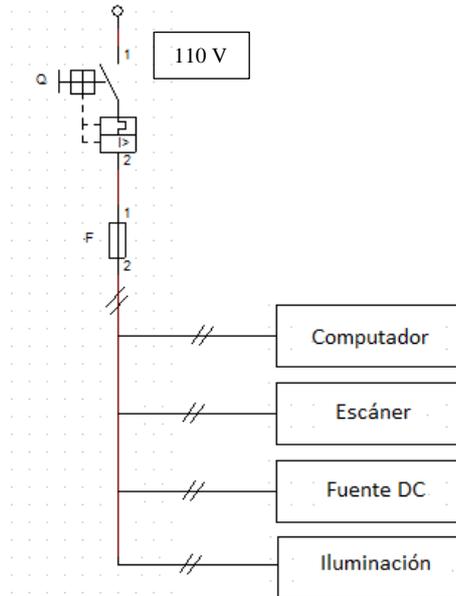


Elaborado por: Wilson Ayo

3.3.5 Diagrama cableado eléctrico

La alimentación de energía se realiza desde la red eléctrica de 110V AC, está protegida por un disyuntor de 30 A, para resguardar los dispositivos electrónicos Figura 3.11.

Figura 3.11 Diagrama eléctrico unifilar.



Elaborado por: Wilson Ayo

3.3.6 Hardware sistema de inspección por visión artificial

En la Figura 3.12 se puede apreciar las partes que integran el hardware del sistema de inspección automático de fallos en PCBs, por visión artificial, incluye el hardware para la adquisición de imágenes, procesamiento digital, visualización de errores, y sistema de desplazamiento.

Figura 3.12 Hardware del sistema de inspección por visión artificial.



Elaborado por: Wilson Ayo

3.4 DESCRIPCIÓN DEL SOFTWARE

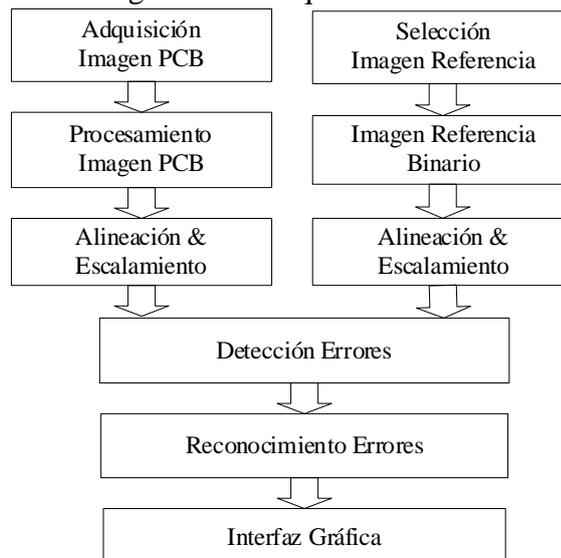
En la segunda parte del capítulo 3, se realiza la descripción, del software utilizado, así como el diseño y desarrollo del software del sistema para la detección automática de fallos en PCB's.

El software está desarrollado en lenguaje de programación Python, ejecutado por un computador, cuenta con librerías como OpenCV, que es una librería de alto nivel, de visión por computador, gratuita, y es empleada para el procesamiento computacional de imágenes, contiene alrededor de 500 funciones, varias de estas funciones se implementan para el desarrollo de la programación y PYQT5, para crear la ventana de trabajo contiene herramientas y funciones GUI potentes, incluye Qt Designer que es un diseñador de interfaz gráfica.

3.5 DIAGRAMA DE BLOQUES

El diseño de software del sistema de inspección automático de PCBs, se ha dividido en 6 etapas Figura 3.13, la primera fase inicia con la adquisición y selección de las imágenes de entrada, luego se describe el tratamiento de estas imágenes hasta transformarlas a un formato binario, la siguiente fase es el análisis y reconocimiento de errores, y la etapa final es la visualización de los errores mediante una interfaz.

Figura 3.13 Diagrama de bloques estructura de software.



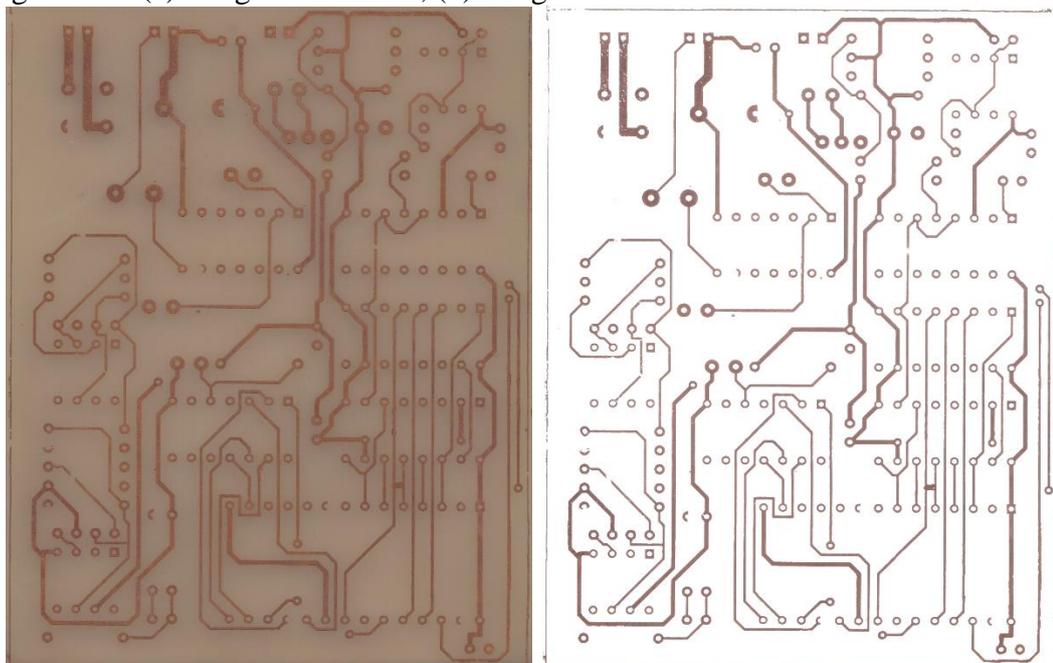
Elaborado por: Hilda Moreno

3.6 ADQUISICIÓN DE LA IMAGEN PCB

La primera etapa dentro de un proceso de visión por computador es la adquisición de la imagen a ser inspeccionada, constituye dos fases; inicialmente, la placa PCB es capturada por un sensor óptico, en este caso el escáner del dispositivo multifuncional Epson L355, con el que se obtiene información relativa de la placa PCB; en la segunda fase, se realiza la digitalización, donde se transforma en una imagen digital, lo que implica la transformación de datos analógicos a datos discretos. La imagen digital está constituida por un conjunto de píxeles, cada pixel brinda información acerca de las regiones y/o área específica de la imagen, y que puede posteriormente ser interpretada y manipulada por el ordenador.

En este proceso busca es garantizar la calidad de la imagen del PCB, por este motivo que es crucial el control de la intensidad lumínica en todas las zonas del PCB, la correcta configuración de los parámetros del escáner: 300dpi (resolución), 48bits (profundidad color), la selección apropiada del formato de compresión JPG (Joint Photographic Experts Group). La imagen del PCB digitalizada es exportada al ordenador mediante conexión wifi, y guardada en un espacio de memoria de este, para los procesos posteriores.

Figura 3.14 (a) Imagen escaneada, (b) Imagen escaneada con sistema de iluminación.

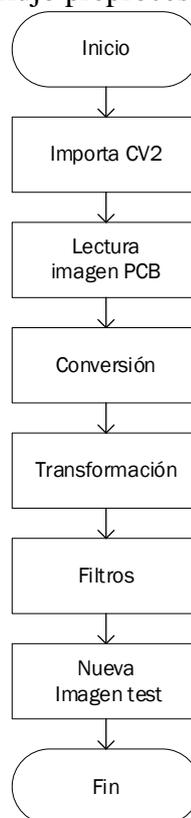


Elaborado por: Hilda Moreno

3.7 PREPROCESAMIENTO DIGITAL IMAGEN PCB

Extraída la imagen PCB y almacenada en un espacio de la memoria en el disco C:/ del computador, se somete a un preprocesamiento, explicado en la Figura 3.15, que contiene una sucesión de pasos, hasta transformarse en una imagen binaria; que inicia con la importación del módulo cv2, lectura de la imagen del PCB, conversión a escala de grises, transformación a una imagen binaria, y atenuación del ruido a través de filtros y suavizado de bordes. Todos estos pasos son desarrollados en la programación en Python y OpenCV, a través de un computador.

Figura 3.15 Diagrama de flujo preprocesamiento de la imagen PCB.



Elaborado por: Hilda Moreno

3.7.1 Importación módulo cv2

Las librerías de OpenCV tienen un sinnúmero de funciones, que son algoritmos desarrollados específicamente para visión artificial y que se utilizan para el desarrollo de la programación de este proyecto; para su funcionamiento únicamente se instala en el computador OpenCV, luego se importa el módulo cv2 y finalmente se llama a la función, para ejecutar la tarea que se requiera.

3.7.2 Lectura de la imagen PCB

Para la lectura de la imagen PCB se utiliza la función `cv2.imread()`, del módulo `cv2`, que recibe dos argumentos, la ruta de la carpeta en el cual se ubica la imagen a ser examinada, y el nombre, luego esta información se guarda en una variable. La imagen del PCB ahora es tratada como una matriz de filas y columnas.

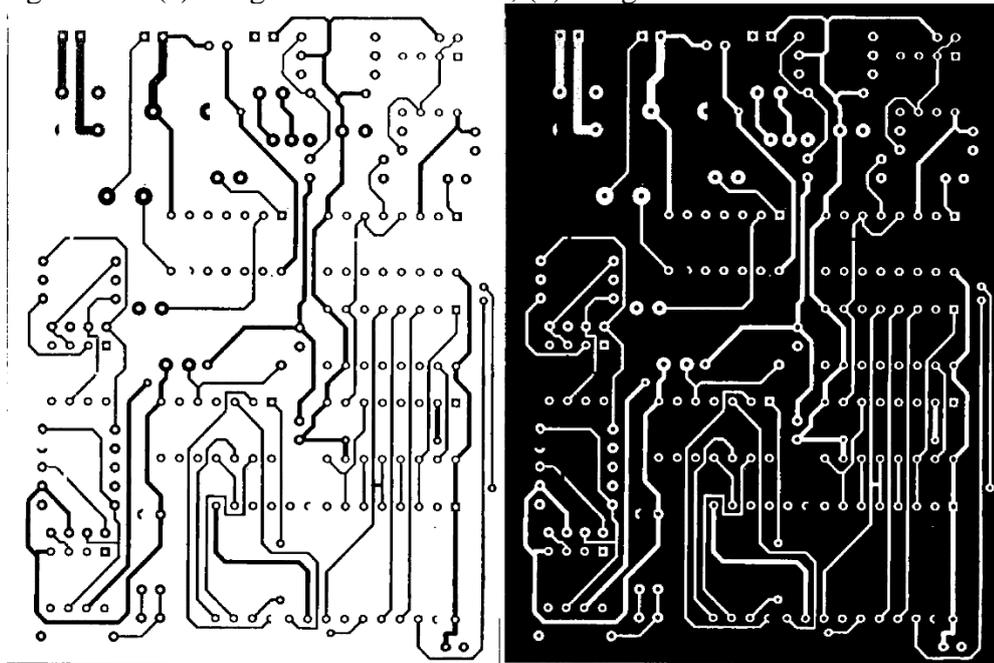
3.7.3 Conversión a escala de grises

Para la conversión a escala de grises, se emplea el método `cv2.cvtColor()`, del módulo `cv2`, que contiene la bandera `cv2.COLOR_RGB2GRAY` que convierte un espacio de color a otro, la imagen I_{PCB} , ahora está representada por un valor de 256 valores en pixeles es decir de 0 a 255: donde 0 representa un pixel de color negro y 255 representa un pixel de color blanco.

3.7.4 Transformación a imagen binaria

Para transformar a binario Figura 3.16 (a), como condición inicial el argumento de entrada debe ser una imagen en escala de grises, utilizando la función `cv2.threshold()`, que establece el umbral, es decir que se le asigna un valor que puede ser únicamente en dos formatos de pixel 0 o 255, en este caso se establece el umbral en \geq que 220 pixeles corresponde a 0 y $<$ que 220 pixeles corresponde a 255.

Figura 3.16 (a) Imagen del PCB binaria, (b) Imagen del PCB binaria invertida.

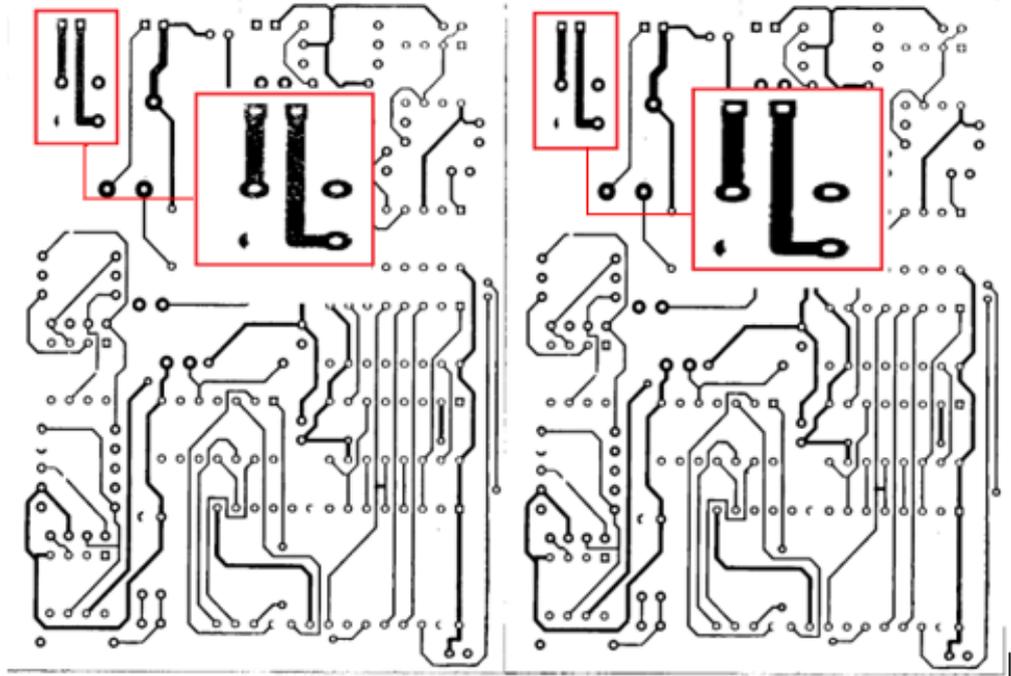


Elaborado por: Hilda Moreno

3.7.5 Atenuación del ruido

El ruido se genera principalmente en la adquisición de la imagen Figura 3.17 (a) específicamente en la captura de la imagen y representa un gran inconveniente, que disminuye la calidad de la imagen, para la atenuación del ruido, se utiliza operaciones morfológicas, mediante la utilización de funciones como: `cv2.erode()`, `cv2.dilate()`, `cv2.morphologyEx()`, que sirven para eliminar píxeles que no pertenecen a los objetos de interés, tal cual se muestra en la Figura 3.17 (b).

Figura 3.17 (a) Imagen del PCB binaria, (b) Imagen del PCB binaria filtrada.



Elaborado por: Hilda Moreno

Una vez concluida la etapa de preprocesamiento en la imagen del PCB (I_{PCB}), se consigue como resultado una nueva imagen de prueba (I_{Test}).

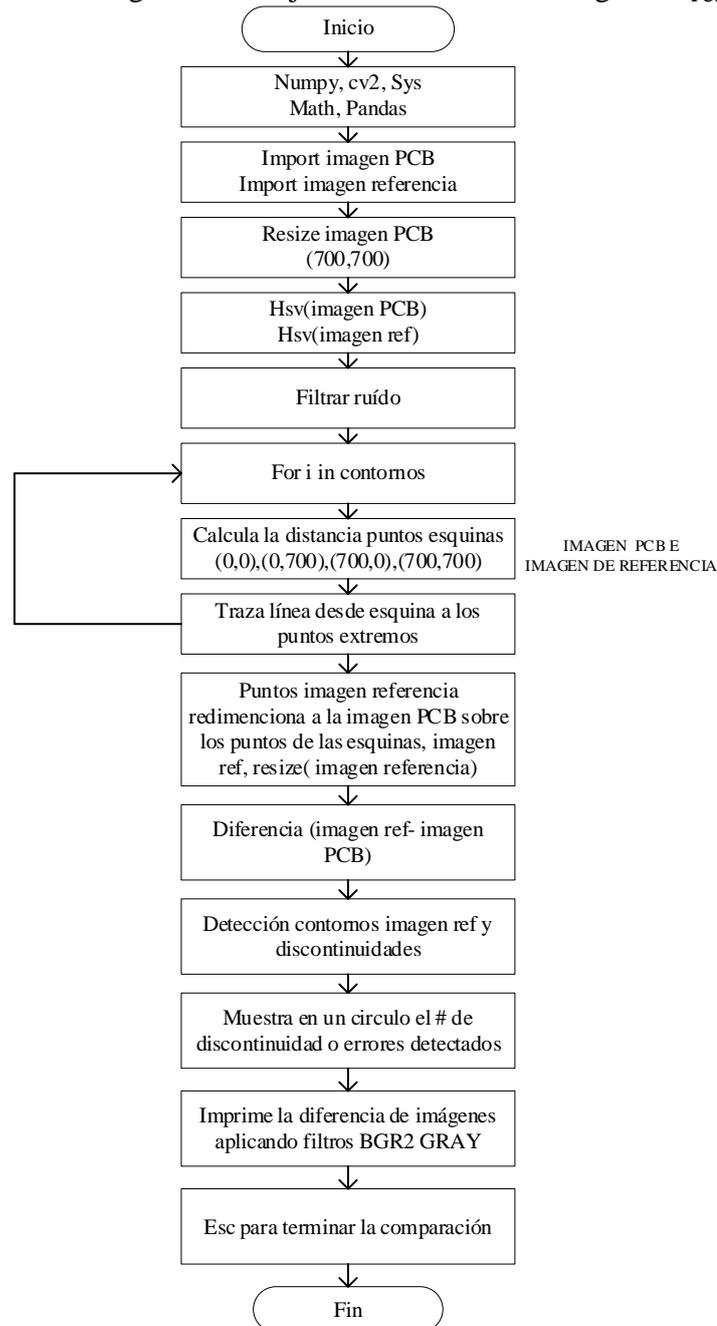
3.8 SELECCIÓN IMAGEN REFERENCIA

Para la selección de la imagen de referencia, no se requiere del mismo procesamiento que se realizó en la imagen del PCB de prueba, únicamente se exporta el archivo del circuito desde el programa Proteus, en formato de imagen, monocromático y con una resolución de 400dpi, luego se transforma a una imagen binaria, finalmente se guarda colocando un nombre, en este caso la denominaremos imagen de referencia (I_{Ref}).

3.9 ALINEACIÓN DE IMÁGENES

La etapa de la alineación de las imágenes es sensible al grado desfase que exista entre las imágenes a ser comparadas, y puede ocasionar inconvenientes, el algoritmo, como se manifiesta en la Figura 3.18, se fundamenta en el cálculo de la distancia de las esquinas de los extremos hacia el objeto más cercano, luego redimensiona a la imagen de referencia.

Figura 3.18 Diagrama de flujo alineación de las imágenes I_{Test} e I_{Ref} .

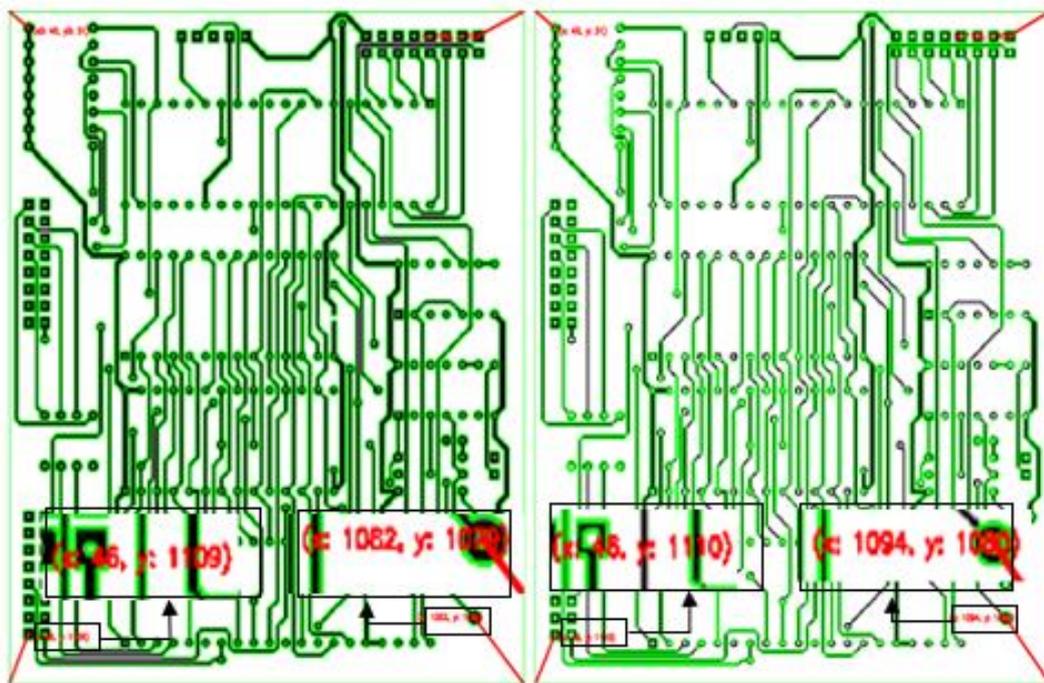


Elaborado por: Hilda Moreno

El primer paso es la importación de las librerías, luego se carga las imágenes de referencia y test, transformadas a binario, es importante la colocación de una guía o un recuadro alrededor de las imágenes para tener una referencia. Una vez realizada las configuraciones necesarias, se redimensiona a la imagen de test con la función `cv2.resize()`, que toma, el nombre de la imagen y las dimensiones de (ancho x alto), en este caso (700,700).

Se efectúa el análisis de contornos para detectar los objetos que se encuentren en los extremos de las esquinas de la imagen de referencia, a continuación, se traza una línea y se calcula la distancia desde los puntos de las esquinas (0,0), (0, 700), (700,0) y (700,700), hacia los centros de los objetos detectados en los extremos, esta misma acción se realiza para la imagen de referencia, de esta manera se redimensiona, con las dimensiones de la imagen de test en la imagen de referencia.

Figura 3.19 Alineación de las imágenes, (a) I_{Ref} , (b) I_{Test}



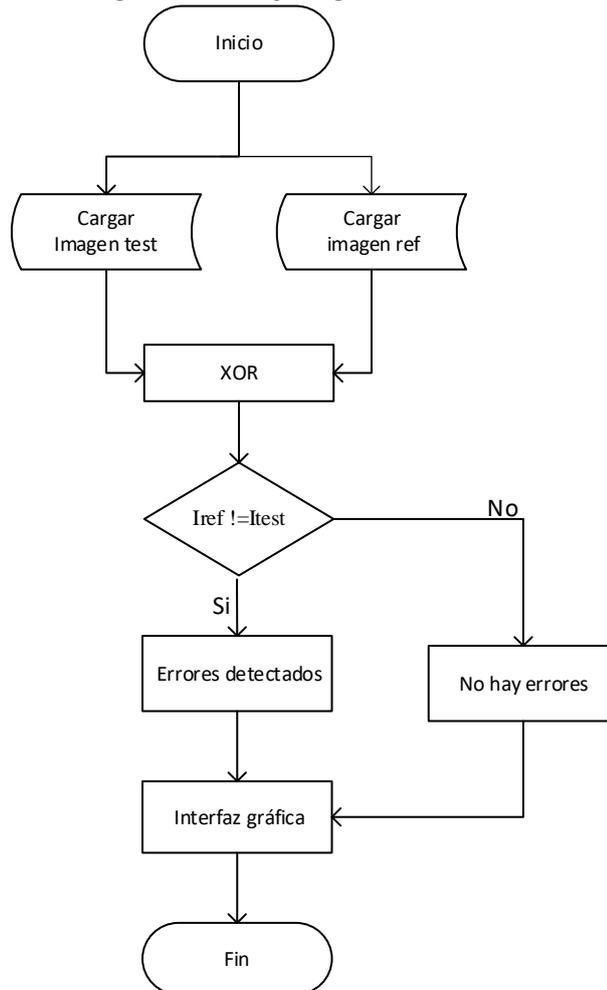
Elaborado por: Hilda Moreno

En la Figura 3.19 se observa la comparación de las coordenadas de los extremos de las esquinas inferiores hacia el centro de los objetos, en este caso; los pads más cercanos de la imagen de prueba, con respecto a la imagen de referencia.

3.10 ALGORITMO DETECCIÓN DE FALLOS EN PCB

El algoritmo consiste en una operación lógica XOR entre las imágenes de test y la imagen de referencia como resultado de esta combinación aritmética se obtiene los defectos del PCB, la Figura 3.20 resume la programación del algoritmo en el siguiente diagrama de flujo.

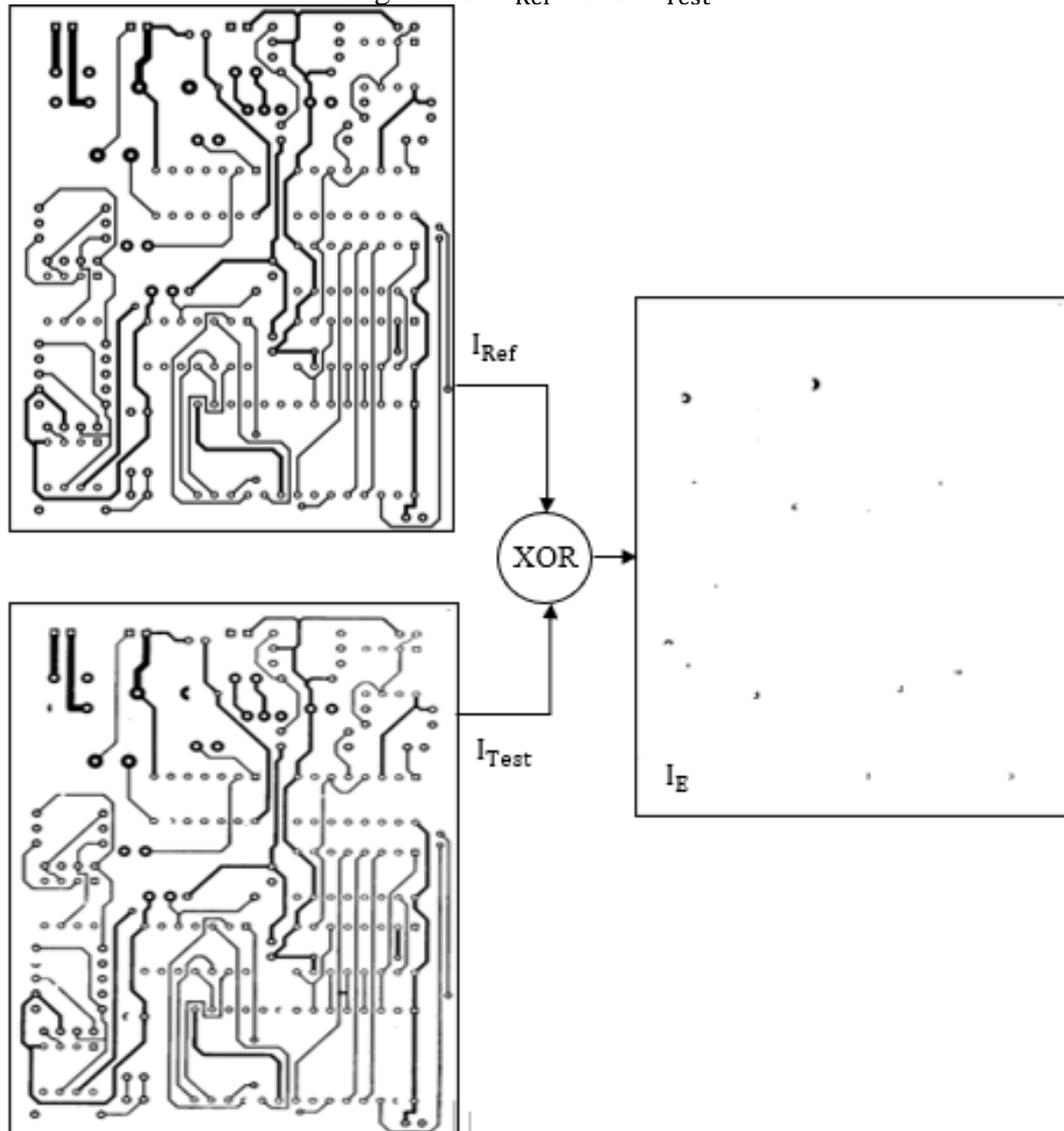
Figura 3.20 Diagrama de flujo algoritmo detección de errores.



Elaborado por: Hilda Moreno

El primer paso; es importar la imagen I_{Ref} e imagen I_{Test} ambas imágenes transformadas a binario, alineadas y escaladas previamente, luego se compara pixel a pixel con las mismas coordenadas (x, y) entre imagen I_{Ref} e imagen I_{Test} , mediante el operador lógico XOR, este operador toma a la salida el valor de falso o 0 cuando ambas entradas poseen el mismo valor, mientras que es verdadero o 1 en los demás casos. La acción se ejecuta con el método `cv2.bitwise_xor`, como resultado obtenemos una nueva imagen denominada: imagen error (I_E).

Figura 3.21 I_{Ref} XOR I_{Test} .



Elaborado por: Hilda Moreno

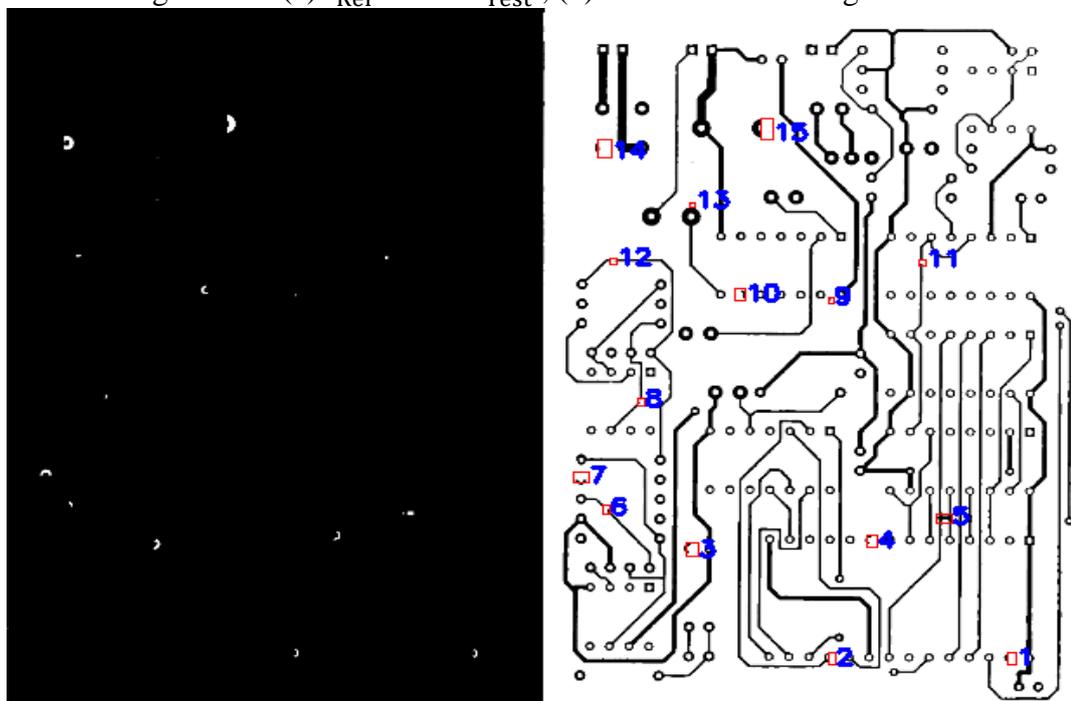
Al compararse las imágenes I_{Ref} e I_{Test} se observa en la Figura 3.21 el resultado de la operación lógica XOR, $I_E(x, y) = I_{Ref}(x, y) \text{ XOR } I_{Test}(x, y)$, donde se detectaron 15 objetos, que pertenecen a los errores de la imagen I_{Test} con respecto a la imagen I_{Ref} .

Para la presentación de los errores, lo primero que se realiza es la filtración de la imagen, mediante operaciones morfológicas, esto con la finalidad de mejorar la imagen, luego se realiza el análisis de contornos, por medio de la función `cv2.findContours()`, que recoge tres argumentos, nombre de la imagen, el modo de recuperación de contornos, con la bandera `RETR_EXTERNAL` recupera solo los contornos externos y el tercer argumento es el modo de aproximación, que traza los

puntos para la construcción del contorno, en este caso se utiliza la bandera `CHAIN_APPROX_SIMPLE` toma solo los puntos estrictamente necesarios.

Detectados los objetos Figura 3.22 (a), se bordean con un rectángulo, inicialmente se define el área de cada uno de los objetos, con la función `cv2.contourArea()`, luego se guarda las dimensiones del rectángulo de cada objeto con la función `cv2.boundingRect()`, y finalmente se dibuja el rectángulo, con la función `cv2.rectangle()`. Para colocar la numeración respectiva de cada defecto, se lo realiza con la función `cv2.putText()`, Figura 3.22(b).

Figura 3.22 (a) $I_{Ref} \text{ XOR } I_{Test}$, (b) Defectos en la imagen PCB.



Elaborado por: Hilda Moreno

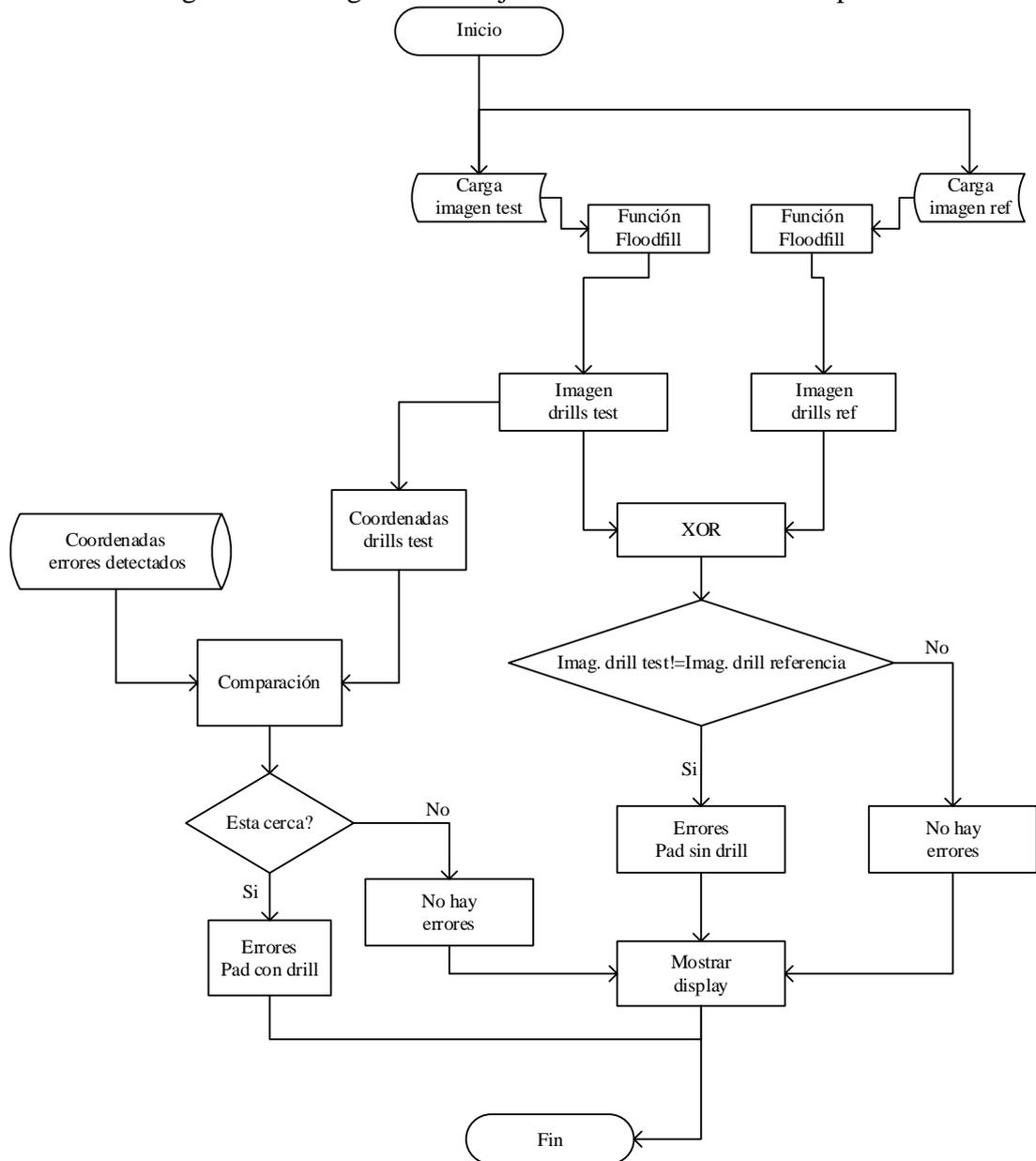
3.11 RECONOCIMIENTO DE ERRORES

Para el reconocimiento de errores se desarrollan algoritmos individuales y diferentes para cada error, que, en base a un análisis estadístico, realizado por la empresa para fines de este proyecto, son los más recurrentes en la fabricación de PCB's.

3.11.1 Daño en el pad

Para el desarrollo del programa se divide en dos partes: daños en pads sin drill (sin huecos) y daños en pads con drill (con huecos), en la Figura 3.23 se detalla la programación de las dos partes del reconocimiento del error daño en los pads.

Figura 3.23 Diagrama de flujo reconocimiento del error pad.

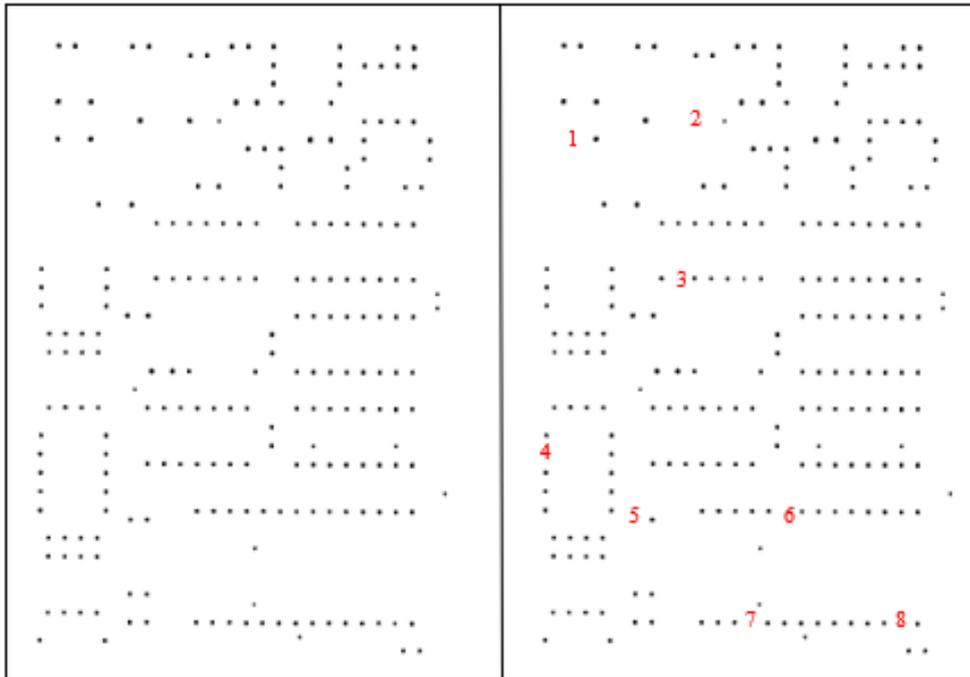


Elaborado por: Hilda Moreno

Primera parte del algoritmo: se carga las imágenes I_{Ref} e I_{Test} ambas imágenes transformadas a binario, alineadas y escaladas, estas imágenes ahora son tratadas como arreglos NumPy dentro una variable, la información digital de estas matrices se emplea como datos de entrada al sistema.

El siguiente paso es emplear en las dos imágenes el método `cv2.floodFill()`, que rastrea el área de todos los huecos (drills) y la rellena de color negro, todo lo demás se elimina, tal cual se observa en la Figura 3.24.

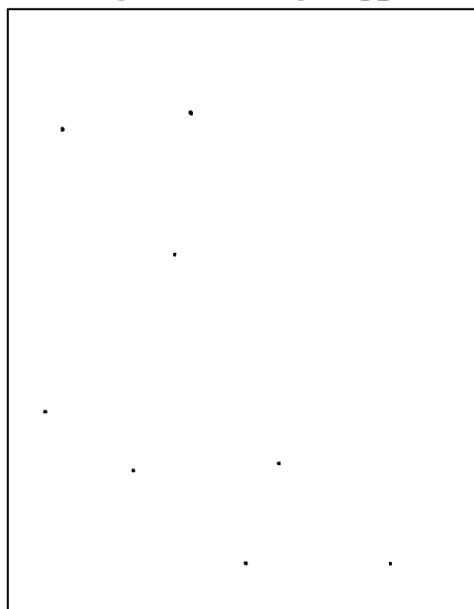
Figura 3.24 (a) Imagen drills de referencia, (b) Imagen drills de test.



Elaborado por: Hilda Moreno

Obtenidas las nuevas imágenes: la imagen drills de referencia (I_{DRef}) e imagen drills de test (I_{DTest}), estas se comparan con la función `cv2.Absdiff()`, este método resta una imagen de otra eliminando todos los objetos que no cambian mientras preserva los que sí, como resultado se obtiene una nueva imagen denominada: imagen drills error (I_{DE}), en la Figura 3.25 se observa los errores relacionados con la pérdida de drill.

Figura 3.25 Imagen I_{DE} .

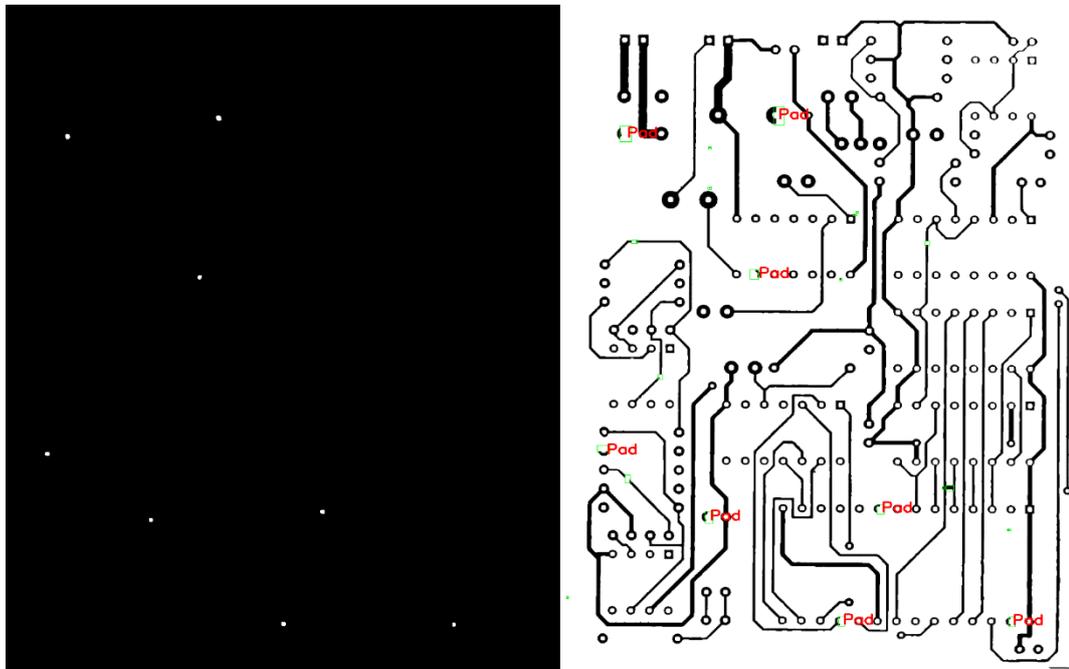


Elaborado por: Hilda Moreno

La segunda parte es el algoritmo para el reconocimiento de errores sin pérdida de drill se compara los datos de las coordenadas de los errores, que fueron guardados en una lista en el proceso anterior y la lista de las coordenadas de los drills de la imagen de test, si el resultado de la comparación de estas dos listas, está dentro del valor mínimo de distancia que se calcula en base al tamaño de cada pad entonces se establece que es un error sin pérdida de drill, en este caso no existen, todos los pads de esta placa están relacionados a daños con pérdida drill.

Para la presentación de los errores daños en pads se coloca el texto “Pad” de color rojo y lo traspasamos a la imagen de entrada imagen I_{Test} , como se evidencia en la Figura 3.26 (b).

Figura 3.26 (a) Contornos pads, (b) Error daño en pads.

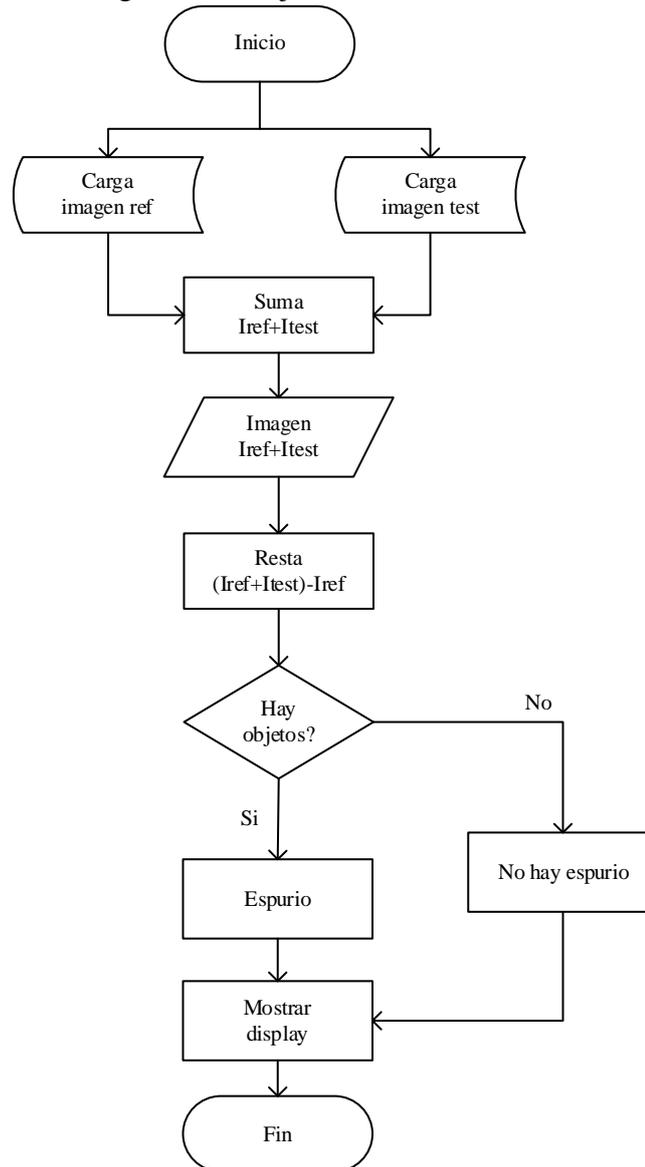


Elaborado por: Hilda Moreno

3.11.2 Espurio

Para el reconocimiento del error por exceso de cobre o espurio, se desarrolla un algoritmo en que el que se realiza dos operaciones matriciales Figura 3.27.

Figura 3.27 Diagrama de flujo, reconocimiento del error espurio.



Elaborado por: Hilda Moreno

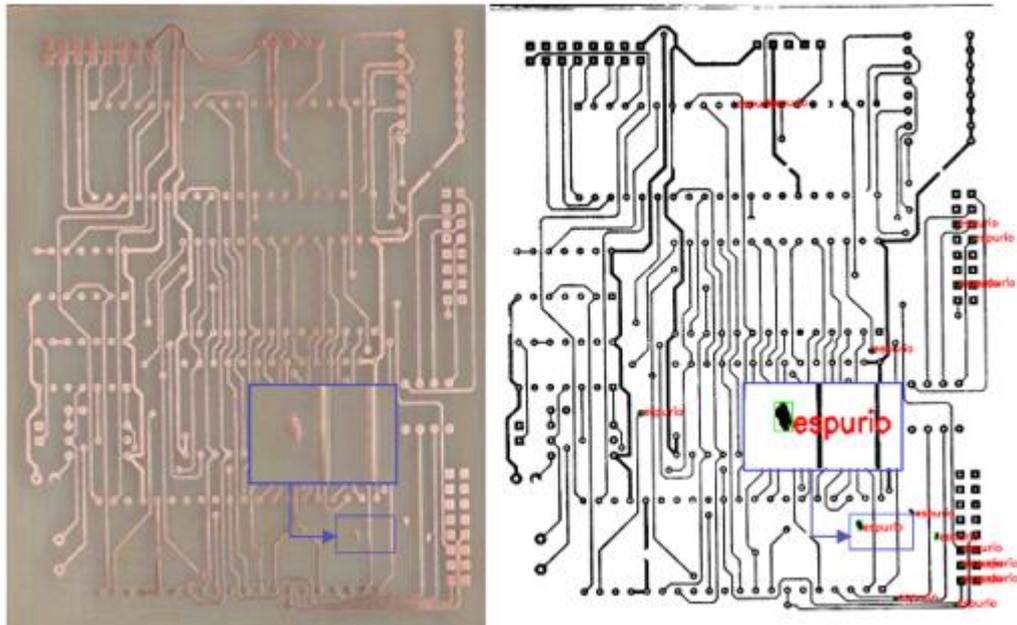
El primer paso es cargar las imágenes de test y referencia, que son los datos de entrada al proceso estas dos imágenes se suman con el método para combinar objetos de dos imágenes en una, utilizando el operador lógico OR que funciona como verdadero o 1 si cualquiera de los elementos es verdadero, de lo contrario es falso o 0, se obtiene como resultado una nueva imagen, luego se resta la nueva imagen resultado de la suma entre I_{Ref} e I_{Test} y la imagen I_{Ref} , con la función `cv2.Absdiff()` y se obtiene los errores por exceso de cobre Figura 3.28, se puede utilizar el análisis detector de bordes canny con OpenCV y Python.

Figura 3.28 Contorno espurio canny.



Elaborado por: Hilda Moreno

Figura 3.29 (a) Espurio PCB, (b) Espurio imagen I_{Test} .

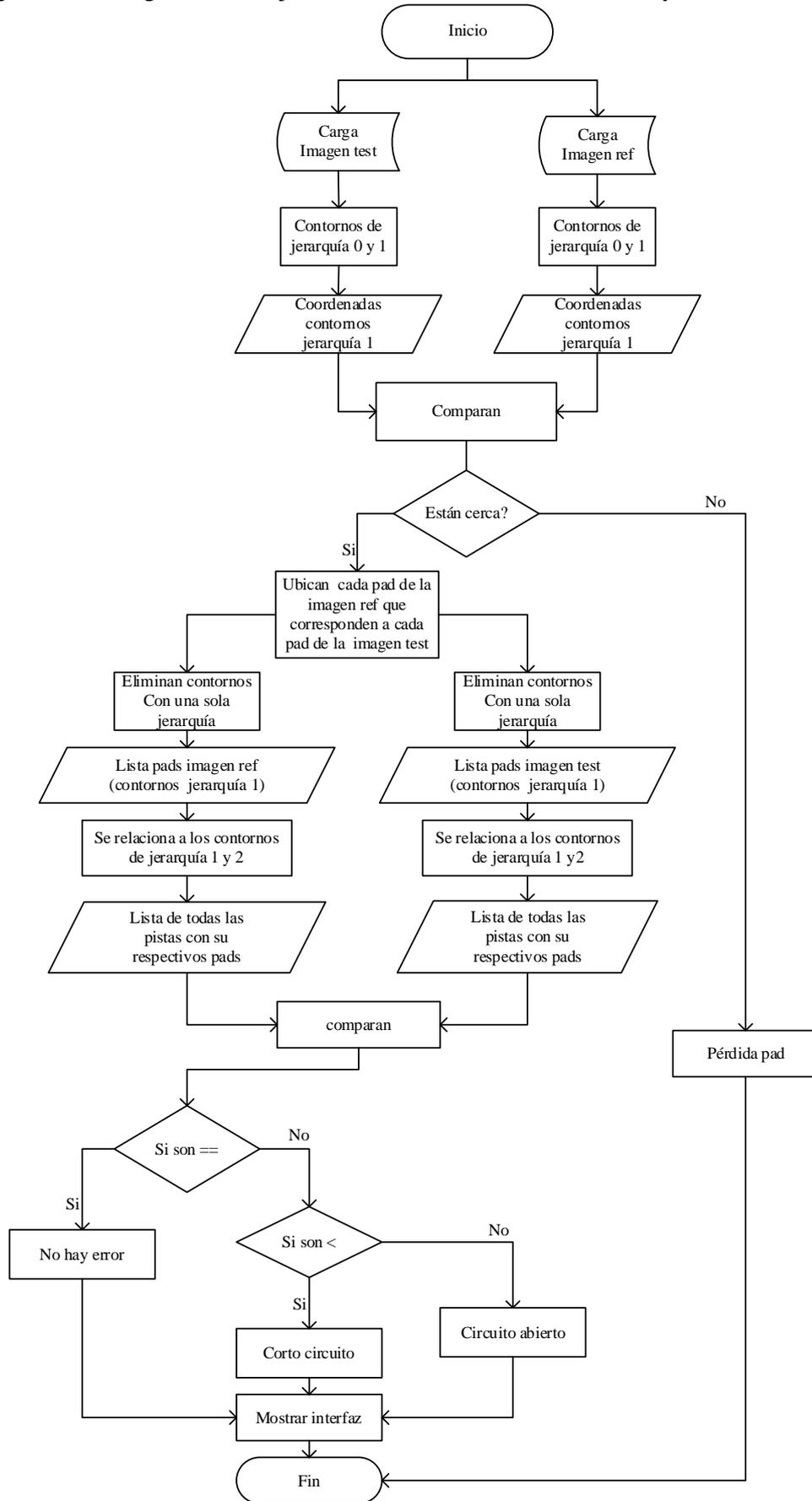


Elaborado por: Hilda Moreno

3.11.3 Circuito abierto y cortocircuito

Para el desarrollo del algoritmo de reconocimiento de errores: cortocircuito y circuito abierto, se utiliza el método de contornos para detectar objetos de una figura anidada hierarchy, en la Figura 3.30, se describe el diagrama de flujo del reconocimiento de los errores de: cortocircuito y circuito abierto.

Figura 3.30 Diagrama de flujo reconocimiento de cortocircuito y circuito abierto.



Elaborado por: Hilda Moreno

Para el desarrollo de la programación se importa las imágenes de referencia y test, el siguiente proceso; es realizar un análisis de contornos, se emplea la función `cv2.findContours()` y mediante la bandera `cv2.RETR_TREE`, se recupera todos los contornos con sus jerarquías, en este caso se utiliza la jerarquía 0 (padre) y 1 (hijo), OpenCV lo representa como una matriz de cuatro valores: [Siguiente, Anterior, First_Child, Parent].

En la Tabla 3.8 se explica la matriz de contornos hierarchy.

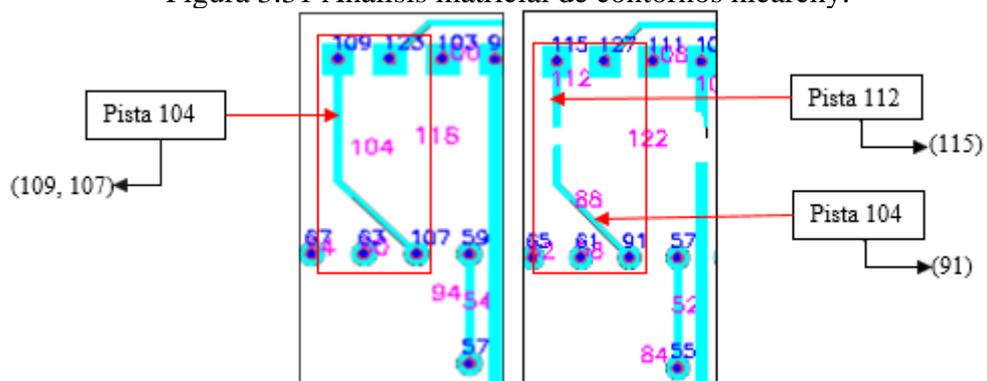
Tabla 3.7 Matriz contornos hierarchy.

Siguiente	Anterior	Hijo	Padre	Término
40	-1	39	37	Siguiente: contorno (mismo nivel jerárquico)
42	38	41	37	Anterior: Contorno (mismo nivel jerárquico)
-1	40	43	37	Hijo: Su primer contorno hijo
48	-1	47	45	Padre: Contorno principal
-1	46	49	45	Menos uno (-1): No existe

Elaborado por: Hilda Moreno

Se ha extraído un extracto de la matriz de contornos para explicar el funcionamiento: existen dos padres el 37 y 45, 37 padre tiene tres hijos que son 39, 41, 43 y 45 padre que tiene dos hijos 47 y 49, lo que significa que hay dos pistas y dentro de cada pista existen 3 y 2 drill respectivamente, de esta forma se puede establecer la relación entre la pista y el drill.

Figura 3.31 Análisis matricial de contornos hierarchy.

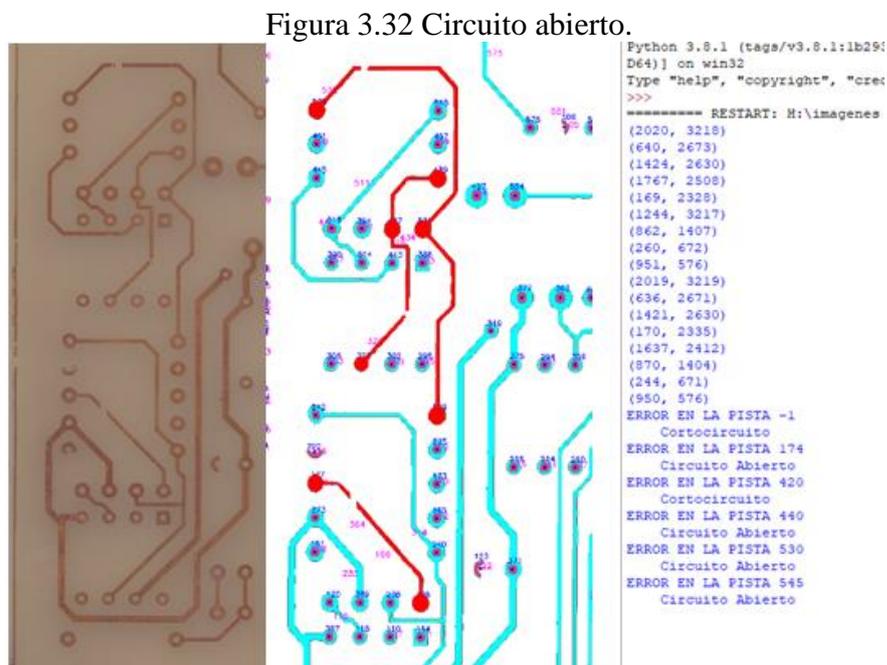


Elaborado por: Hilda Moreno

En la imagen de la Figura 3.31 se puede apreciar el análisis matricial de contornos hierarchy, se toma el contorno extremo (pista), en la imagen de referencia es la 104 tiene dos contornos internos que son 109 y 107 mientras que, en la imagen de test, esta

misma pista se han separado formando dos pistas 112 y 88 cada una con un drill 115 y 91 respectivamente, significa que la pista está cortada, este análisis se aplica también para el error de cortocircuito, pero de forma contrapuesta.

El análisis matricial de contornos se realiza en ambas imágenes: de test y referencia, este método solo se va a establecer en pistas de mínimo dos drills, con las listas depuradas e identificados cada elemento correctamente con respecto a la imagen de referencia, basándose en la ubicación de los centros de las coordenadas de los drills, se compara las listas, si se encuentra menos drills dentro de las pistas de la imagen de test con respecto a la imagen de referencia representa un cortocircuito y si se encuentra más drills corresponde a un circuito abierto, Figura 3.32.



Elaborado por: Hilda Moreno

3.12 INTERFAZ GRÁFICA

La GUI se creó utilizando widgets del marco PyQt5 y el diseño se creó en Qt Designer, la ventana de trabajo es fácil de utilizar, permite interactuar fácilmente como una extensión de los programas de detección y reconocimiento de defectos de Python.

3.12.1 Implementación

La interfaz gráfica Figura 3.33, está compuesta por los siguientes elementos que se detalla a continuación: panel de control [1], área de visualización [2], tabla de estado [3], barra de menús [4], cuadro de texto [5] y líneas texto [6].

Figura 3.33 Ventana de trabajo.



Elaborado por: Hilda Moreno

Se importa las librerías PYQT5, que se muestran en la siguiente línea de código.

```
from PyQt5.QtWidgets import QWidget, QProgressBar, QPushButton, QApplication
from PyQt5 import QtCore, QtGui, QtWidgets, QtPrintSupport; luego se llaman a los
métodos para crear la ventana principal, de la misma manera como se realiza la
programación para detección y reconocimiento de errores.
```

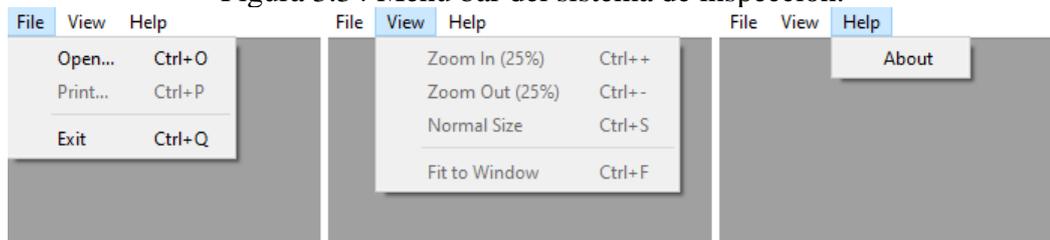
La clase principal que proporciona el marco o esqueleto es: QMainWindow, es la plantilla más grande en la que se coloca el visor de imágenes, menú bar, tabla de estado y otros widgets. Para la creación de la ventana principal es necesario identificar las clases principales que son: QGraphicsView, QMenuBar, y QDockWidget estas clases representan los pilares fundamentales en las que añadiremos todos los elementos.

QGraphicsView: Es el área donde se muestra la imagen a ser examinada, para configurar el rectángulo del área, se ejecuta mediante la función setSceneRect(), que

se ajustan al tamaño de la imagen de prueba, mediante barras desplazables una horizontal y otra vertical .

QMenuBar: Para crear el menú Figura 3.34, de la ventana principal y que contiene las opciones File, View, y Help, se agrega menús llamando al método `menuBar()`, luego se añade el menú con `addMenu()`, finalmente para vincular cada botón de menú se asocia con el objeto `QAction`, cuenta con atajos.

Figura 3.34 Menú bar del sistema de inspección.



Elaborado por: Hilda Moreno

QDockWidget: Se encuentran: tablas de estado, botones, líneas y cuadros de texto.

- **QTable:** Para añadir la tabla en la que se visualiza los errores, con su respectiva ubicación en el sistema de coordenadas (x, y), se invoca al constructor `QTableWidget`, luego se crea las funciones y propiedades para establecer el número de filas y columnas y demás parámetros como el encabezado vertical.
- **QPushButton:** Los botones Push-Button se llama al constructor `QPushButton` que crea el widget, luego se coloca los parámetros: nombre, tamaño, posición, color, finalmente se vincula a la función que va a ejecutar. Para esta aplicación se crea los siguientes botones:
 - Start: Iniciar captura
 - Check: Iniciar la inspección
 - Clasificación: Ver errores: espurio, pads, cortocircuito y circuito abierto
 - Exit: Salir de la aplicación
- **QLineEdit:** Es el campo donde se muestra las dimensiones del área del PCB. El widget se llama `QLineEdit` y tiene los métodos `setText()` para establecer el valor del cuadro de texto y `text()` para obtener el valor. Podemos establecer el tamaño del cuadro de texto usando el método `resize()`. La posición se puede establecer usando el método `move(x,y)`.

CAPÍTULO 4

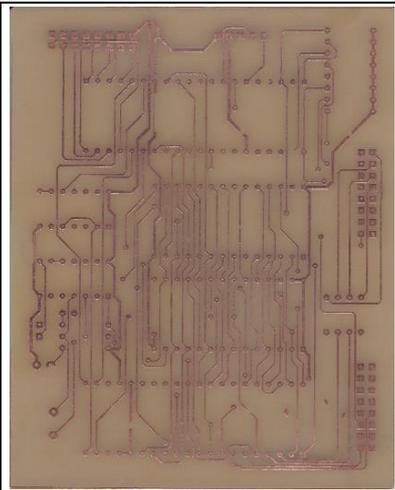
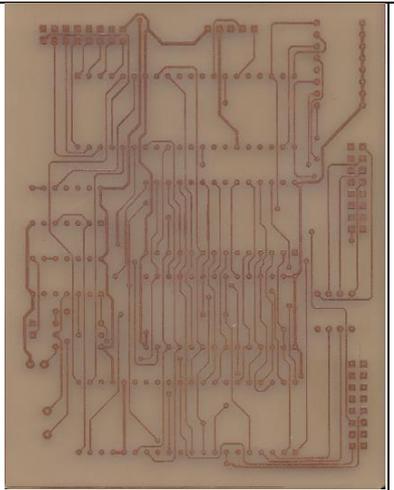
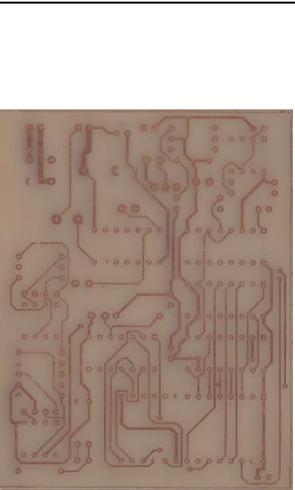
PRUEBAS Y FUNCIONAMIENTO

En este capítulo se analiza las pruebas de funcionamiento y resultados del prototipo del sistema de inspección automático por visión artificial para detección de fallos en PCBs.

4.1 FABRICACIÓN PLACAS PCB

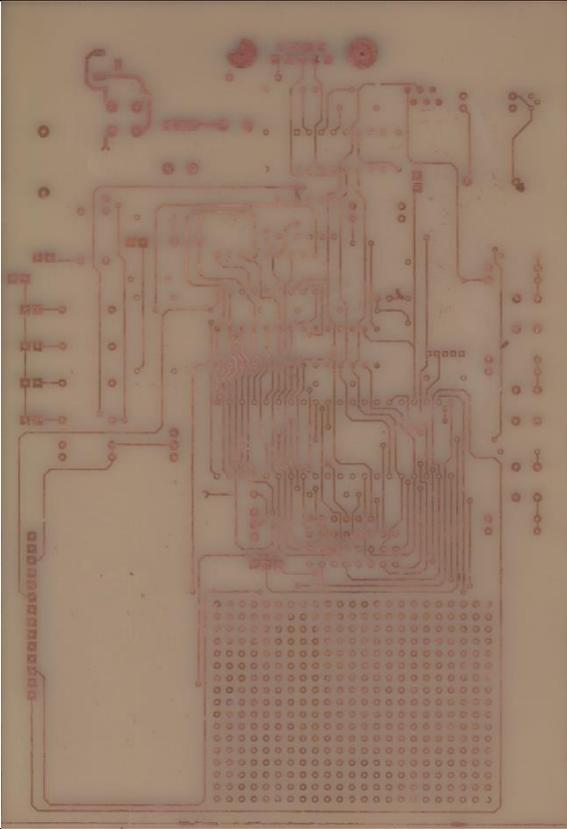
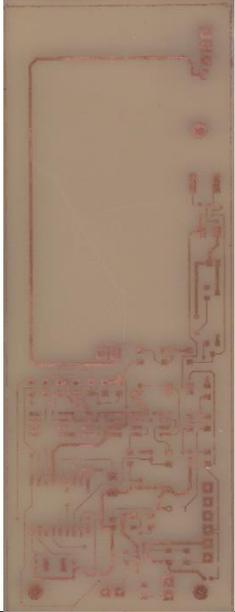
Para fines de este proyecto se han fabricado 8 placas de cobre de diferentes tamaños y densidades, para establecer la densidad se basó en umbrales: < 15 densidad baja, >16 & >25 densidad media y > 26 densidad alta, verificadas en la Tabla 4.1 y Tabla 4.2, las placas PCB han sido elaboradas por la técnica del planchado perfeccionado, que consiste en el traspaso térmico del tóner de un acetato impreso (placa PCB en papel couche) a una placa de cobre (baquelita fibra de vidrio), y luego colocar al ácido férrico. El material del PCB es fenólico y fibra de vidrio, los errores fueron intencionales e involuntarios; para colocar los errores, se basa en los datos estadísticos de una investigación previa durante la fabricación manual de los circuitos impresos de la empresa DINNOVASOFT, presentada en la justificación, donde se determina que los errores recurrentes son: espurio, daño en los pads, cortocircuito y circuito abierto.

Tabla 4.1 PCBs 7, 8, 2 a ser examinados.

PCB 7	PCB 8	PCB 2
		
Área: 12,23x9,62 cm	Área: 12,2x9,6 cm	Área: 8.65x11.08 cm
Densidad: alta	Densidad: alta	Densidad: media
Grosor Pista: 1mm, 0.5mm	Grosor Pista: 1mm, 0.5mm	Grosor Pista: 1mm, 0.5mm
Errores: 27	Errores: 2	Errores: 16

Elaborado por: Hilda Moreno

Tabla 4.2 PCBs 5, 6 a ser examinados

	
PCB 5	PCB 6
Área: 17.64X12.11	Área: 11.48x4.49
Densidad: alta	Densidad: media
Grosor Pista: 1mm, 0.5mm	Grosor Pista: 1mm, 0.5mm
Errores: 22	Errores: 13

Elaborado por: Hilda Moreno

4.2 PRUEBA DE INSPECCIÓN AUTOMÁTICA POR VISIÓN ARTIFICIAL

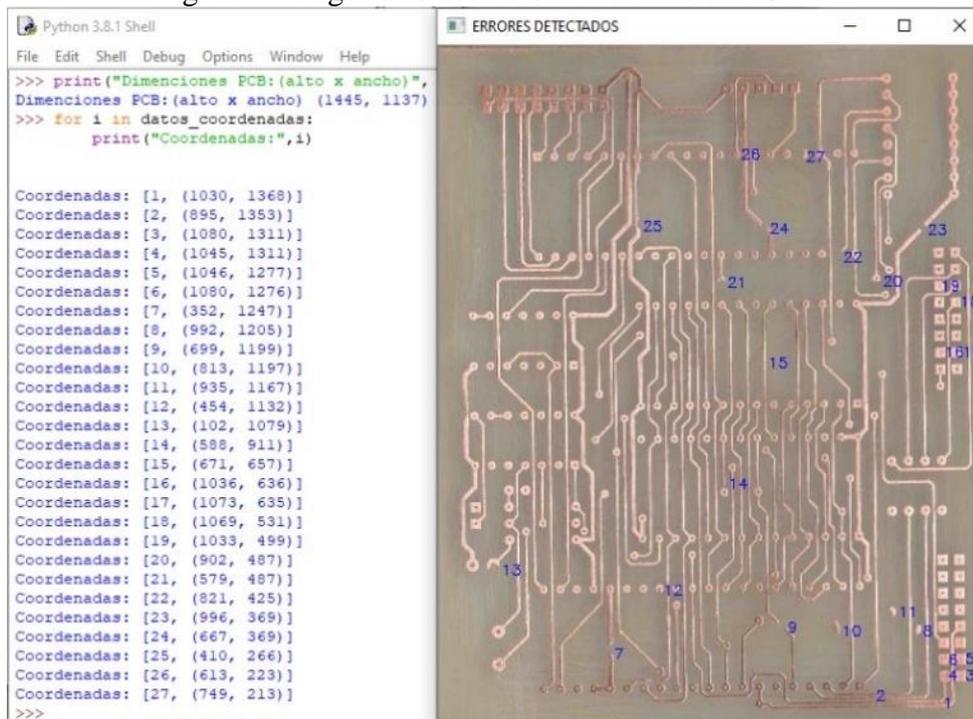
Esta prueba consiste en realizar la inspección en 8 placas PCB, utilizando el sistema de inspección automático por visión artificial (I.V.A.), para esta prueba la persona encargada de examinar las placas, no necesita tener conocimientos avanzados, simplemente seguir las instrucciones del funcionamiento de la aplicación del “Funcionamiento Sistema De Inspección Por Visión Artificial”.

Para la verificación del funcionamiento del sistema y para fines de una mejor explicación en esta parte, se ha considerado dos circuitos impresos PCB; una placa tiene errores intencionales y la otra tiene errores involuntarios.

En la Figura 4.1 se puede apreciar el PCB en este caso con errores que se han colocado intencionalmente, alado se ha impreso los defectos encontrados en el PCB por el algoritmo de detección de errores que se encuentran guardados en una variable.

EL algoritmo detecta los errores desde la parte inferior hacia la superior, desde el número 1 hasta el número de defectos que encuentre en este caso 27 defectos.

Figura 4.1 Algoritmo detección de errores en PCB 7.



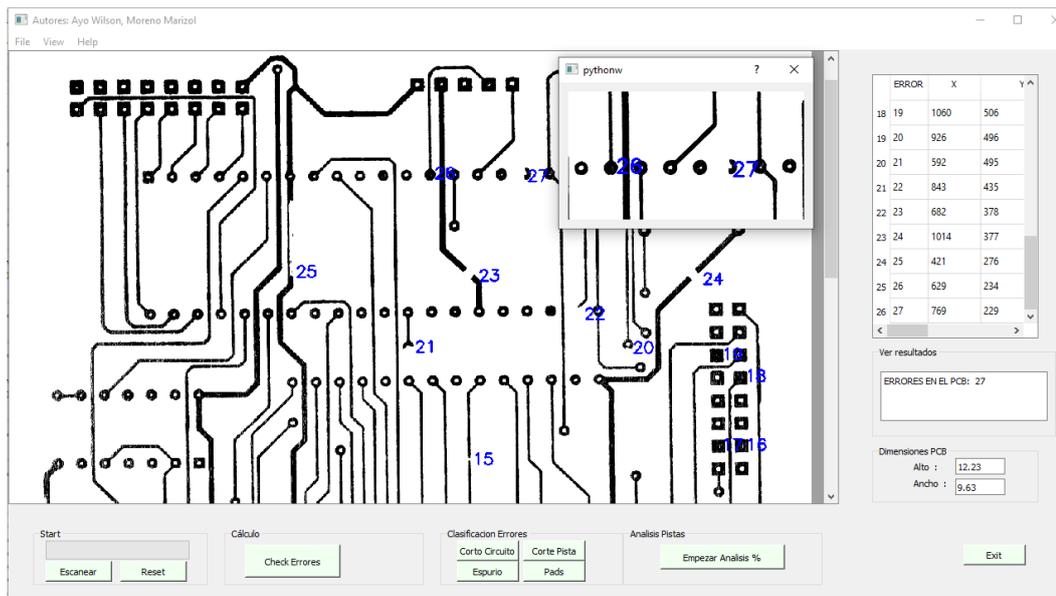
Elaborado por: Hilda Moreno

El la Figura 4.2 se observa el funcionamiento del sistema de inspección por visión artificial que ha detectado 27 defectos, lo que indica que esta placa tiene errores graves para el funcionamiento, se detalla cada defecto en la tabla de estado con su respectiva numeración y ubicación en el sistema de coordenadas (x, y), en la parte inferior en el cuadro de texto se resume el número total de los errores detectados y por último la dimensión del PCB, en este caso es de (1445x1137) pixeles (puntos/pulgada); para calcular el tamaño de la impresión sabiendo los puntos por pulgada se utiliza la fórmula: $(x * 2.54)/n$ Ec. 4.1

donde: x: pixeles impresos, n: puntos por pulgada (300ppp).

La dimensión del PCB es de (12,23x9,62) cm.

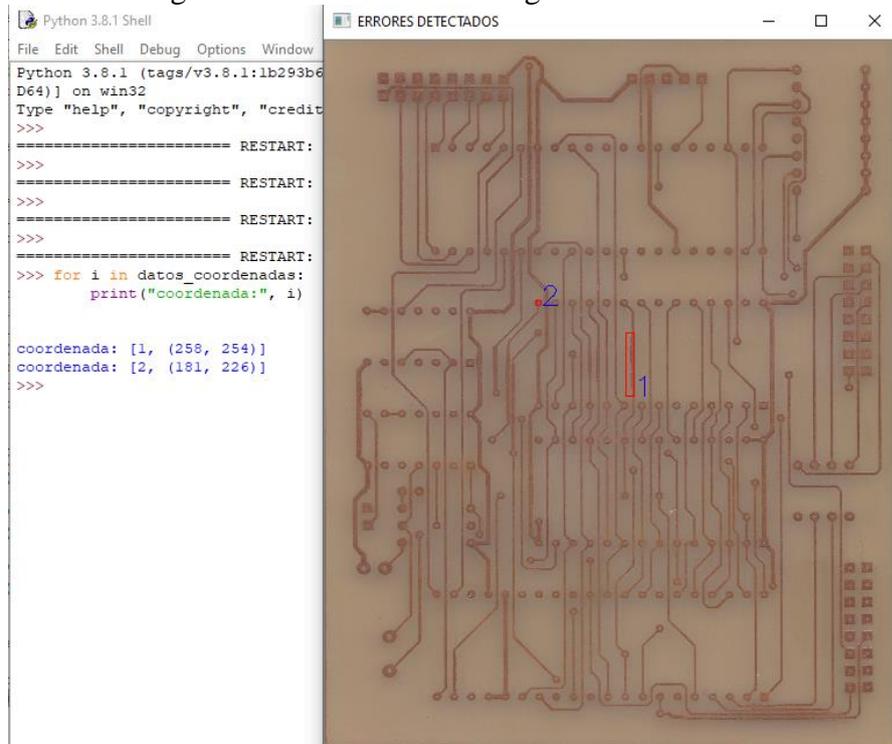
Figura 4.2 Funcionamiento aplicación para PCB 7.



Elaborado por: Hilda Moreno

De igual manera se realiza el mismo análisis en este caso del PCB sin errores, sin embargo y como se observa existen dos errores involuntarios, que no representan errores fatales para el funcionamiento del circuito un error corresponde a la pérdida de un drill en un pad y el otro error es espurio sobre la pista Figura 4.3.

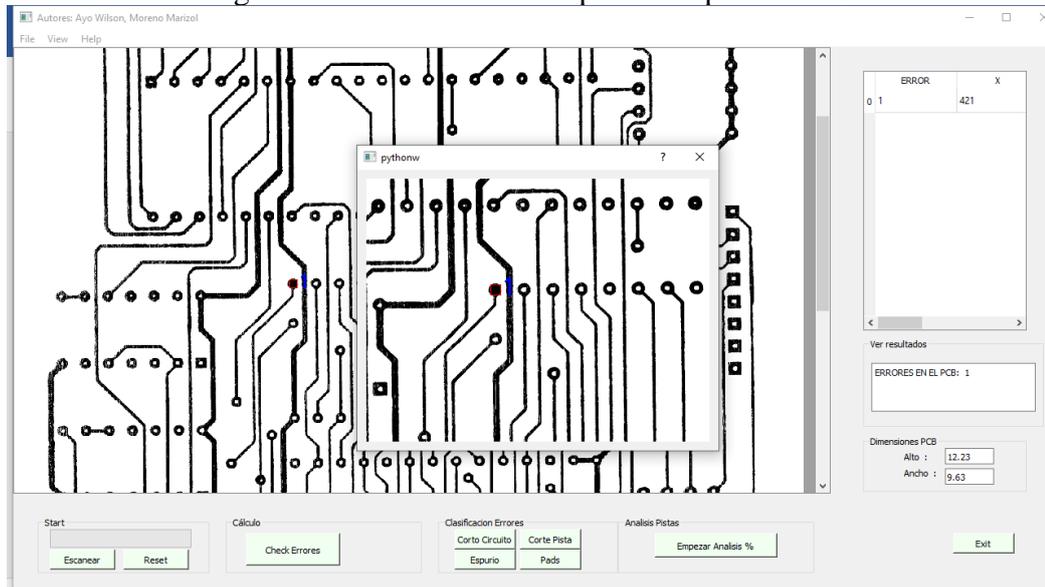
Figura 4.3 Funcionamiento algoritmo en PCB 8.



Elaborado por: Hilda Moreno

La Figura 4.4, demuestra que la aplicación, detecta 1 error (pérdida drill) en un pad.

Figura 4.4 Funcionamiento aplicación para PCB 8.



Elaborado por: Hilda Moreno

En la Tabla 4.3, se describe los errores detectados en el PCB por el sistema de visión artificial, y los tiempos que emplearon en hallar los defectos en los PCBs.

Tabla 4.3 Inspección por visión artificial (I.V.A.).

N.º PCB	Área PCB [cm]	Densidad [# pistas]	Error Intencional PCB	Error I.V.A.	Tiempo Ejecución [s]	Tiempo escaneado [s]	Tiempo Total [s]
1	14,27x13,57	alta	24	23	2,76	27,66	30,42
2	8,65x11,08	media	16	16	2,44	25,23	27,67
3	11,81x12,8	baja	5	5	2,51	22,28	24,79
4	5,48x8,19	media	8	9	2,11	16,30	18,41
5	17,64x12,11	alta	22	21	2,27	57,80	60,07
6	11,48x4,49	media	13	13	2,37	21,13	23,5
7	12,23x9,62	alta	29	27	2,29	26,41	28,7
8	12,23x9,62	alta	2	1	2,14	26,39	28,53

Elaborado por: Hilda Moreno

4.3 PRUEBA DE INSPECCIÓN POR VISIÓN HUMANA

Esta prueba consiste en realizar la inspección de las 8 placas PCB por visión humana, para esta prueba se pidió a dos personas realizar la inspección, una persona tiene 3 años realizando esta tarea, es la encargada de realizar la inspección de las placas PCB de la empresa DINNOVASOFT (I.H.V.E.), mientras que la otra persona no tiene experiencia en inspección (I.H.V.).

En la Tabla 4.4, se muestra los errores detectados por las dos personas, con y sin experiencia y los tiempos que se tardaron en hallar los defectos de los PCBs.

Tabla 4.4 Inspección por visión humana con y sin experiencia.

N.º PCB	Área PCB [cm]	Densidad [# pistas]	Error PCB	Error I.V.H.E.	Tiempo I.V.H.E. [s]	Error I.V.H.	Tiempo I.V.H. [s]
1	14,27x13,57	alta	24	23	58,22	21	168,68
2	8,65x11,08	media	16	15	45,66	14	104,75
3	11,81x12,8	baja	5	5	25,89	5	55,44
4	5,48x8,19	media	8	8	43,26	8	68,19
5	17,64x12,11	alta	22	20	74,33	20	211,22
6	11,48x4,49	media	13	13	34,85	13	163,81
7	12,23x9,62	alta	29	26	46,88	24	186,26
8	12,23x9,62	alta	2	2	12,21	1	40,23

Elaborado por: Hilda Moreno

4.4 RESULTADOS

Tomados los datos se calcula el error relativo, dividiendo el error absoluto entre el valor real como se muestra en la siguiente ecuación:

$$E_R = \frac{|\text{error absoluto} - \text{error real}|}{\text{error real}} \quad \text{Ec. 4.2.}$$

En la Tabla 4.5, se presenta el cálculo de los errores relativos de los dos métodos de inspección en las 8 placas PCBs.

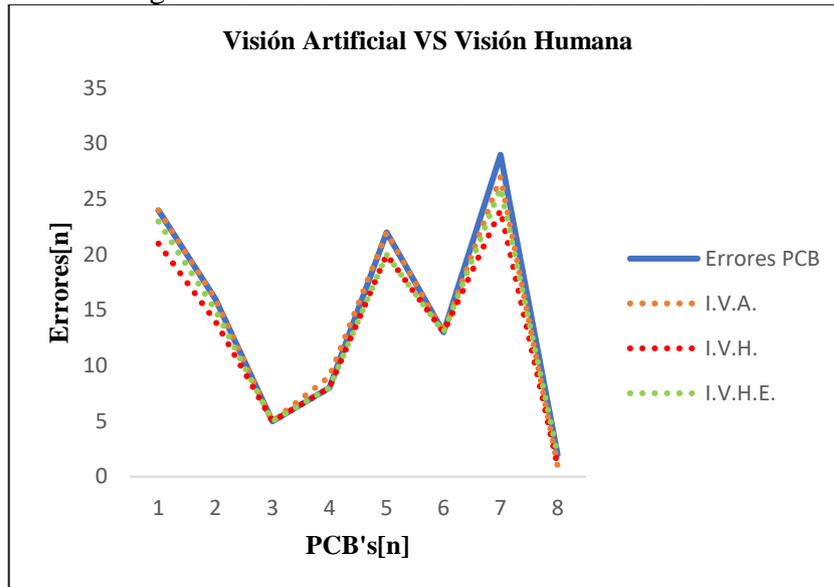
Tabla 4.5 Visión artificial vs visión humana (número de errores).

N.º PCB	Error Intencional	I.V.A.	I.V.H.	I.V.H.E.	Error I.V.A. [%]	Error I.V.H. [%]	Error I.V.H.E. [%]
1	24	24	21	23	0,00	2,52	1,68
2	16	16	14	15	0,00	1,68	0,84
3	5	5	5	5	0,00	0,00	0,00
4	8	9	8	8	0,84	0,84	0,00
5	22	22	20	20	0,00	1,68	0,00
6	13	13	13	13	0,00	0	0,00
7	29	27	24	26	1,68	2,52	1,68
8	2	1	1	2	0,84	0	0,84
Suma	119	117	106	112	3,36%	9,24%	5,04%

Elaborado por: Hilda Moreno

Los resultados; detalla los errores que presentan los dos métodos con respecto a los errores intencionales e involuntarios, el sistema de inspección por visión artificial tiene un error de 3,36%, la persona experimentada tiene 5,04% y la persona sin experiencia tiene un error de 9,24%

Figura 4.5 Visión Artificial VS Visión Humana



Elaborado por: Hilda Moreno

En la Figura 4.5 la línea azul representa los errores intencionales colocados en la placa, mientras que la línea continua de color naranja representa los errores detectados por visión artificial, estas dos líneas casi tienen la misma tendencia, las líneas continuas de color verde y rojo representan a la inspección por visión humana, siendo la línea continua de color rojo la línea que se encuentra más desviada con respecto a la línea de color azul y representa a la visión humana sin experiencia, podemos observar que con respecto a la línea de color azul existen mayor diferencia en las placas 1-2-5-7 debido a la cantidad de pistas y pads que poseen, estas se hace más difícil encontrar un error por el método de visión humana a diferencia de las placas 1-3-4-6 son más pequeñas con pistas mucho más simples donde existe menos diferencia en la cantidad de errores encontrados, sin embargo la placa 8 es de alta densidad pero solo tiene 2 errores, por lo que se observa que todas tienen casi la misma tendencia.

En la Tabla 4.6 se muestra la diferencia del tiempo entre las técnicas de inspección de PCBs de visión artificial vs visión humana.

Tabla 4.6 Visión artificial vs visión humana(tiempo).

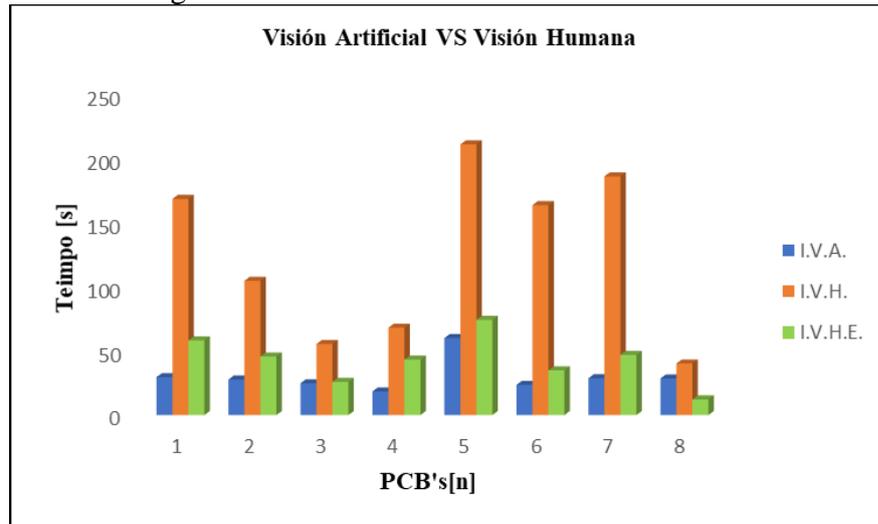
N.º PCB	Error Intencional	I.V.A. [s]	I.V.H. [s]	I.V.H.E. [s]
1	24	29,53	168,68	58,22
2	16	27,67	104,75	45,66
3	5	24,79	55,44	25,89
4	8	18,41	68,19	43,26
5	22	60,07	211,22	74,33
6	13	23,5	163,81	34,85
7	27	28,7	186,26	46,88
8	2	28,53	40,23	12,21
Suma	117	241,2[s]	998,58[s]	341,3[s]
		4,02[min]	16,64[min]	5,69[min]

Elaborado por: Hilda Moreno

En las columnas 3-4-5 se puede apreciar los tiempos que se tardaron en encontrar los errores en las 8 placas PCBs, por los dos métodos, el tiempo menor corresponde al sistema de inspección por visión artificial con 4,02min luego tenemos a la persona con experiencia con 5,69min y por último tenemos a la persona sin experiencia con 16,64min.

Esto se debe gracias a la capacidad de respuesta del ordenador, pero también se debe a una parte específica de la programación del algoritmo de detección de errores que es el escalamiento de la imagen en este caso se redujo la imagen de 0.4/1, significa cuatro veces el tamaño original, obteniendo los resultados expuestos en las tablas, lo que se mantiene constante son los tiempos en la captura de las imágenes que corresponden a la velocidad del escáner: a una resolución de 300 ppp, como referencia, monocromático: 2.4 milisegundos por línea y color: 9.5 milisegundos por línea.

Figura 4.6 Visión Artificial VS Visión Humana



Elaborado por: Hilda Moreno

En la Figura 4.6 se observa la diferencia de la visión artificial y la visión humana, la gráfica de color azul corresponde a la visión artificial mientras que el gráfico de color verde representa la visión humana con experiencia, la gráfica de color naranja representa la visión humana sin experiencia se puede apreciar la diferencia que existe entre estas.

Para fines de verificación y funcionamiento, se realiza tres repeticiones en la prueba de inspección por el método de: visión artificial, ejecutado por la aplicación, en las 8 placas PCB, la Tabla 4.7, señala los resultados de la prueba, cabe recalcar que los valores que más se repitieron fueron tomados para el análisis de comparación.

Tabla 4.7 Repeticiones del sistema de inspección mediante visión artificial.

Número PBC	Repetición 1 [# errores]	Repetición 2 [# errores]	Repetición 3 [# errores]
PCB 1	23	23	23
PCB 2	16	16	16
PCB 3	5	5	5
PCB 4	9	9	8
PCB 5	22	22	22
PCB 6	13	13	13
PCB 7	27	25	27
PCB 8	1	2	1

Elaborado por: Hilda Moreno

CAPÍTULO 5

5.1 CONCLUSIONES

Se desarrolla un sistema de inspección automático para la detección y localización de fallos recurrentes en PCBs como: cortocircuito, circuito abierto, exceso de cobre y daños en pads, por visión artificial a través del lenguaje de programación Python y las librerías OpenCV, mediante técnicas de procesamiento digital de imágenes por computador, se utiliza algoritmos para el análisis y reconocimiento de defectos en base a la comparación de dos imágenes, una corresponde al circuito impreso real PCB y la otra a una imagen de la PCB sin errores.

Se implementa el hardware para el sistema de inspección automático, para la detección de fallos en placas de PCBs de hasta 21x30cm, que consiste en una estructura metálica que contiene el hardware para la adquisición de la imagen, dentro de esta se encuentra el escáner para la captación de la imagen, una fuente auxiliar de iluminación y un sistema de desplazamiento para transportar el PCB, de esta forma simplemente se coloca el PCB en el porta placas, se acciona un pulsador y este lo transporta hacia la ventana de escáner para ser escaneado con mayor facilidad. En la repisa que se encuentra al costado izquierdo de armario se ubica, el ordenador el cual mediante la programación desarrollada en Python y OpenCV, realiza el procesamiento digital para el análisis y reconocimiento de los defectos en el PCB, finalmente a través de la interacción de una interfaz gráfica desarrollada en Python y PYQT5 se puede observar los defectos sobre la imagen de prueba.

Se diseña un sistema de captura y adquisición de imágenes, que consiste en una cámara metálica de (50x60) cm de color oscuro cerrada, iluminada con una fuente de luz auxiliar, y que contiene al escáner, estos ajustes permiten intensificar la cantidad de luz sobre la superficie del PCB, mejorar la distribución de la luz desde el centro a la extremos de la placa, y controlar la variación lumínica de forma contraria que con tapa cerrada del escáner, que tiene un fondo blanco, de esta manera y con la ayuda del filtrado de la imagen, operaciones morfológicas, detección de bordes se obtiene a la entrada imágenes de mejor calidad, para el procesamiento digital, análisis y reconocimiento de defectos en los PCBs.

En este proyecto se desarrolla una interfaz gráfica funcional, sencilla y rápida de utilizar, que facilita al usuario, el acceso al sistema de inspección automático de PCBs, la interfaz posee botones específicos para cada acción señalados correctamente y agrupados según su función, el visor de imagen está desarrollada con barras desplazables que se ajustan al tamaño del PCB y puede ser modificado según se requiera, en general los usuarios no tienen inconvenientes para manipular y percibir los resultados.

Se realiza un análisis comparativo de detección de errores, entre inspección por visión artificial e inspección por visión humana, se obtiene como resultado que la visión por computador es más rápida que el ojo humano incluso un humano con experiencia, el tiempo de la captura en promedio de las 8 placas realizado por la aplicación es de 27 segundos, y el tiempo de ejecución de maquina es de 2 segundos. Mientras que en la prueba de ubicación de los defectos en el PCB realizado por humanos, el tiempo es mayor y supera el porcentaje de error en especial en circuitos más pequeños y densos.

5.2 RECOMENDACIONES

Se recomienda utilizar para proyectos de visión artificial un ordenador que cuente con un procesador y memoria capaces de soportar la ejecución de algoritmos y procesamientos digitales de imágenes que son básicamente operaciones simultáneas y múltiples que se realizan en tiempos relativamente cortos, se pudo verificar en los niveles de consumo del administrador de tareas del PC que son proporcionales a la dimensión y densidad de las pistas de los circuitos impresos.

Si se requiere de un sistema de inspección en tiempo real, se puede hacer uso de otro dispositivo electrónico para la captura o mejorar la versión del escáner, o cualquier dispositivo que sea capaz de obtener una imagen de calidad sin ralentizar el tiempo en la inspección.

Se recomienda la optimización del programa, en la recursividad y reutilización de los algoritmos, esto con el fin que se ejecuten operaciones en menor tiempo, y reduzca el número de líneas de código, para obtener mayor versatilidad y flexibilidad en el editor de texto del código.

REFERENCIAS

- Alberto, G. (2011). Controladores PID y Controladores Difusos. Revista de la Ingeniería Industrial, 13.
- Ampuño, G. A. (2014). Implementación y análisis de rendimiento de un control industrial de nivel para tanques con fluidos, basado en lógica difusa. MASKANA, Ingeniería, 10.
- Ashish Singh, M. G. (noviembre de 2014). Automatic Detection of PCB Defects. Obtenido de Automatic Detection of PCB Defects: file:///H:/DOCUMENTOS/TESIS/Automatic_Detection_of_PCB_Defects.pdf
- Bravo, L. (14 de Marzo de 2012). Control I. Obtenido de <http://leobravocontrol1.blogspot.com/2012/03/que-es-un-sistema-de-control.html>
- Chauhan, A. P. (8 de Julio de 2011). SMTNET. Obtenido de Detection of Bare PCB Defects by Image: https://www.smtnet.com/library/files/upload/WCE2011_pp1597-1601.pdf
- Cognex. (16 de Septiembre de 2017). Assemblymag. Obtenido de INTRODUCTION TO MACHINE VISION: https://www.assemblymag.com/ext/resources/White_Papers/Sep16/Introduction-to-Machine-Vision.pdf
- Danilo, A., & Ing William, M. (01 de Mayo de 2015). Diseño y desarrollo de una plataforma de entrenamiento para el control de procesos continuos. Obtenido de Diseño y desarrollo de una plataforma de entrenamiento para el control de procesos continuos: <https://dspace.ups.edu.ec/bitstream/123456789/10156/1/UPS%20-%20ST001779.pdf>
- Doniwar, A. (3 de Marzo de 2017). IOPScience. Obtenido de PCB Faults Detection Using Image Processing: <https://iopscience.iop.org/article/10.1088/1757-899X/225/1/012244/pdf>
- EcuRed. (23 de Febrero de 2012). EcuRed. Obtenido de EcuRed: https://www.ecured.cu/Circuito_integrado_LM358

- Electromecánica, M. (21 de Enero de 2019). Electromecánica MM. Obtenido de https://www.electromecanicamm.com.ar/buscador.php?txt_buscar=bomba+leo
- González, M. M. (2012). Visión por Computador para UAS . Obtenido de Visión por Computador para UAS : http://oa.upm.es/14316/2/PFC_MONICO_MANUEL_PONCE_GONZALEZ_B.pdf
- Jiménez, R. F. (2011). Diseño e implementación de un sistema embebido para la adquisición y transmisión de señales biomédicas a través de la red celular. Colombia.
- Kumar, Y. (1 de Enero de 2017). UERT. Obtenido de Automatic Misalignment Defects Detection & Correction in PCB using SURF: <https://www.ijert.org/automatic-misalignment-defects-detection-correction-in-pcb-using-surf-speed-up-robust-features-technique>
- Kunte, O. (enero de 2018). Semantic Scholar. Obtenido de Automatic Optical Inspection of Printed: https://pdfs.semanticscholar.org/2614/216b1d564aa984993a8798a2efe4a2c129c4.pdf?_ga=2.3669936.123074305.1591785193-174917060.1591785193
- Londe, P. P. (20 de Junio de 2014). Inpresso. Obtenido de Automatic PCB Defects Detection and Classification using Matlab: <http://inpressco.com/wp-content/uploads/2014/06/Paper1802119-2123.pdf>
- Luna, G. M. (2002). Introducción a la Logica Difusa Centro de investigación y estudios avanzados México. México.
- Mechatronics, N. (20 de Enero de 2019). Naylam Mechatronics. Obtenido de Naylam Mechatronics: <https://naylampmechatronics.com/atmel/111-atmega328.html>
- Mendoza, A., Nehider, L., Neimir, B., Javier, R., & Eugenio, Y. (14 de Abril de 2018). Revista UIS Ingenierías. Obtenido de Revista UIS Ingenierías: <https://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/6290/8185>
- Morales Rodrigo, S. R. (Septiembre de 2018). Ribuni. Obtenido de Desarrollo de una metodología para la elaboración de tarjetas PCB de: <http://ribuni.uni.edu.ni/2265/1/92097.pdf>
- Ms. VinitaP, M. s. (enero de 2016). PCB Fault Detection by Image Subtraction Method. Obtenido de PCB Fault Detection by Image Subtraction Method:

file:///H:/DOCUMENTOS/TESIS/ff6482833d10b441978f086169a2085907af.pdf

Neelum Dave, V. T. (mayo de 2016). PCB Defect Detection Using Image Processing And Embedded System . Obtenido de PCB Defect Detection Using Image Processing And Embedded System : file:///H:/DOCUMENTOS/TESIS/6fe4785a05ab31f9ab25c523c794b8064ce7.pdf

Ogata, K. (2010). Introduccion a los sistemas de control. En Ingenieria de control moderna (págs. 1-40). Madrid: Pearson Education.

Pérez, I. R. (2016). Sistemas Microprocesados. Quito.

Punzenberger, I. (02 de Enero de 2018). COPADATA. Obtenido de <https://www.copadata.com/es/soluciones-hmi-scada/interfaz-hombre-maquina-hmi/>

Querejeta Simbeni, P. (Julio de 2017). UNS. Obtenido de Procesamiento digital de imágenes: <http://lcr.uns.edu.ar/fvc/notasdeaplicacion/fvc-querejetasimbenipedro.pdf>

Salvetti, I. D. (23 de Mayo de 2012). Electrónica Industrial. Obtenido de <http://eet602ei.blogspot.com/2012/05/sistemas-de-control-lazo-abiertocerrado.html>

Saúl, G. (gosto de 2014). UVADOC. Obtenido de Universidad de Valladolid: <https://uvadoc.uva.es/bitstream/handle/10324/13153/TFG-I-174.pdf;jsessionid=34E3317A8A589A058647F50C770BC27B?sequence=1>

Sergio Sada Pezonaga, J. A. (e junio de 2015). Tratamiento de imágenes con ruido impulsivo mediante reglas difusas y algoritmos genéticos. Obtenido de Tratamiento de imágenes con ruido impulsivo mediante reglas difusas y algoritmos genéticos: <http://academica-e.unavarra.es/bitstream/handle/2454/19033/MemoriaTFG--SergioSada.pdf?sequence=1>

Siemens. (02 de Enero de 2018). Siemens global web site. Obtenido de https://w5.siemens.com/spain/web/es/industry/automatizacion/hmi/pages/simatic_hmi_1.aspx

Silvia, F. N. (Marzo de 2012). Biblio3. Obtenido de Universidad Rafael Landívar: <http://biblio3.url.edu.gt/Tesis/2012/05/43/Fuentes-Silvia.pdf>

- Ulloa javier, D. D. (21 de marzo de 2017). Picmania. Obtenido de Tecnicas de montaje de Circuitos Impresos: https://picmania.garcia-cuervo.net/recursos/redpictutorials/fabricacion_pcb/pcbs_preliminar.pdf
- Viera-Maza, G. (marzo de 2017). PROCESAMIENTO DE IMÁGENES USANDO OPENCV APLICADO EN RASPBERRY PI PARA LA CLASIFICACIÓN DEL CACAO. Obtenido de PROCESAMIENTO DE IMÁGENES USANDO OPENCV APLICADO EN RASPBERRY PI PARA LA CLASIFICACIÓN DEL CACAO: https://pirhua.udep.edu.pe/bitstream/handle/11042/2916/IME_218.pdf
- Weihrich, K. y. (2012). Academia. Obtenido de McGRAW-HILL: https://www.academia.edu/15364931/Administraci%C3%B3n_14_ed_-_Harold_Koontz_Weihrich_y_Cannice