SISTEMA PARA IDENTIFICACIÓN DE FALLAS PARA CONECTOR PRINCIPAL DE PRODUCTO TERMINADO CON AYUDA DE MACHINE LEARNING

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA

MONTES RAMÍREZ MIGUEL ANTONIO

DIRECTORA

M.C. GUILLERMINA MUÑOZ ZAMORA

CO DIRECTOR

M.C. OMAR VELARDE ANAYA

H. NOGALES, SONORA, MÉXICO. **JUNIO DE 2022**

Nogales, Sonora, 23/junio/2022
Oficio No. DEPI/179/2022.

REYNALDO GUTIÉRREZ GUTIÉRREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
PRESENTE.

Por este medio le comunicamos a Usted que el trabajo de Tesis denominado: "**SISTEMA PARA IDENTIFICACIÓN DE FALLAS PARA CONECTOR PRINCIPAL DE PRODUCTO TERMINADO CON AYUDA DE MACHINE LEARNING**.", que presentó el alumno **MONTES RAMÍREZ MIGUEL ANTONIO**, con número de control 18341003, candidato a obtener el grado de **Maestro en Sistemas Computacionales**, ha sido revisado por los miembros del Comité Tutorial y cubiertas las observaciones realizadas, se Autoriza su Impresión y se Acepta para su Evaluación en la presentación del Examen de Grado.

Agradeciendo de antemano el apoyo brindado al presente, le reitero mi consideración distinguida.

ATENTAMENTE
Excelencia en Educación Tecnológica·
"La Ciencia y la Tecnología para la Liberación del Hombre"·

SIGIFREDO GARCIA ALVA
REVISOR

MANUEL OMAR MERANZA CASTILLON
REVISOR

OMAR VELARDE ANAYA
REVISOR

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLOGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE NOGALES
DIVISIÓN DE ESTUDIOS DE,
POSGRADO E INVESTIGACIÓN

H. Nogales Sonora, 23/junio/2022.
Oficio No. DEPI/180/2022.

MIGUEL ANTONIO MONTES RAMÍREZ
CANDIDATO A OBTENER EL GRADO DE MAESTRO EN SISTEMAS COMPUTACIONALES
PRESENTE.

De acuerdo con el Reglamento de Titulación del Sistema Nacional de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que el Comité Tutorial realizó, con respecto a su Tesis titulada: "**SISTEMA PARA IDENTIFICACIÓN DE FALLAS PARA CONECTOR PRINCIPAL DE PRODUCTO TERMINADO CON AYUDA DE MACHINE LEARNING**", la División de Estudios de Posgrado e Investigación Autoriza su Impresión.

Agradeciendo de antemano el apoyo brindado al presente, le reitero mi consideración distinguida.

ATENTAMENTE
Excelencia en Educación Tecnológica®
"La Ciencia y la Tecnología para la Liberación del Hombre"®

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE NOGALES
DIVISIÓN DE ESTUDIOS DE,
POSGRADO E INVESTIGACIÓN

REYNALDO GUTIÉRREZ GUTIÉRREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

ccp. Archivo

RGG/cmmj

SISTEMA PARA IDENTIFICACIÓN DE FALLAS PARA CONECTOR PRINCIPAL DE PRODUCTO TERMINADO CON AYUDA DE MACHINE LEARNING

RESUMEN

Este trabajo plantea el desarrollo de un sistema de visión artificial utilizando machine learning por medio de un modelo de clasificación de imágenes como solución al problema de fallas ocasionadas por pines opacos en el *main connector* del radio de FCA (Fiat Chrysler Automobiles) MP Refresh en la estación Pin&Fakra (P&F), así como el conteo de 40 pines en la parte B del *main connector* mediante un script o secuencia de inspección que consiste en el procesamiento de imágenes. El sistema es usado para la inspección visual y clasificación del conector principal. Esto permitirá clasificar los conectores con suficiente brillo en los pines, o si existe opacidad en ellos, lo que conlleva tener un mejor manejo de material al contar con una contención para las unidades que presenten deficiencia en el brillo de los pines.

El modelo de clasificación de imágenes se basa en el uso de Machine Learning (ML.Net) de Visual Studio en la interfaz de Model Builder, logrando una exactitud del 96%. Este modelo crea los archivos de librerías necesarios para implementarlos en cualquier lenguaje de programación, sin embargo, en el desarrollo se presentan implementados en el software LabWindows/CVI. La adquisición de las imágenes se hace mediante la cámara Mako G-319C de Allied Vision, la cual es robusta y capaz de trabajar en ambientes industriales de 24hr/7dias. El procesamiento de las imágenes del conector principal se desarrolla en el software de Vision Assistant de National Instruments. Con la ayuda de Vision Assistant se realiza un análisis de partículas, que nos permite el conteo de los 40 pines de la parte B del *main connector*.

De acuerdo con la hipótesis presentada de reducción de fallas de un 40.6% a un 10%, con las pruebas realizadas se logra una reducción del 18% de las fallas.

ABSTRACT

This work set out the development of an artificial vision system using machine learning by means of an images classification model as a solution to fails problem caused for opaque pins in the main connector of FCA (Fiat Chrysler Automobiles) MP Refresh's radio at the workstation Pin&Fakra (P&F), as well as count the 40 pins in main connector's side B, by means of a script or a sequence inspection that it consists in the processing of images. The system is used for visual inspection and main connector classification. This will let to classify the connectors with brightness enough on pins, or if there is opacity on them, this implicates to have a better material control because it has a containment for pieces with less brightness on pins.

The Image classification model is based on using machine learning of Visual Studio in the model builder interface achieving an accuracy of 96%. This model creates the necessary library files to implement them in any kind of programming language, however the development is implemented in LabWindows/CVI software. The images are gotten by means of Mako G-319 camera by Allied Vision, which is heavy duty and able to work in an industrial environment of 24hrs/7days. The processing of images of main connector is developed in Vision Assistant software of National Instruments. With the help of Vision Assistant is made particles analysis, that it lets to count 40 pins of main connector's side B.

According to the hypothesis established in reducing fails from 40.6% to 10%, with the tests made is achieved to decrease 18% of fails.

AGRADECIMIENTOS

Quiero agradecer primeramente a Dios, por el privilegio que me ha permitido de llegar hasta esta etapa de mi vida, por darme salud y estabilidad, y por ser partícipe de su infinita misericordia día a día. Tal como dice el profeta Samuel en la palabra de Dios:

"Hasta aquí el Señor nos ha ayudado."

1 Samuel 7.12 NTV

Agradezco infinitamente a mis padres, el Sr. Miguel A. Montes Mendoza y Sra. María Del Consuelo Ramírez Gutiérrez, ya que ellos han sido una fuente de inspiración en el trabajo duro, en el esfuerzo y en la calidad de vida que he tratado de vivir hasta este momento. Agradezco su comprensión y su apoyo a lo largo de esta etapa. Gracias por sembrar en mí la palabra de Dios y los valores para quererme superar cada día.

A mi hermano Ing. Jorge I. Montes Ramírez, por su apoyo durante todo este tiempo. Gracias por esos recordatorios de que siempre debemos seguir adelante a pesar de las dificultades que se nos presenten en el camino.

A mis amigos y compañeros M.C. Gustavo Tautímez e Ing. Jaudier Navarro, gracias por siempre estar dispuestos a ayudarme en el desarrollo de este proyecto, gracias por la base de conocimientos que me han dado al trabajar con ustedes y por demostrarme que todo se puede llevar a cabo.

Finalmente, gracias a mis maestros, M.C. Omar Velarde Anaya, por los conocimientos brindados y su dirección en el desarrollo del proyecto. A mi directora de tesis M.C. Guillermina Muñoz Zamora por su paciencia y comprensión, por confiar en mí incondicionalmente y siempre estar al pendiente de cada una de las cosas que hacía.

DEDICATORIA

A mis padres. Sin su ayuda y motivación, no hubiese podido llegar a cumplir este sueño. Los quiero demasiado, esto es el resultado del esfuerzo que han hecho en formarme como persona y darme las herramientas para poder lograrlo. Ustedes son merecedores de toda mi honra. ¡Los Amo!

ÍNDICE DE CONTENIDO

Página

ÍNDICE DE TABLAS

Página

ÍNDICE DE FIGURAS

Página

INTRODUCCIÓN

CAPÍTULO I. ANTECEDENTES

1.1.- Continental en México

Continental Automotive produce y desarrolla componentes automotrices, llantas y productos de caucho que se exportan a distintos países. En México tiene 21 locaciones, entre fábricas, oficinas, y centros de investigación y desarrollo, en 12 estados de la república, siendo una pieza clave de la industria de la movilidad en México (Continental M. , 2022). Continental Automotive Nogales se dedica al desarrollo de componentes automotrices, enfocados a la conectividad y a la seguridad de los automóviles. En los últimos meses la corporación ha tenido cambios en sus unidades de negocio, incluyendo estos cambios en la industria 4.0, el cual se caracteriza por la integración de nuevas tecnologías y modelos de negocio, que lleva a mejorar los procesos industriales. El control de calidad en la industria 4.0 se denomina Calidad 4.0 y consiste en la interconexión entre las personas, máquinas y datos mediante diferentes tecnologías con el objetivo automatizar, analizar y obtener resultados positivos que permitan a la empresa abaratar costes y obtener mayores beneficios. (Peña Lorenzo, 2020).

1.1.1.- Visión Artificial por Unidad de Negocio

La visión artificial es una de las tecnologías de mayor auge en el campo de la calidad 4.0, esta tecnología consiste en adquirir, procesar y analizar imágenes mediante distintos algoritmos. En la tabla 1.1 podemos ver las distintas unidades de negocio con la cantidad de estaciones de visión del departamento de Ingeniería de Pruebas con esta tecnología.

Tabla 1.1 Unidades de Negocio y Cantidad De Estaciones de Visión por Back-End

Unidad de Negocio	Back - End de Producción	Estaciones con VA
User Experience (UX)	PSA AIO (Peugeot y Citroën)	2
	Renault XBB	2
	MP Refresh 1 (FCA)	3
	MP Refresh 2 (FCA)	3
	Flat Panel (FCA)	4
	VPX2 (FCA)	3
	CTP Base (FCA)	1
	RES (FCA)	2
Smart Mobility (SMY)	LDM2 Celda 1 (Ford Motor)	2
	LDM2 Celda 2 (Ford Motor)	0
	LDM2 Celda 3 (Ford Motor)	1
Architecture and Networking (AN)	Global BCM	1
	Subaru	2
	CBC Chrysler	1
	Toyota	1
	Hyundai	2
	CBC Daimler	0
	CSMM	0
	CSWM	0
	DCU	0
	DCU MFA2	1
	DC - DC	0
	Ford Gen3	4
	EPM	0
	ITBM	0
	KART	0
	MSM	0
	GM MSM	0
	PQ25	0
	RF Hub	0
	PLGM	0
FEM	FEM HC	1
	FEM LC	0
	Total, de Estaciones	**36**

Cada una de estas estaciones cuenta con diversas tecnologías de visión artificial, entre las cuales se pueden encontrar de National Instruments (Vision Assistant y Vision Builder), Keyence y Cognex, las cuales realizan inspección al 100% del producto terminado 24/7 de manera automática, el cual puede involucrar las siguientes tareas:

- Presencia o ausencia de características especiales.
- Dimensiones de esas características.
- Imperfecciones y daños.

1.1.2.- User Experience

User Experience (UX) es una unidad de negocios del sector automotriz de Continental, encargada del desarrollo de soluciones visuales, así como manufactura de radios automotrices para que la experiencia del usuario se ajuste a sus necesidades.

1.1.2.1.- Enfoque de UX

Las necesidades de los usuarios finales están cambiando, el diseño interior y la experiencia de usuario se convierten en factores decisivos en la decisión de comprar un vehículo. La unidad de negocios UX desarrolla productos y contenidos innovadores que no solo hacen que la conducción sea menos estresante, sino que permiten nuevas experiencias. El área de negocio combina tecnologías de visualización y entrada, así como soluciones de audio y cámara para el interior del vehículo. (Continental, 2022)

1.1.2.2.- UX en Nogales

Actualmente se cuenta con dos Back – End de MP Refresh para FCA (Fiat Chrysler Automobiles), en los cuales cada uno tiene una estación de inspección de producto final llamadas Pin&Fakra, las cuales se encargan de la detección de presencia y ausencia de pines en los diferentes conectores del radio que se manufactura en la celda de producción, como se puede ver en la Figura 1.1.

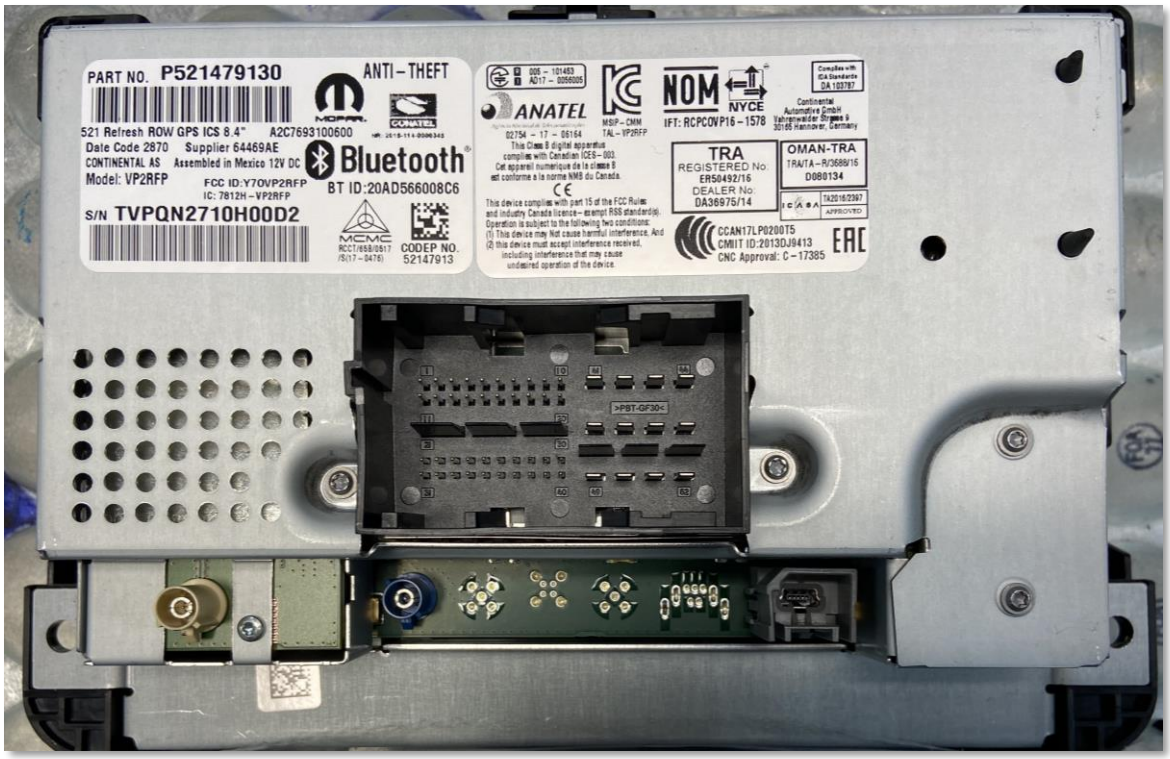

Figura 1.1 Parte posterior de radio de FCA

1.2.- Definición Actual del Problema

Hay diversas variantes de estos radios, cada uno dependiendo de la tecnología con la que el cliente lo requiera, sin embargo, todos cuentan con un elemento en común, que es el conector principal o *“main connector”*, este conector contiene los

pines de encendido y control del radio. Las pruebas realizadas actualmente en Pin&Fakra para el radio son las siguientes:

1. *main connector*
    a. *main connector* pattern
    b. Pin Count A
    c. Alineación de los pines del conector A
    d. Pin Count B
    e. Alineación de los pines del conector B

2. Tuner 1
    a. Fakra Pattern
    b. Pin Present
    c. Alignment
    d. Color
3. Tuner 2
    a. Fakra Pattern
    b. Pin Present
    c. Alignment
    d. Color
4. GPS
    a. Fakra Pattern
    b. Pin Present
    c. Alignment
    d. Color
5. USB
    a. USB Pattern
    b. Pin Present
    c. Pin1 Aligment
    d. Pin2 Aligment
    e. Pin3 Aligment

f. Pin4 Aligment
g. Pin5 Aligment

En datos obtenido en el año 2021 de algunas de las estaciones de visión, se puede notar que la celda de producción con mayor impacto en fallas es la de MP Refresh (Figura 1.2). La celda de producción MP Refresh también puede ser llamada como VP2 Refresh, ya que así si se encuentra en la Figura 1.2.

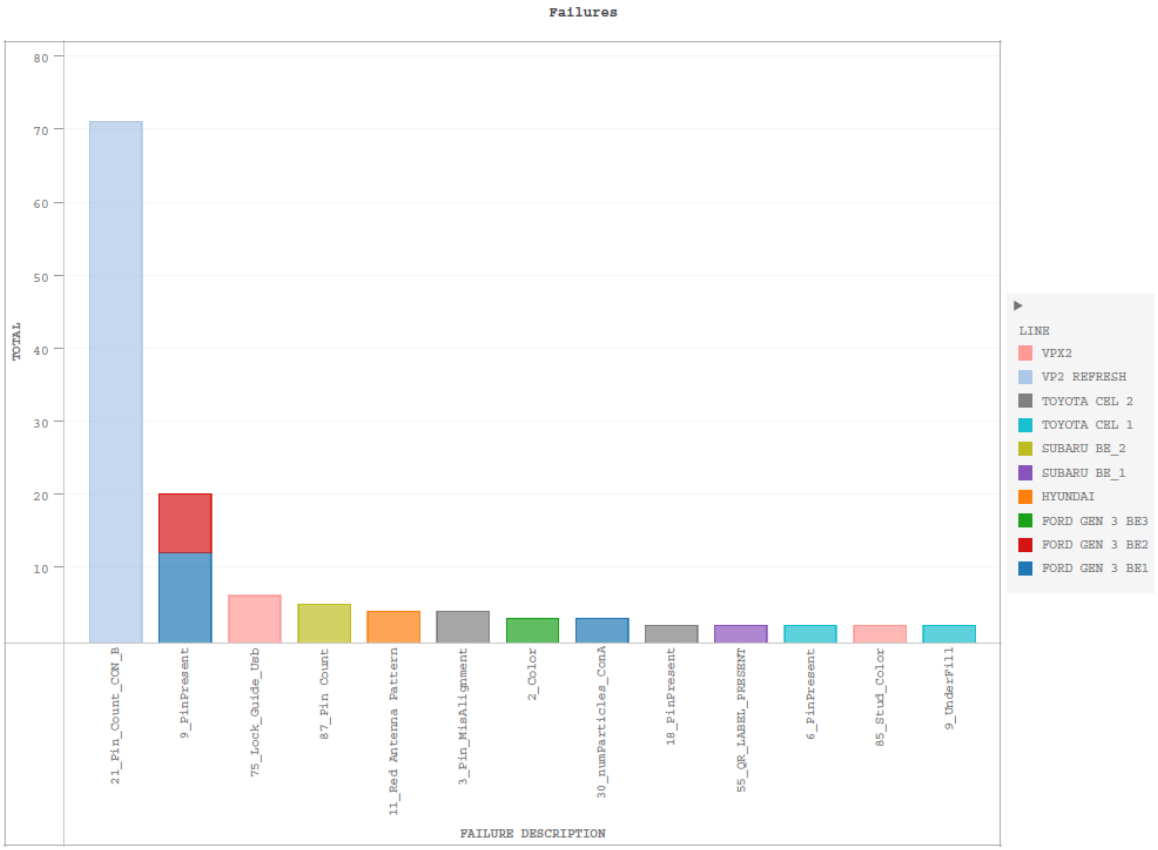

Figura 1.2 Fallas de equipos de visión de diferentes productos

El término "A3" se refiere a una hoja de papel de tamaño internacional, que mide aproximadamente 11 por 17 pulgadas. Sin embargo, dentro de Toyota y para otras empresas lean, que son compañías que basan sus procesos en herramientas de

Manufactura Esbelta (Lean Manufacturing) el término significa mucho más. Toyota tuvo la idea de que cada problema puede y debería capturarse en una sola hoja de papel. Esto permite que el equipo multidisciplinario pueda ver el tema en cuestión a través del mismo lente. El formato y redacción de una hoja de A3 es flexible, las organizaciones ajustan el diseño a sus requisitos (Shook, 2008, pp. 7-11). Esta herramienta se utiliza para implementar la gestión PDCA (plan, do, check, act) en todos los departamentos y niveles de la organización. (Durwad K. Sobek II and Art Smalley, 2008, p. 29).

Basados en la herramienta A3 para la resolución de problemas (Figura 1.4) de la línea en cuestión (línea VP2 Refresh), se puede ver cuáles son las mayores fallas que impactan el indicador *First Pass Yield* (Rendimiento de Primera Pasada), siendo una de ellas la falla en la prueba “21_Pin_Count_CON_B” de la estación Pin&Fakra. Esta prueba considera solamente el conector B del *main connector* (Figura 1.3).

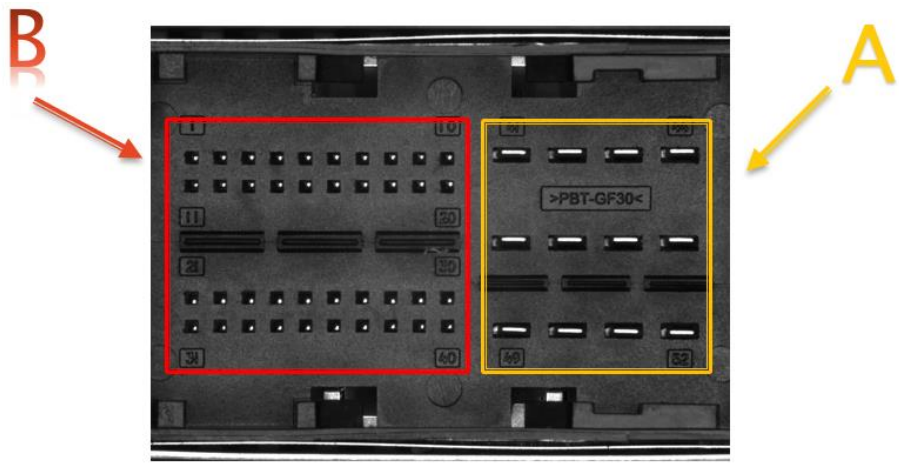

Figura 1.3 Parte A y B del Conector Principal

Al considerar la información de la estación en particular, encontramos que esta falla impacta en un 40.6% el FPY (*First Pass Yield*) de la estación (Tabla 1.2).

Tabla 1.2 Principales fallas de la estación Pin&Fakra

Row Labels	Failed Units	%Impact	%Acum.
21_Pin_Count_CON_B	9287	40.6%	40.6%
81_TUNER_1_Alignment	3191	13.9%	54.5%
86_GPS_Alignment	2609	11.4%	65.9%
86_SDAR_Alignment	2450	10.7%	76.7%
91_DAB_Alignment	1299	5.7%	82.3%
3_Pin_Count_CON_A	1038	4.5%	86.9%
80_TUNER_1_Pin_Present	547	2.4%	89.3%
85_SDAR_Pin_Present	289	1.3%	90.5%
90_DAB_Pin_Present	274	1.2%	91.7%
92_DAB_Color	249	1.1%	92.8%
103_USB_2_Pin_Present	146	0.6%	93.4%
95_USB_Pin_Present	132	0.6%	94.0%
90_USB_Pin_Present	120	0.5%	94.5%
87_SDAR_Color	90	0.4%	94.9%
68_Pin_21_31_Axis_Y	84	0.4%	95.3%
58_Pin_1_11_Axis_Y	80	0.3%	95.7%
Others	993	4.3%	100.0%
Grand Total	22878	100.0%	

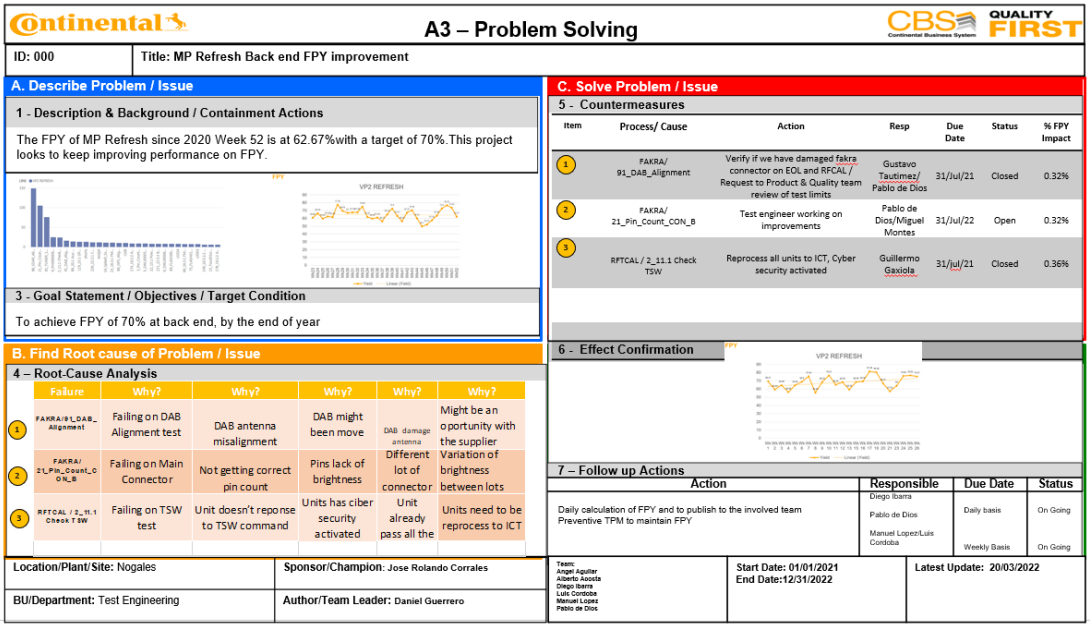

Figura 1.4 A3 – Problem Solving

En la sección B del A3 (Figura 1.5) utilizamos los cinco ¿Por qué? para análisis de problemas:

Falla 21_Pin_Count_CON_B:

1. Fallo en el conector principal.
2. No obteniendo el recuento correcto de pines.
3. Pines con falta de brillo.
4. Diferente lote de conector.
5. Variación de brillo entre lotes.

B. Find Root cause of Problem / Issue

4 – Root-Cause Analysis

	Failure	Why?	Why?	Why?	Why?	Why?
1	FAKRA/91_DAB_Alignment	Failing on DAB Alignment test	DAB antenna misalignment	DAB might been move	DAB damage antenna	Might be an oportunity with the supplier
2	FAKRA/ 21_Pin_Count_CON_B	Failing on Main Connector	Not getting correct pin count	Pins lack of brightness	Different lot of connector	Variation of brightness between lots
3	RFTCAL / 2_11.1 Check TSW	Failing on TSW test	Unit doesn't reponse to TSW command	Units has ciber security activated	Unit already pass all the	Units need to be reprocess to ICT

Figura 1.5 Sección B de A3

Después de las herramientas de resolución de problemas utilizadas, se encuentra que la cantidad de fallas en la estación para la prueba 21_Pin_Count_CON_B es debido a la opacidad presentada por los pines del conector B del conector principal del radio.

1.3.- Estado del arte

A continuación, se muestran algunos proyectos en la industria enfocados en inspección por medio de visión artificial, que fueron desarrollados utilizando Visión Artificial, *Deep Learning* y *Machine Learning*.

En Hermosillo, Sonora México, se desarrolló un sistema basado en visión artificial para la industria automotriz, el cual se utiliza para la evaluación de la calidad en el ensamble de conectores de bolsas de aire automotrices. El sistema clasificador de los conectores se realizó utilizando un algoritmo de redes neuronales convolucionales, el modelo de redes neuronales fue entrenado bajo metodologías existentes recomendadas para estos modelos. Los resultados obtenidos muestran aceptabilidad para la industria automotriz, reduciendo el tiempo de inspección, y logrando disminuir las quejas de cliente por piezas con defecto. (Rodarte Leyva, 2021)

Por otro lado, en Ecuador se desarrolló un sistema prototipo de luces frontales con segmentación para automotores empleando técnicas de visión artificial difusas. Para el sistema de detección de los vehículos se utilizó el algoritmo "*Haar Cascade*", propuesto por Paul Viola y Michael Jones en el artículo "*Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade*" (Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade, 2001). La función de cascada se entrena a partir de múltiples imágenes positivas y negativas, una vez culminado el entrenamiento se utiliza para detectar los objetos en otras imágenes. (Jhinson Edgar Coyago Tomalo, Jonathan Javier Coyago Tomalo, 2019)

En el *Beijing Institute of Technology* se desarrolló un proyecto llamado "Detección cuantitativa de aflojamiento de sujetadores roscados mediante aprendizaje profundo basado en visión y teoría de imágenes geométricas". Este artículo propone un método para calcular cuantitativamente la longitud del perno expuesto para detectar el aflojamiento mediante el aprendizaje profundo basado en la visión y la teoría de imágenes geométricas. En dicho proyecto se utiliza una Faster-RCNN (*Faster Regional Convolutional Neural Network*) la cual se encarga del módulo de ubicación

de la región de interés en los pernos o tornillos que se busca inspeccionar debido a problemas ocasionados por aflojamiento. (Hao Gong, Xinjian Deng, Jianhua Liu, Jiayu Huang, 2021)

En el Instituto Tecnológico de Nogales en la División de Estudios de Posgrado e Investigación se publicó un artículo del proyecto “Diseño de interfaces de inspección de calidad utilizando redes neuronales y visión artificial para la industria de manufactura”, en el cual se aborda el tema de inspección de conectores de cables por medio de redes neuronales artificiales, así como la utilización de la técnica de erosión de imagen contra una imagen maestra. Los resultados de la validación de este sistema arrojan un 83% de exactitud para identificación del cable. (Luis Arturo Medina Muñoz, Samuel González-López, Jesús Antonio Mayorquín Robles, Gabriel Antonio López Valencia, 2019)

La empresa Continental Automotive en Nogales cuenta con un sistema de inspección a base de visión artificial en el área de Back-End de producción de Subaru en el que se utiliza una cámara con tecnología de *Deep Learning* embebida, la cual se emplea para la detección en presencia/ausencia de *foam pad* en los diversos conectores del módulo electrónico. La cámara pertenece al entorno de Cognex.

1.4.- Planteamiento del problema

La incorrecta identificación y conteo de pines en la parte B de *main connector* genera fallas falsas en la estación de Pin&Fakra (P&F) en un 40.60% de las fallas total del equipo, debido a la opacidad de los conectores.

1.5.- Preguntas del problema

1. ¿Un algoritmo basado en *Machine Learning* (ML) es capaz de identificar diferentes tipos de conectores?

2. ¿Qué utilidad tiene desarrollar un sistema de visión artificial basado en ML?
3. ¿Con que precisión es identificado un *main connector* con pines opacos, y un *main connector* sin pines opacos?
4. ¿Qué porcentaje de fallas falsas disminuirá implementando ML en el área de producción, específicamente en la estación de P&F?

1.6.- Hipótesis

Mediante el uso de un sistema de visión artificial con ayuda de *Machine Learning* (ML), se puede mejorar el proceso de inspección con la identificación de *main connector* con opacidad en los pines y conteo del número de pines en la parte B, reduciendo el número de fallas de un 40.6% a un 10%, así como tener una mayor posibilidad de identificar pines dañados.

1.7.- Justificación

El desarrollo de este proyecto es de importancia, ya que permitirá un ahorro de $15,000 USD para la empresa, ya que con solo $5,000 USD se podría implementar un sistema de bajo costo basado en IA con ML en las estaciones de P&F, con la calidad suficiente para cumplir con los objetivos establecidos, contribuyendo a la reducción de quejas del cliente, mejorando el ciclo de prueba y reduciendo el porcentaje de fallas.

1.8.- Objetivo general

Desarrollar un sistema de visión con inteligencia artificial, mediante el uso de una red neuronal, el cual pueda mejorar la calidad en la detección de fallas falsas y conteo de pines de *main connector* para el proceso de Pin&Fakra.

1.9.- Objetivos específicos

1. El sistema deberá identificar el *main connector* mediante una red neuronal.
2. Entrenar la red neuronal con 100 imágenes de pines con opacidad en el *main connector*.
3. Entrenar la red neuronal con 100 imágenes de pines con suficiente brillo y sin opacidad en el *main connector*.
4. Diseñar una interfaz para realizar las pruebas del sistema de visión.
5. Desplegar en pantalla el tipo de conector identificado.
6. Desplegar en pantalla si el *connector* identificado es un conector con opacidad en los pines o sin opacidad.
7. Desplegar el porcentaje de exactitud en la identificación del conector.
8. Contar cada pin de la parte B del *main connector* y desplegar el conteo total.

1.10.- Alcances

Este sistema solamente se implementará para la estación P&F del producto de radio MP Refresh. La identificación de los conectores se realizará únicamente para el *main connector*.

El conteo de pines solamente se implementará en la parte B del *main connector*.

Este proyecto no está enfocado a identificar el daño físico de las antenas, del USB, o del *main connector*, o algún daño en el funcionamiento eléctrico del radio.

1.11.- Limitaciones

El tiempo establecido para la cotización del material a utilizar, el diseño, el desarrollo, la evaluación y la implementación del proyecto es de 5 meses.

Tiempo de máquina de 3 horas a la semana para pruebas.

1.12.- Métodos, técnicas y procedimientos

El procedimiento que se llevará a cabo para validar la hipótesis es el siguiente:

1. Análisis del sistema actual. Estación P&F sin algoritmo de IA para la identificación de conectores y el conteo de pines.
2. Búsqueda de información acerca de algoritmos existentes de identificación y reconocimiento de objetos.
3. Obtener información detallada acerca de los principales ofensores en fallas de la estación.
4. Implementación del algoritmo de identificación y detección de objetos. (Entrenamiento de Red Neuronal)
5. Crear DLL del algoritmo de identificación y detección de objetos.
6. Implementar el DLL en el código de inspección de estación P&F.
7. Probar el algoritmo con 100 imágenes de conectores.
8. Desplegar resultado de identificación y conteo de pines en Test Plan (Pruebas realizadas por los equipos).
9. Análisis de resultados.

CAPÍTULO II. MARCO TEÓRICO

2.1.- Inteligencia Artificial

En las últimas décadas han surgido varias definiciones de inteligencia artificial (IA), no obstante, nos gustaría presentar la definición que utiliza John McCarthy en su artículo “WHAT IS ARTIFICIAL INTELLIGENCE?” (McCarthy, Revised 2004), “Es la ciencia y la ingeniería de hacer maquinas inteligentes, especialmente programas de computadora inteligentes relacionado con la tarea similar de usar computadoras para comprender la inteligencia humana, pero la IA no tiene que limitarse a métodos que son biológicamente observables”. Sin embargo, décadas antes, en 1950 se publicó el artículo “COMPUTING MACHINERY AND INTELLIGENCE” por A.M. Turing (Turing, 1950), conocido como el padre la informática. En su artículo, Turing realiza la siguiente pregunta: "¿Pueden pensar las máquinas?", de aquí surge lo que conocemos como la prueba de Turing, en la que un interrogador humano trataría de distinguir entre una respuesta de texto humana y una computadora. Si bien esta prueba ha sido objeto de un gran escrutinio desde su publicación, sigue siendo una parte importante de la historia de la IA, así como un concepto continuo dentro de la filosofía, ya que utiliza ideas en torno a la lingüística.

Seguido de esto, Stuart Russell y Peter Norvig publicaron el libro llamado Inteligencia artificial: un enfoque moderno (Stuart Russell and Peter Norvig, 1995), el cual se convirtió en uno de los principales materiales de estudio de la IA. En el libro se profundiza en 4 definiciones de la IA, los cuales diferencian los sistemas informáticos sobre la base de la racionalidad y el pensamiento frente a la actuación:

Enfoque humano:

- Sistemas que piensan como humanos
- Sistemas que actúan como humanos

Enfoque ideal:

- Sistemas que piensan racionalmente

- Sistemas que actúan racionalmente

La definición de Alan Turing habría entrado en la categoría de sistemas que actúan como personas.

En pocas palabras, la IA es un campo que logra combinar la informática con datos solidos que nos permiten la resolución de problemas, la cual abarca subcampos de aprendizaje automático (*Machine Learning*) y aprendizaje profundo (*Deep Learning*), cuyos conceptos son mencionados con frecuencia junto con la inteligencia artificial. Los algoritmos de *Machine Learning* y *Deep Learning* tienen como objetivo buscar sistemas expertos que hagan predicciones o clasificaciones basados en datos de entrada.

2.1.1.- Tipos de Inteligencia Artificial

La inteligencia artificial en el área de la tecnología de la información para hacer referencia al nivel de desarrollo de las máquinas suele clasificarse en dos tipos, inteligencia artificial débil, e inteligencia artificial fuerte.

2.1.1.1.- Inteligencia Artificial Débil

La inteligencia artificial estrecha o débil (ANI), es una inteligencia entrenada y enfocada en la realización de tareas específicas. Esta IA impulsa la mayor parte de IA que nos rodea, permite algunas aplicaciones muy sólidas, como Siri de Apple, Alexa de Amazon, Watson de IBM y vehículos autónomos.

2.1.1.2.- Inteligencia Artificial Fuerte

Esta inteligencia se compone de inteligencia artificial general (AGI) y de super inteligencia artificial (ASI). La IA general, es una forma teórica de IA en la que una máquina tendría una inteligencia igual a la de los humanos; tendría una conciencia autoconsciente que tiene la capacidad de resolver problemas, aprender y planificar

para el futuro. Mientras que la ASI superaría la inteligencia y la capacidad del cerebro humano. Si bien la IA fuerte sigue siendo completamente teórica y no tiene ejemplos prácticos en uso en la actualidad, eso no significa que los investigadores de IA no estén explorando su desarrollo.

2.1.2.- Deep learning vs machine learning

El aprendizaje profundo (DL) y el aprendizaje automático (ML) suelen utilizarse indistintamente, por ello es necesario resaltar los matices que existen entre ellos, como se menciona con anterioridad ambos son subcampos de la IA, y el aprendizaje profundo es en realidad un subcampo del aprendizaje automático, como podemos ver en la Figura 2.1.

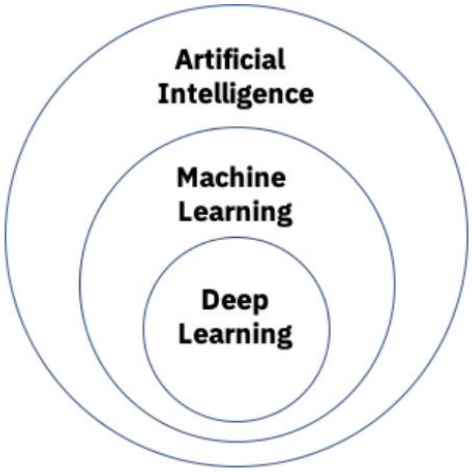

Figura 2.1 Campos del aprendizaje automático: AI, ML y DL

El aprendizaje profundo se compone de redes neuronales. El aprendizaje “profundo” (Deep Learning) se refiere a una red neuronal compuesta por más de tres capas, que incluirían las entradas y las salidas, puede considerarse un algoritmo de

aprendizaje profundo. Esto generalmente se representa usando el diagrama de la Figura 2.2.

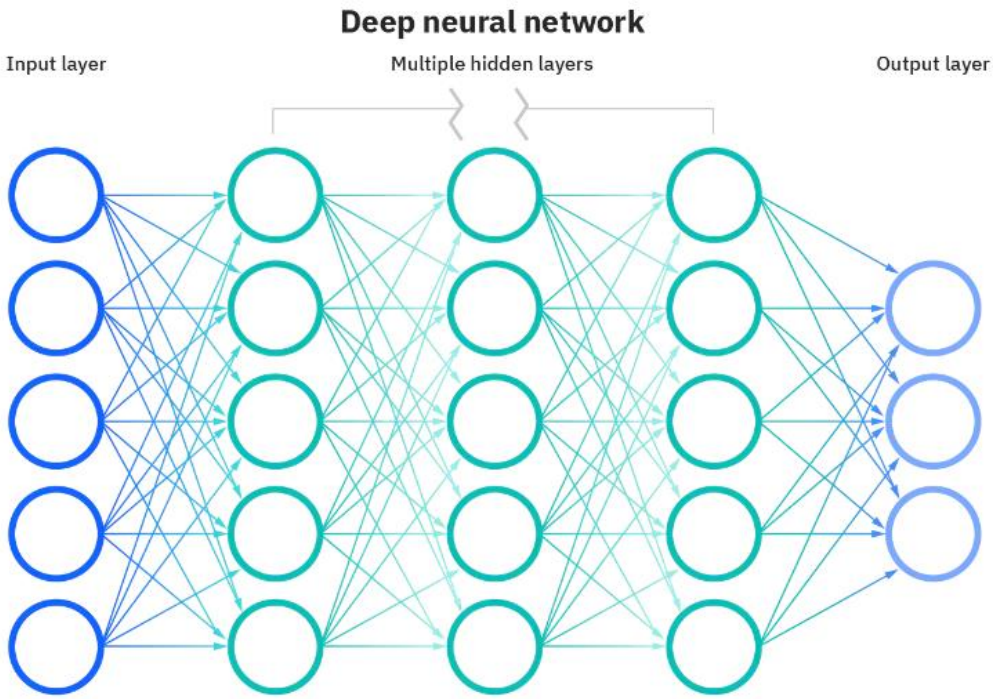

Figura 2.2 Diagrama general de aprendizaje profundo

La manera en la que el ML y el DL difieren es cómo es que aprenden el algoritmo. El aprendizaje profundo automatiza gran parte del proceso de extracción de características, lo que elimina parte de la intervención humana requerida de forma manual y permite el uso de conjuntos de datos más grandes. Se puede pensar en el Deep Learning como un aprendizaje automático escalable, mientras que en el aprendizaje automático "clásico" depende más de la intervención humana para aprender. En este caso son los humanos con experiencia quienes determinan la jerarquía de las características para comprender qué diferencias hay entre las entradas de datos. Esto requiere de datos más estructurados para que la red neuronal logre un aprendizaje óptimo. (Education, 2020)

Deep learning puede aprovechar que los datos estén etiquetados, a esto también se le conoce como aprendizaje supervisado, para informar su algoritmo. Puede ingerir datos no estructurados en su forma sin procesar, por ejemplo, texto e imágenes, y puede determinar automáticamente la jerarquía de características que distinguen las diferentes categorías de datos entre sí. A diferencia del aprendizaje automático, no requiere intervención humana para procesar los datos. (Education, 2020)

2.1.3.- Aplicaciones de la Inteligencia Artificial

En la actualidad existen numerosas aplicaciones de la IA en el mundo real, a continuación, se presentan los ejemplos más comunes:

a) Reconocimiento de voz: también se conoce como reconocimiento automático de voz (ASR), reconocimiento de voz por computadora o conversión de voz a texto, y es una capacidad que utiliza el procesamiento de lenguaje natural (NLP) para procesar el habla humana en un formato escrito. Muchos dispositivos móviles incorporan reconocimiento de voz en sus sistemas para realizar búsquedas por voz.
b) Servicio al cliente: los agentes virtuales en línea están reemplazando a los agentes humanos a lo largo del recorrido del cliente. Responden preguntas frecuentes (FAQ) sobre temas, como el envío, o brindan asesoramiento personalizado, productos de venta cruzada o sugerencias de tamaños para los usuarios, cambiando la forma en que pensamos sobre la participación del cliente en los sitios web y las plataformas de redes sociales. Los ejemplos incluyen bots de mensajería en sitios de comercio electrónico con agentes virtuales.
c) Visión por computadora: esta tecnología de inteligencia artificial permite que las computadoras y los sistemas obtengan información significativa

de imágenes digitales, videos y otras entradas visuales, y en función de esas entradas, puede tomar medidas. Impulsada por redes neuronales convolucionales, la visión por computadora tiene aplicaciones dentro del etiquetado de fotos en las redes sociales, imágenes de radiología en el cuidado de la salud y automóviles autónomos dentro de la industria automotriz.

d) Motores de recomendación: utilizando datos de comportamiento de consumo pasados, los algoritmos de IA pueden ayudar a descubrir tendencias de datos que se pueden usar para desarrollar estrategias de venta cruzada más efectivas. Esto se utiliza para hacer recomendaciones de complementos relevantes a los clientes. (Education, 2020)

2.1.4.- Red DNN + ResNet50

ResNet es la abreviatura de *Residual Networks*, que es una red neuronal clásica utilizada como columna vertebral para gran parte de las tareas de visión artificial. Este modelo fue el ganador del desafío ImageNet en 2015 (Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei, 2015). ImageNet es una base de datos de imágenes organizada según la jerarquía de WordNet, en la que cada nodo de la jerarquía está representado por cientos y miles de imágenes. El proyecto ha sido fundamental en el avance de la visión por computadora y la investigación de aprendizaje profundo. Los datos están disponibles de forma gratuita para los investigadores para uso no comercial. El avance con ResNet fue que permitió entrenar redes neuronales profundas (DNN o Deep Neural Network) con hasta 150 capas. AlexNet, el ganador de ImageNet 2012 (Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei, 2015) y el modelo que aparentemente inició el enfoque en el aprendizaje profundo tenía solo

8 capas convolucionales, la red VGG tenía 19 e Inception o GoogleNet tenía 22 capas y ResNet152 tenía 152 capas como podemos notar en la Figura 2.3. Una ResNet50 es una versión más pequeña de ResNet152.

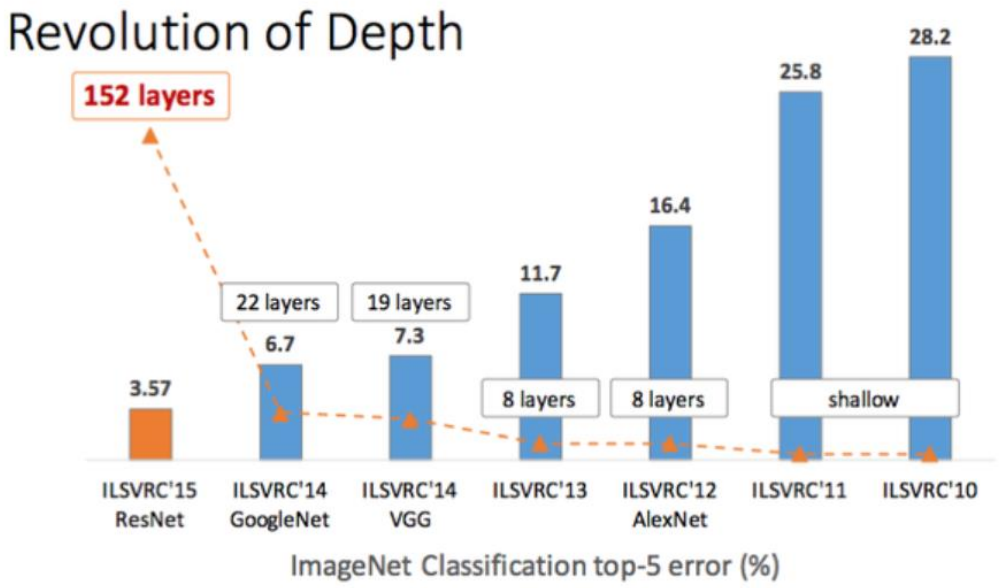

Figura 2.3 Revolución de profundidad en capas

El artículo Deep Residual Learning for Image Recognition de Microsoft Research (Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015) hace mención que las redes neuronales más profundas son más difíciles de entrenar y se presenta un marco de aprendizaje residual para facilitar el entrenamiento de redes que son sustancialmente más profundas. En sus experimentos se reformulan explícitamente las capas como funciones residuales de aprendizaje con referencia a las entradas de la capa, en lugar de aprender funciones no referenciadas. También brinda la evidencia que muestra que estas redes residuales son más fáciles de optimizar y pueden ganar precisión a partir de una profundidad considerablemente mayor. Un conjunto de estas redes residuales logra un error del 3.57 % en el conjunto de prueba de ImageNet.

2.2.- Herramientas utilizadas

Las herramientas presentadas a continuación tienen como finalidad el desarrollo del proyecto.

2.2.1.- Hardware

El Hardware utilizado consta de los elementos que se describen en las siguientes secciones.

2.2.1.1.- Cámara Mako G-319 PoE

La cámara Mako G-319 (Figura 2.4) utiliza el sensor Sony IMX265, el cual ejecuta 37,0 fotogramas por segundo a una resolución de 3,2 MP. Es compatible con GigE Vision, contiene una carcasa industrial compacta y resistente. Incluye funcionalidades avanzadas como el protocolo de tiempo de precisión (PTP), los comandos de acción Trigger over Ethernet (ToE) y Power over Ethernet (PoE).

Figura 2.4 Cámara Mako G-319

Esta cámara cuenta con las especificaciones que se muestran en la Tabla 2.1.

Tabla 2.1 Especificaciones de la cámara Mako G-319

Interfaz	IEEE 802.3 1000BASE-T, IEEE 802.3af (PoE)
Resolución	2064 (H) × 1544 (V)
Sensor	Sony IMX265
Tipo de Sensor	CMOS
Tamaño del Sensor	Type 1/1.8
Tamaño del Pixel	3.45 µm × 3.45 µm
Modo de Obturador	Obturador global
Montura de lentes (disponibles)	C-Mount, CS-Mount
Max. velocidad de fotogramas a máxima resolución	37 fps
ADC	12 Bit
Búfer de Imágen (RAM)	64

Es adecuada para todas las aplicaciones típicas en visión artificial, como, por ejemplo: robótica, control de calidad, machine visión, logística, entre otras. Las características que tiene esta cámara en los datos de salida son:

a. Profundidad de bits de 8/12 Bit.
b. Formatos de píxeles monocromáticos: Mono8, Mono12, Mono12Packed.
c. Formatos de píxeles sin procesar: BayerRG8, BayerRG12, BayerRG12Packed.

2.2.1.2.- Lente TC23064 - Bi-telecentric

Los lentes telecéntricos suelen representar un punto clave en los sistemas de medición alimentados mediante visión artificial. Este lente puede aprovechar sensores de alta resolución como el mostrado en la Figura 2.4, adquiriendo imágenes con fidelidad y precisión. Dentro de las ventajas de este lente (Figura 2.5) podemos encontrar las siguientes:

a. Alta telecentricidad para imágenes de objetos gruesos.
b. Distorsión casi nula para mediciones precisas.
c. Excelente resolución para cámaras de alta resolución.
d. Diseño simple y robusto para entornos industriales.
e. Fácil inserción de filtros.

Figura 2.5 Lente TC23064

Las especificaciones ópticas, del field of view (campo de visión) y mecánicas del TC23064 las podemos encontrar en las Tablas 2.2, 2.3 y 2.4.

Tabla 2.2 Especificaciones ópticas

Magnificación	(x)	0.138
Diámetro del círculo de la imagen	(mm)	11.0
Tamaño máximo del sensor		2/3″
Distancia de trabajo	(mm)	181.8
Número f de trabajo (*f*/N)		8
Telecentrismo típico (max)	(deg)	< 0.05 (0.08)
Distorsión típica (max)	(%)	< 0.03 (0.07)
Profundidad de campo	(mm)	21.70

Tabla 2.3 Especificaciones del campo de visión

Sensores	Unidad de medida	Campo de Visión
1/3”	(mm × mm)	34.78 x 26.09
1/2.5”	(mm × mm)	41.30 x 31.01
1/2”	(mm × mm)	46.38 x 34.78
1/1.8”	(mm × mm)	51.67 x 38.62
2/3”	(mm × mm)	61.59 x 51.38

Tabla 2.4 Especificaciones mecánicas

Montaje		C
Ajuste de fase		No
Longitud	(mm)	245.5
Diámetro frontal	(mm)	100
Masa	(g)	1026

2.2.1.3.- Lámpara HPR2-100SW

La High-Power Ring Lights serie2 (HPR2, Figura 2.6) es una lampara de anillo que emite luz blanca y tiene un diámetro exterior de 116 mm, un diámetro interior de 66 mm y una altura de 26.4 mm. El voltaje de alimentación es de 24VDC. Proporciona luz difusa de los LED sin desperdicio utilizando un mecanismo de iluminación único.

Incluso si se cambia la distancia de la pieza de trabajo a la unidad de luz, hay poca variación en la región uniforme, por lo tanto, se puede utilizar para una amplia variedad de usos.

Figura 2.6 Lámpara HPR2

2.2.1.4.- NETGEAR Ethernet Switch GS305PP

Switch no administrado gigabit de 5 puertos con 4 puertos PoE+ y la cantidad de energía de 83W para alimentar puntos de acceso inalámbricos, IP de cámaras y de teléfonos. Cuenta con una configuración simple, sin software para instalar y sin mantenimiento. La tecnología PoE equilibra automáticamente la potencia PoE al nivel más granular en cada puerto, según las necesidades del dispositivo, como podemos ver en la Figura 2.7.

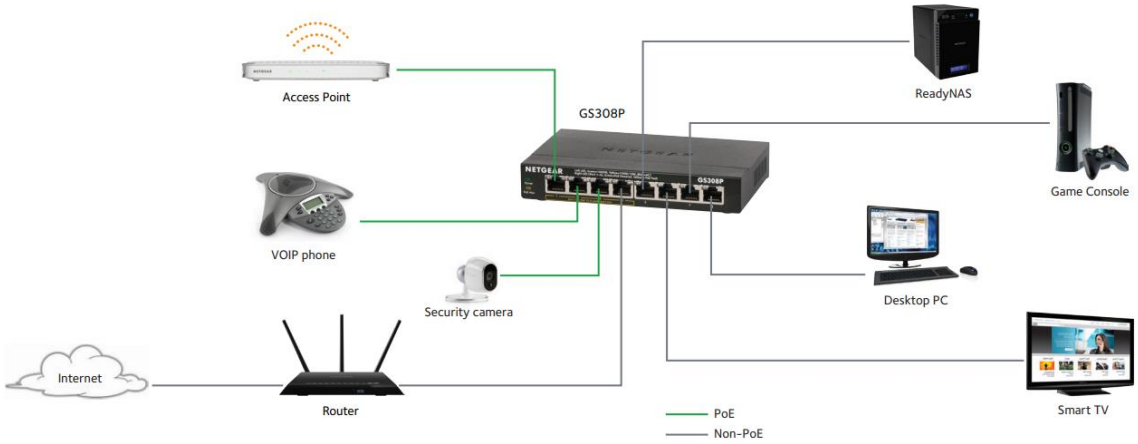

Figura 2.7 Ejemplo de conexión del Switch Ethernet

2.2.1.5.- Cable GigE Cat6, S/FTP, 2xRJ-45

Cable GigE (Figura 2.8) para trasmisión de datos, contiene cable trenzado y blindado con un conector de bloqueo de clic RJ-45 en ambos extremos.

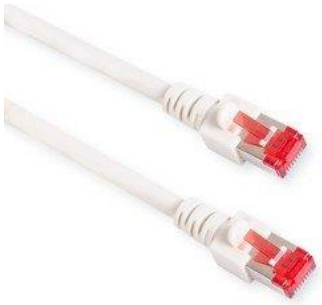

Figura 2.8 Cable GigE Cat6

2.2.1.6.- XP Power Supply VCS100US24

Fuente de alimentación industrial AC-DC de la serie VCS de XP (Figura 2.9). Tiene salida única y cuenta con montaje en chasis para aplicaciones industriales, se adaptan a aplicaciones que requieren una fuente de alimentación cerrada con terminales de tornillo. Contiene un control de ajuste de voltaje accesible para el usuario, proporciona un ajuste de ±10 % del voltaje de salida, permitiendo el uso de voltajes no estándar o para compensar las pérdidas del cable. Estas unidades altamente eficientes enfriadas por convección no requieren disipación de calor adicional ni flujo de aire forzado, lo que ahorra espacio y costos adicionales. Las especificaciones las podemos encontrar en la Tabla 2.5.

Figura 2.9 Power Supply VCS1000US24

Tabla 2.5 Especificaciones power supply VCS100US24

Atributos del Producto	Valores del Atributo
Configuración	Incluida
Dimensiones	6.26L x 3.87" W x 1.65" H (159.0mm x 98.2mm x 42.0mm)
Eficiencia	0.87
Voltaje de Entrada	100 VAC; 110 VAC; 115 VAC; 120 VAC; 127 VAC; 190 VAC; 200 VAC; 220 VAC; 230 VAC; 240 VAC
Rango del Voltaje de Entrada	90-264 VAC
Voltaje de Insolación	3kV (3000V)
Tipo de Montaje	Montaje en panel

Numero de Salidas	1 salida
Operación	Interruptor
Corriente de Salida	4.17 A
Potencia de Salida	100 W
Voltaje de Salida	24 VDC
Corriente de Pico	4.17A
Tipo Principal	AC-DC
Serie	Series VCS
Características Especiales	Voltaje de salidad ajustable; Capacidad de entrada DC
Tipo	Empotrada

2.2.1.7.- SeaLevel 420U

La SeaI/O-420U (Figura 2.10) es un controlador y monitor de 16 entradas ópticamente aisladas y 8 salidas de relé de forma C (SPDT) a través de conexión USB. Las entradas pueden oscilar entre 5 y 30 V CC, mientras que los relés de forma C cambian hasta 60 VCC a 2 A. Las entradas y salidas se agrupan en segmentos de cuatro bits que comparten un común para facilitar el cableado a través de bloques de terminales extraíbles de 3,5 mm. El SeaI/O-420U se alimenta de su fuente de 9-30 VCC. La comunicación con el SeaI/O-420U es usando el protocolo Modbus RTU estándar de la industria o usando las bibliotecas de software Sealevel SeaMAX API de su programa de aplicación. Los controladores y las utilidades del software SeaMAX de Sealevel facilitan la instalación y el funcionamiento con los sistemas operativos Microsoft Windows y Linux. Las especificaciones de la SeaI/O – 420U las podemos encontrar en la Tabla 2.6.

Figura 2.10 SeaI/O-420U

Tabla 2.6 Especificaciones SeaI/O-420U

Características técnicas	
Familia	SeaI/O
Tiempo de rebote de contacto	7ms max.
Corriente del contacto	2A max.
Tiempo de operación del contacto	2ms max.
Tiempo de liberación del contacto	1ms max.
Voltaje del contacto	60VDC max.
Digital I/O	Salidas de relé de forma C, entradas aisladas
Dimensiones	7.5 (L) x 5.1 (W) x 1.3 (H)
Temperatura extendida	Llame para opciones
Rango de humedad	10 – 90% de humedad relativa, sin condensación
Aislamiento de entrada	300V
Rango de entrada	5-30 VDC
Interfaz(es) de host	USB
Número de entradas/salidas	16 entradas / 8 salidas
Temperatura de funcionamiento	0°C a 70°C (32°F a 158°F)
# de Puertos	16/8
Requisito de energía	9-30 VDC @ 2.2W
Cumple con RoHS	Sí
Temperatura de almacenamiento	-50°C a 105°C (-58°F a 221°F)
Especificación USB	1.1 Compatible con 2.0

2.2.1.8.- HP ProDesk 600 G6 – mini desktop

La Hp ProDesk 600 G6 (Figura 2.11) es una computadora de escritorio compacta, potente rendimiento y seguridad integrada en un factor de forma altamente adaptable. Cuenta con Windows10 Pro. Dentro de sus características podemos encontrar un rápido rendimiento, conectividad y capacidad de respuesta, debido a su procesador Intel® Core™ de 10.ª generación. Las especificaciones esenciales las podemos ver en la Tabla 2.7.

Figura 2.11 HP ProDesk 600 G6

Tabla 2.7 Especificaciones HP ProDesk 600 G6

Características Técnicas	
Sistema operativo	El dispositivo viene con Windows 10 y una actualización gratuita de Windows 11 o puede estar precargado con Windows 11.
Procesador	Intel® Core™ i5-10500T (Frecuencia base de 2.3 GHz, hasta 3.8 GHz con tecnología Intel® Turbo Boost, Caché L3 de 12 MB, 6 cores)
Memoria	8 GB DDR4-2666 MHz RAM (1 x 8 GB)
Disco duro	256 GB PCIe® NVMe™ TLC M.2 SSD

2.2.2.- Software

Los Softwares utilizados se presentan a continuación.

2.2.2.1.- SeaMax para Windows v3.5.0

SeaMax v3.5.0 es el software utilizado para el control de la SeaLevel 420U. Los drivers y las utilidades del software SeaMAX de Sealevel facilitan la instalación y el funcionamiento con los sistemas operativos Windows. Es compatible con las versiones de 32 y 64 bits de Windows 10, Windows 8.2 y 7.

2.2.2.2.- NI Max

NI Measurement & Automation Explorer (NI Max) proporciona acceso a su hardware de NI. Es un software gratuito que no se puede descargar por sí solo, pero se incluye y se instala automáticamente con todos los drivers de NI (NI-VISA, NI-DAQmx, etc.) y la configuración del sistema de NI. Con NI Max se puede:

- Configurar hardware y software de NI
- Crear y editar canales, tareas, interfaces, escalas e instrumentos virtuales (ej. Crear tareas para dispositivos NI-DAQmx en NI MAX)
- Ejecutar diagnósticos del sistema y ejecutar paneles de prueba (p. ej., usar paneles de prueba en el Explorador de medición y automatización para dispositivos compatibles con NI-DAQmx)
- Ver dispositivos/instrumentos conectados al sistema y software instalado en el sistema. (What is NI Measurement & Automation Explorer (NI MAX), 2022):

2.2.2.3.- Microsoft Visual Studio 2022

Es un entorno de desarrollo integrado (IDE, Integrated Development Environment) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC.

Un entorno de desarrollo integrado (IDE) es un programa con numerosas características que respalda muchos aspectos del desarrollo de software. El IDE de Visual Studio es un panel de inicio creativo que se puede usar para editar, depurar y compilar código y después publicar una aplicación. Aparte del editor y el depurador estándar que proporcionan la mayoría de IDE, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más características para facilitar el proceso de desarrollo de software. (Terry G. Lee, Gordon Hogenson, 2022)

El IDE de Visual Studio 2022 cuenta las siguientes características:

1. Productivo: Escalar para trabajar en proyectos de cualquier tamaño y complejidad con un IDE de 64 bits. Codificar con un nuevo editor de Razor que pueda refactorizar entre archivos. Diagnosticar problemas con visualizaciones para operaciones asincrónicas y analizadores automáticos.
2. Moderno: Desarrollar aplicaciones multiplataforma para dispositivos móviles y de escritorio con .NET MAUI. Crear interfaces de usuario web con capacidad de respuesta en C# con Blazor. Compilar, depurar y probar aplicaciones de .NET y C++ en entornos de Linux. Puede usar funcionalidades de recarga activa en aplicaciones .NET y C++. Edita las páginas de ASP.NET en ejecución en la vista de diseñador web.
3. Innovador: Finalizaciones de código con tecnología de IA. Trabajar juntos en tiempo real con sesiones de codificación compartidas. Clonar repositorios, navegar por elementos de trabajo y organización de líneas

individuales para confirmaciones. Configurar automáticamente los flujos de trabajo de CI/CD que se pueden implementar en Azure.

Esta versión de Visual Studio es de 64 bits, facilita el trabajo con proyectos aún más grandes y cargas de trabajo más complejas. Cuenta con IntelliCode, que es un poderoso conjunto de herramientas de finalización automática de código que comprenden el contexto del código: nombres de variables, funciones y el tipo de código que está escribiendo. Esto hace que IntelliCode pueda completar hasta una línea completa a la vez, lo que lo ayuda a programar de forma más precisa y segura (Microsoft, Visual Studio 2022, 2022).

2.2.2.4.- NI Vision Development Module

El Vision Development Module (VDM) es una herramienta que ayuda a desarrollar software e implementar aplicaciones de visión artificial y procesamiento de imágenes. Se puede usar con el entorno de programación gráfica LabVIEW, C, C++, y C# para sistemas Windows y LabVIEW en sistemas en tiempo real. El módulo incluye IP para procesadores y FPGA, cuenta con importadores de modelos para realizar inferencia usando modelos de aprendizaje profundo desarrollados en TensorFlow. (¿Qué es el Módulo Vision Development?, 2022)

Cuenta con una extensa biblioteca de funciones, en las que se puede tener acceso a cientos de algoritmos de procesamiento de imágenes y funciones de visión artificial para mejorar imágenes, verificar presencia, ubicar características, identificar objetos, medir partes y más. VDM también incluye el Vision Assistant, el cual es una herramienta de diseño de algoritmos que simplifica el diseño del sistema de visión, ayudando a desarrollar algoritmos para implementar en CPUs o FPGAs.

Se puede elegir el hardware adecuado para su aplicación y configurar cámaras, adquirir imágenes y analizar resultados de inspección para desarrollar sistemas de visión artificial totalmente personalizados.

De igual forma el VDM puede ayudar a resolver retos complejos de visión artificial ya que cuenta con motores de interferencia de aprendizaje profundo, con la ayuda e implementación de modelos de aprendizaje profundo desarrollados en TensorFlow, TensorFlow es una herramienta de Google, líder en la industria para desarrollo de aprendizaje profundo de fuente abierta.

2.2.2.5.- ML.Net

ML.NET es un marco de machine learning gratuito, de código abierto y multiplataforma creado específicamente para desarrolladores de .NET. Se pueden desarrollar e integrar modelos de machine learning personalizados en sus aplicaciones .NET, sin necesidad de experiencia previa en aprendizaje automático. ML.Net es una plataforma extensible, con herramientas en Visual Studio. (Microsoft, .NET Machine Learning & AI, 2022)

ML.NET brinda la capacidad de agregar aprendizaje automático a las aplicaciones .NET, ya sea en escenarios en línea o fuera de línea. Con esta capacidad, puede realizar predicciones automáticas utilizando los datos disponibles para su aplicación. Las aplicaciones de aprendizaje automático utilizan patrones en los datos para hacer predicciones en lugar de tener que programarlas explícitamente. Central para ML.NET es un modelo de aprendizaje automático. El modelo especifica los pasos necesarios para transformar datos de entrada en una predicción. Con ML.NET, puede entrenar un modelo personalizado especificando un algoritmo, o se pueden importar modelos TensorFlow y ONNX previamente entrenados. Una vez que se tenga un modelo, puede ser agregado a la aplicación para hacer las predicciones. Se ejecuta en Windows, Linux y macOS con .NET Core o Windows con .NET Framework. 64 bits es compatible con todas las plataformas. 32 bits es compatible con Windows, excepto para la funcionalidad relacionada con TensorFlow, LightGBM y ONNX. (Luis Quintanilla, David Coulter, Nick Schonning, 2021)

Ejemplos del tipo de predicciones que puede realizar con ML.NET:

1. Clasificación/Categorización: Divide automáticamente los comentarios de los clientes en categorías positivas y negativas.
2. Regresión/Predecir valores continuos: Prediga el precio de las casas según el tamaño y la ubicación.
3. Detección de anomalías: Detectar transacciones bancarias fraudulentas.
4. Recomendaciones: Sugerir productos que los compradores en línea pueden querer comprar, en función de sus compras anteriores.
5. Serie temporal/datos secuenciales: Pronosticar el clima/ventas de productos.
6. Clasificación de imágenes: Categorizar patologías en imágenes médicas.

2.2.2.5.1.- Flujo de trabajo del código (Code Workflow)

El siguiente diagrama (ver Figura 2.12) representa la estructura del código o flujo de trabajo del código de la aplicación de ML.Net, así como el proceso iterativo de desarrollo del modelo:

- Recopilar y cargar datos de entrenamiento en un objeto IDataView.
- Especificar una canalización (pipeline) de operaciones para extraer funciones y aplicar un algoritmo de machine learning.
- Entrenar un modelo llamando a Fit () en el pipeline.
- Evaluar el modelo e iterar para mejorar.
- Guardar el modelo en formato binario, para usarlo en una aplicación.
- Volver a cargar el modelo en un objeto ITransformer.
- Hacer predicciones llamando a CreatePredictionEngine.Predict().

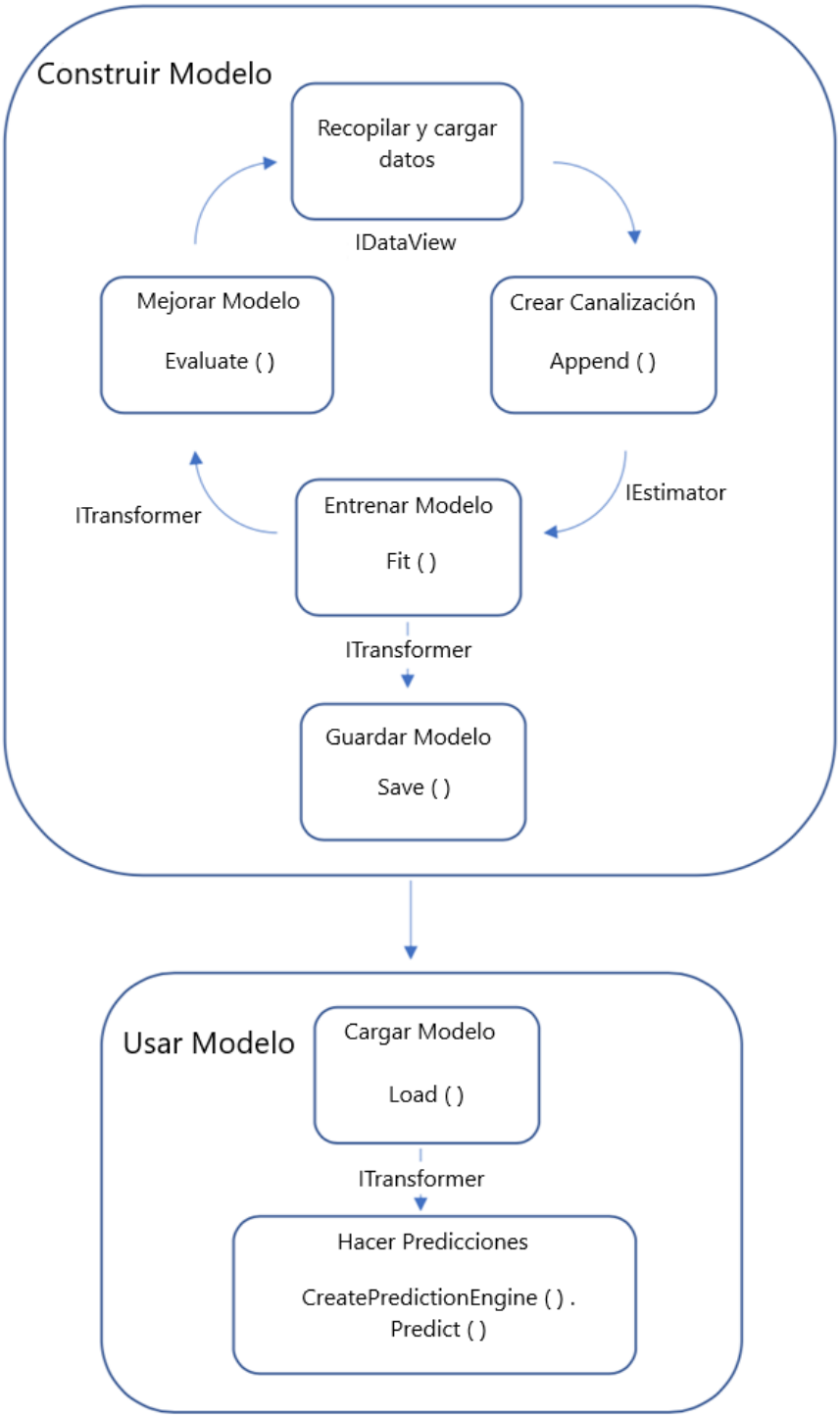

Figura 2.12 Code workflow ML.Net

2.2.2.6.- LabWindows/CVI 2013

LabWindows/CVI es un entorno de desarrollo de software ANSI C con un extenso juego de herramientas de programación para crear aplicaciones de pruebas y medidas, ayuda a crear aplicaciones de ingeniería personalizadas. Se puede utilizar para administrar proyectos, editar y depurar código fuente, desarrollar una interfaz de usuario y probar salida de código y rendimiento en un espacio de trabajo simplificado y seccionado. Incluye herramientas para depuración avanzada, documentación del código e implementación de sistemas, de esta manera se pueden integrar sistemas de administración de datos, requerimientos y control de código fuente. Este Software facilita la rápida adquisición de datos desde instrumentos GPIB, USB, serial, Ethernet, PXI, VXI y FPGA usando las bibliotecas integradas de E/S de instrumentos, controladores de instrumentos integrados. (¿Qué es LabWindows™/CVI?, 2022)

LabWindows/CVI proporciona las herramientas y los componentes para crear interfaces de usuario personalizables destinadas a aplicaciones de pruebas y medidas automatizadas, y necesarias para crear experiencias de usuario informativas e intuitivas. Con las funciones de clic y arrastre, puede personalizar fácilmente los componentes de la Interfaz de Usuario que se esté desarrollando.

En la Tabla 2.8 podemos ver las versiones de LabWindows/CVI del 2010, hasta la versión más actual.

Para este proyecto se estará utilizando la versión LabWindows / CVI 2013 SP1. Los sistemas operativos que son originalmente compatibles con esta versión los podemos encontrar en la Tabla 2.9.

Tabla 2.8 Versiones de LabWindows/CVI

Nombre / versión	Número de compilación	Fecha	Notas y soporte del sistema operativo
LabWindows / CVI 2010	10	2010	Soporte Linux
LabWindows / CVI 2010 SP1			
LabWindows / CVI 2012	12,0	2012	
LabWindows / CVI 2012 SP1			
LabWindows / CVI 2013	13,0	2013	Compilador cambiado a Clang 2.9. Nuevo depurador ejecutándose en su propio proceso.
LabWindows / CVI 2013 SP1			
LabWindows / CVI 2013 SP2			
LabWindows / CVI 2015	15	2015	actualizar a Clang 3.3
LabWindows / CVI 2015 SP1	15,1	2016	
LabWindows / CVI 2017	17	2017	Puntos de seguimiento, resaltado de palabras / semánticas, puntos de interrupción específicos del hilo, comentar / descomentar
LabWindows / CVI 2019	19,0	Mayo de 2019	Actualizaciones del editor de código Source: zoom, fragmentos de código, ediciones de varias líneas
LabWindows / CVI 2020	20,0	Sep. De 2020	Soporte UTF-8

Tabla 2.9 SO Soportado originalmente para LabWindows/CVI 2013 SPI

SO Soportado
Windows Server 2008 R2 64-bit
Windows XP (SP3) 32-bit
Windows Server 2003 R2 32-bit
Windows 8.1
Windows 8
Windows 7
Windows Vista (SP1) 32-bit
Windows Server 2012 R2 64-bit
Windows Vista (SP1) 64-bit

CAPÍTULO III. ANÁLISIS

En este capítulo se realizó el análisis de los requerimientos del sistema, así como especificar los usuarios del sistema a desarrollar.

El análisis de los requerimientos es el estado más importante y fundamental dentro del SDLC (Software Development Life Cycle), una vez hecho el análisis de los requerimientos el siguiente paso es claramente definir y documentar cada uno de los requerimientos.

3.1.- Usuarios del sistema

Personas adultas entre 18 y 45 años con puestos de técnicos de calidad, técnicos de sistemas de pruebas e ingenieros de sistemas de pruebas.

3.2.- Requerimientos

En las tablas 3.1 y 3.2 se presentan los requerimientos del sistema para la Interfaz de Usuario (IU), así como la adquisición, procesamiento, e identificación de imagen del tipo de conector, y el despliegue de resultados en IU. Los requerimientos se dividieron en dos módulos, módulo de interfaz de usuario y módulo de clasificación de imagen y conteo de pines.

Tabla 3.1 Requerimientos del módulo interfaz de usuario

ID del Requerimiento	Descripción del Requerimiento	Categoría
IU 1.0	La interfaz deberá contar con un panel de herramientas de captura de imagen: control de iluminación, disparo (trigger) de adquisición de imagen, y limpiado de panel de imagen.	Evidente
IU 1.1	La IU deberá tener un botón para el encendido/apagado de la lámpara.	Evidente

IU 1.2	La IU tendrá un botón para activar el disparador (trigger) de la cámara y obtener la imagen.	Evidente
IU 1.3	La IU deberá tener un botón que permita limpiar el panel donde se muestre la imagen capturada.	Evidente
IU 1.4	La interfaz deberá contar con un panel donde se muestre la imagen identificada.	Evidente
IU 1.5	La interfaz deberá mostrar un panel con las pruebas a realizar con la imagen: clasificación, conteo de pines.	Evidente
IU 1.6	La IU mostrara un panel con los resultados obtenidos en la prueba.	Evidente
IU 1.7	El panel de pruebas deberá contar con un botón para la clasificación de la imagen.	Evidente
IU 1.8	El panel de prueba contendrá además del botón de clasificación, un botón para el conteo de pines.	Evidente
IU 1.9	El panel de resultados deberá contener dos cajas de texto que mostraran el porcentaje de exactitud de imagen buena, o el porcentaje de exactitud de imagen con opacidad en los pines.	Evidente
IU 1.10	El panel de resultados contendrá una caja de texto adicional que muestre el conteo de pines de la imagen clasificada como buena.	Evidente
IU 1.11	La IU deberá contar con una ventana de advertencia si se identifica una imagen con opacidad en los pines.	Evidente
IU 1.12	La interfaz deberá contar con un botón de "Salir" para terminar la utilización de la interfaz.	Evidente
IU 1.13	El botón de “Salir” deberá apagar la lámpara en dado caso esta esté encendida.	Evidente/ Oculto
IU 1.14	Al ejecutar la interfaz deberá mostrar un mensaje en el que mencione si la SeaLevel 420U no pudo ser inicializada.	Evidente
IU 1.15	Al inicializar la cámara la IU deberá mostrar un mensaje en pantalla mencionando la inicialización de cámara Mako.	Evidente
IU 1.16	Al presionar el botón de disparo de la cámara (trigger) en el panel de herramientas de imagen deberá aparecer en pantalla una ventana indicando que la imagen ha sido capturada sin ningún problema.	Evidente
IU 1.17	Al adquirir la imagen, ésta deberá guardarse en una dirección previamente establecida en formato .png.	Oculto

Tabla 3.2 Requerimientos del módulo clasificación de imagen y conteo de pines

ID del Requerimiento	Descripción del Requerimiento	Categoría
CI 1.0	El sistema deberá adquirir la imagen a identificar de una dirección predeterminada al presionar el botón “Clasificar”.	Oculto
CI 1.1	El sistema deberá ejecutar las librerías de ML.Net de algoritmo de clasificación de imágenes.	Oculto
CI 1.2	Las librerías de clasificación de imágenes deberán estar en la misma dirección de la IU.	Oculto
CI 1.3	El sistema clasificará el conector por medio de un algoritmo basado en machine learning.	Oculto
CI 1.4	Al presionar el botón conteo de pines deberá ejecutarse el algoritmo desarrollado para el conteo por medio de Vision Assistant.	Oculto

CAPÍTULO IV. DISEÑO

4.1.- Diseño de la Interfaz de Usuario

Vivimos en un mundo de productos de alta tecnología, y virtualmente todos ellos, por ejemplo, equipos electrónicos para el consumidor, equipo industrial, sistemas corporativos, sistemas militares, software de computadoras personales y webapps, requieren interacción humana. Si un producto ha de alcanzar éxito requiere buena usabilidad: medición cualitativa de la facilidad y eficiencia con la que un humano emplea las funciones y características que ofrece el producto de alta tecnología. La usabilidad importa, ya sea que una interfaz haya sido diseñada para un reproductor de música digital o para el sistema de control de armas de un avión de combate. Si los mecanismos de la interfaz están bien diseñados, el usuario se desliza por la interacción a un ritmo suave que hace que el trabajo se realice sin esfuerzo. Pero si la interfaz fue mal concebida, el usuario avanza y retrocede, y el resultado final es frustración y poca eficiencia en el trabajo.

¿Por qué es importante? Si el software es difícil de usar, fuerza al usuario a cometer errores, o si frustra sus esfuerzos para alcanzar las metas, entonces no le gustará, sin que importe el poder computacional que tenga, el contenido que entregue o las funciones que ofrezca. La interfaz tiene que estar bien hecha porque moldea la percepción que el usuario tiene del software. (Diseño de la interfaz de usuario, 2010)

El diseño de la interfaz de usuario comienza con la identificación de los requerimientos del usuario, la tarea y el ambiente. Una vez identificadas las tareas del usuario, se crean y analizan los escenarios para éste y se define un conjunto de objetos y acciones de la interfaz. Esto forma la base para crear una plantilla de la pantalla que ilustra el diseño gráfico y la colocación de los iconos, la definición de textos descriptivos, la especificación y títulos de las ventanas, y la especificación de aspectos mayores y menores del menú. Con el empleo de herramientas, se hace el prototipo, se implementa en definitiva el modelo del diseño y se evalúa la calidad del resultado.

En este capítulo se muestran los pasos que se siguieron para realizar el diseño de la interfaz de usuario y la herramienta en la que se elaboró el prototipo.

4.2.- Las reglas doradas en el diseño de la interfaz de usuario

En su libro sobre el diseño de la interfaz, Theo Mandel definió tres reglas doradas:

1. Dejar el control al usuario. Este punto se refiere a un sistema que reaccione a las necesidades del usuario y que le sea de ayuda para que las cosas se hagan. Que el usuario controle la interfaz y no la interfaz al usuario.
2. Reducir la carga de memoria del usuario. Entre más cosas tenga que recordar el usuario, más fácil será que cometa errores al interactuar con el sistema. Es por esto por lo que una interfaz de usuario bien diseñada no sobrecarga la memoria del usuario. Siempre que sea posible, el sistema debe "recordar" la información pertinente y ayudar al usuario con un escenario de interacción que lo ayude a recordar.
3. Hacer que la interfaz sea consistente. La interfaz debe presentar y obtener información en forma consistente. Esto implica: 1) que toda la información se organice de acuerdo con reglas de diseño que se respeten en todas las pantallas desplegadas, 2) que los mecanismos de entrada se limiten a un conjunto pequeño usado en forma consistente en toda la aplicación, y 3) que los mecanismos para pasar de una tarea a otra se definan e implementen de modo consistente.

Estas reglas constituyen la base de un conjunto de principios de diseño de la interfaz de usuario que guían este aspecto tan importante del diseño del software.

4.2.2.- Elementos que conforman la interfaz de usuario

El primer paso fue elaborar un diagrama de flujo de la interfaz y de los elementos que la conforman, para posteriormente desarrollar el boceto prototipo en una herramienta de diseño.

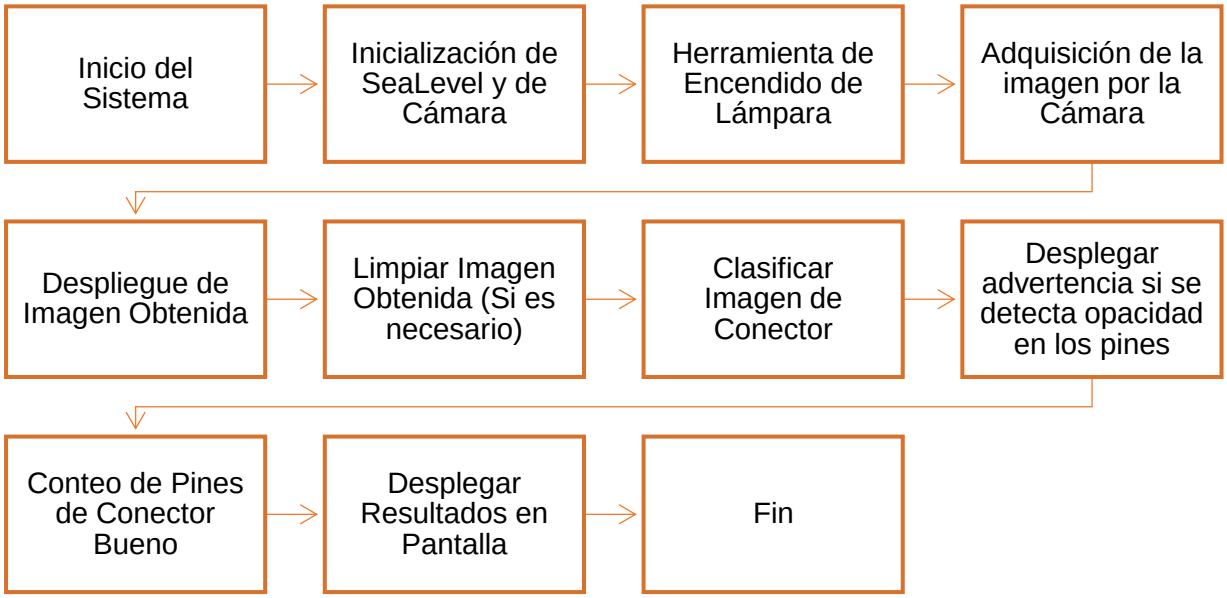

Figura 4.1 Diagrama de Flujo de la Interfaz de Usuario

4.3.- Desarrollo en la plataforma Balsamiq Wireframes

A continuación, se muestra la interfaz de usuario y sus elementos en boceto desarrollado en la aplicación Balsamiq. Balsamiq Wireframes es una herramienta rápida de creación de tramas de interfaz de usuario de baja fidelidad que reproduce la experiencia de dibujar en un bloc de notas o pizarra, pero usando una computadora.

En la Figura 4.2 podemos ver el boceto de lo que será la interfaz de usuario con la mayor parte de sus elementos. Los elementos se encontrarán en idioma inglés debido al entorno en el que trabaja el usuario.

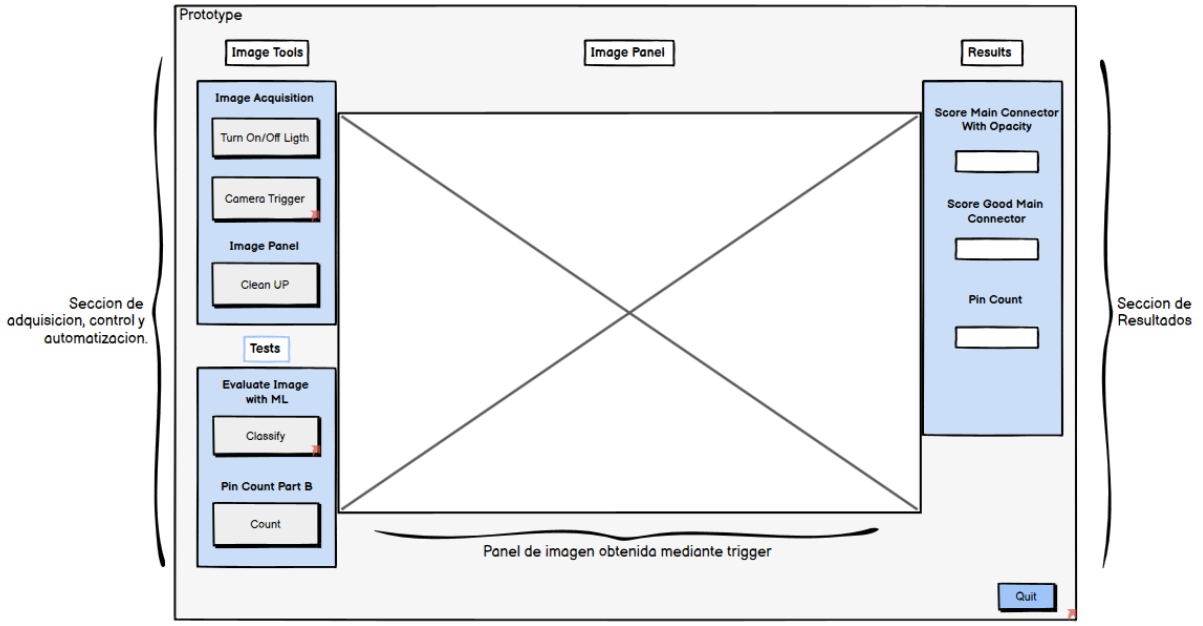

Figura 4.2 Boceto de la Interfaz de Usuario en Balsamiq

4.3.1.- Vista de elementos

En base a los elementos presentados en el CAPÍTULO III. ANÁLISIS, donde se menciona los requerimientos que debe tener la interfaz de usuario se mostrará cada uno de los elementos por separado y su vista en la pantalla en boceto. Posteriormente se presentará la interfaz en el entorno de LabWindows/CVI.

- Image Tools (Herramientas de Image):

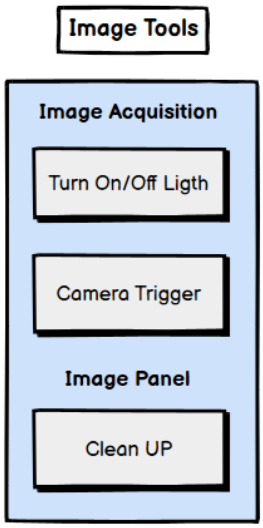

Figura 4.3 Image Tools

- Botón Turn On/Off Light (Encendido/Apagado de Lampara): Se utilizó la imagen presentada en la Figura 4.4 y está ubicado en la parte superior izquierda debajo del texto “Image Adquisición”.

Figura 4.4 Botón Turn On/Off Light

- Botón Camera Trigger (Disparo de Cámara): Se utilizó un botón del mismo color que el de Turn On/Off Ligth y se encuentra debajo del boton Turn On/Off Ligth como se muestra en la Figura 4.5.

Figura 4.5 Botón Camera Trigger

- Botón Clean UP (Limpiar): Se muestra en la siguiente Figura 4.6 y se ubica debajo de la etiqueta Image Panel en la parte inferior de la sección Image Tools.

Figura 4.6 Botón Clean Up

- Sección Tests (Pruebas): La sección de pruebas (Figura 4.7) está ubicada debajo de la sección de herramientas de imagen, cuenta con dos botones, botón de Classify y botón de Count.

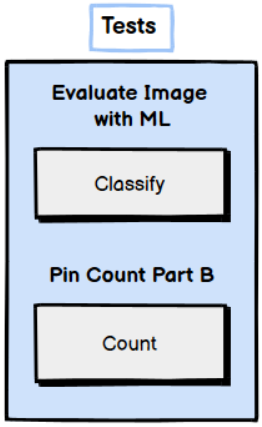

Figura 4.7 Sección Tests

- Botón Classify (Clasificar): Se ubica en la parte inferior izquierda de la interfaz y debajo de la etiqueta “Evaluate Image with ML” como se puede notar en la Figura 4.7.

- Botón Count (Contar): Se encuentra debajo del botón Classify y de la etiqueta “Pin Count Part B”, el cual podemos ver en la Figura 4.7.

- Image Panel (Panel de Imagen): Esta en el centro de la interfaz, entre las secciones de herramientas de imagen y los resultados, la cual mostrará la imagen obtenida por la cámara. Estará representado por un recuadro como el de la Figura 4.8.

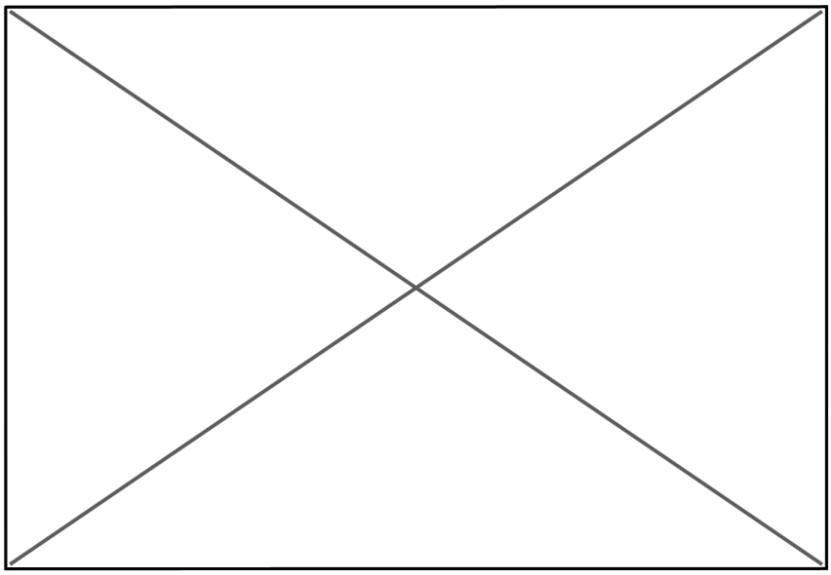

Figura 4.8 Image Panel

- Sección Results (Resultados): Esta sección se localiza del lado derecho de la interfaz, la cual cuenta con 3 cajas de texto, para los 3 resultados necesarios, que son:
    - Score *main connector* with opacity,
    - Score Good *main connector*
    - Pin Count

Estos elementos los podemos encontrar en la Figura 4.9.

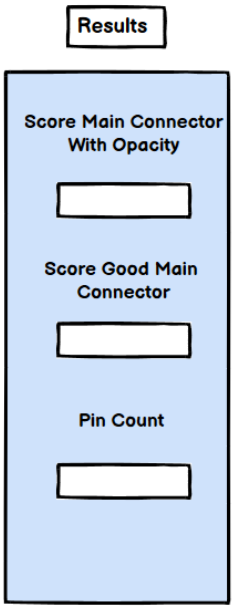

Figura 4.9 Sección Results

- Mensaje de Inicialización de Cámara: Este mensaje aparece al momento de ejecutar la interfaz de usuario, en la Figura 4.10 se observa el mensaje.

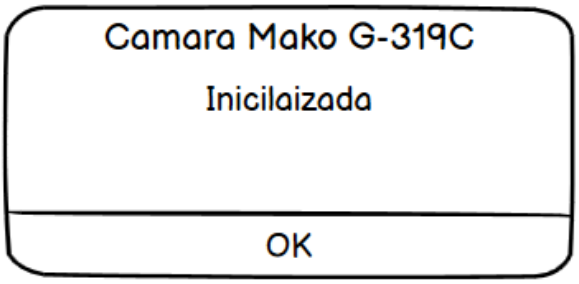

Figura 4.10 Mensaje de Cámara Inicializada

- Botón de Quit (Abandonar): Se creó un botón para abandonar la interfaz, el cual se encuentra en la parte inferior derecha. En la Figura 4.11 se muestra el botón Quit.

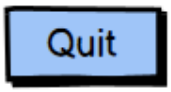

Figura 4.11 Botón de Quit

- Mensaje de Imagen Obtenida: Este mensaje aparece en el momento de ejecutar el botón de Trigger para la adquisición de la imagen, en la Figura 4.12 se observa el mensaje.

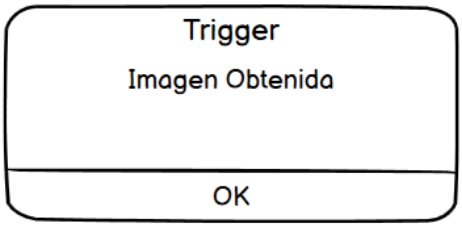

Figura 4.12 Mensaje de Imagen Obtenida

- Mensaje de Fallo en la Inicialización de SeaLevel 420U: Este mensaje aparece en la ejecución de la interfaz de usuario, como indicador si se tiene problemas al tomar control de la SeaLevel 420U. La Figura 4.13 nos muestra el mensaje.

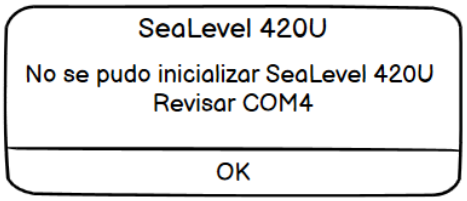

Figura 4.13 Mensaje de Fallo en la inicialización de la SeaLevel 420U

A continuación, en la Figura 4.14 se muestra la ventana de advertencia, que aparecerá en dado caso se detecte un conector con opacidad en los pines.

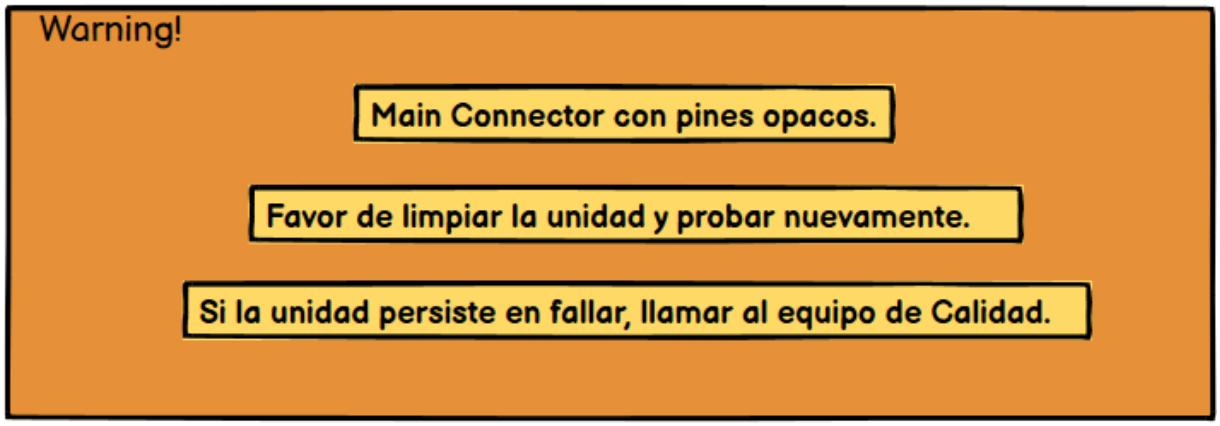

Figura 4.14 Pantalla con mensaje de advertencia

CAPÍTULO V. DESARROLLO

En este capítulo se muestra cada uno de los pasos realizados en la etapa de desarrollo del proyecto. Para esto se ha dividido en 4 secciones, la primera para el desarrollo de la clasificación de imágenes por medio de ML.Net en Visual Studio, en la segunda sección se mostrará el set up del hardware para la adquisición de la imagen, en la tercera el desarrollo del script o secuencia para el conteo de pines de la parte B del conector principal, y finalmente el proyecto en LabWindows/CVI donde estará la interfaz de usuario para utilizar las secciones anteriores. Para cada una de las secciones es necesario tener instalados los softwares mencionados en el CAPÍTULO II. MARCO TEÓRICO. Los elementos de las secciones presentadas a continuación se encontrarán en inglés debido a que las aplicaciones utilizadas cuentan con este idioma por defecto.

5.1- Iniciar la aplicación para el proyecto de Visual Studio con ML.Net

Antes de iniciar la aplicación en Visual Studio, se recomienda agregar la opción de ML.NET Model Builder (Figura 5.1) en la configuración de .Net desktop development al momento de instalarlo, ya que, si no se tiene esa configuración, será necesario agregarla posteriormente, debido a que con ML.NET Model Builder se modelará la clasificación de imágenes por medio de machine learning.

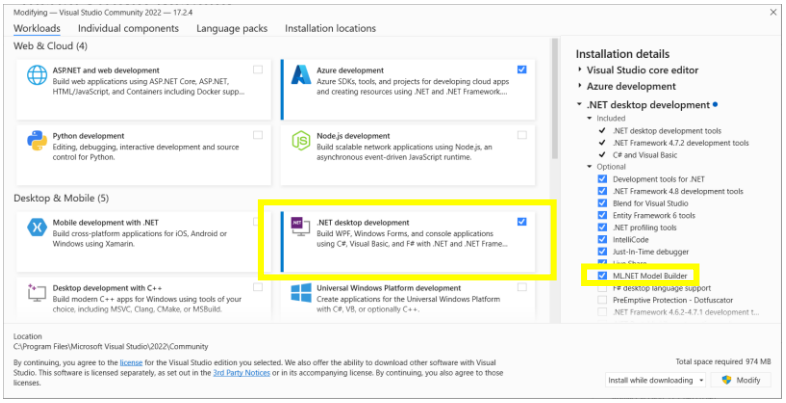

Figura 5.1 ML.NET Model Builder en Visual Studio

Al iniciar la aplicación procedemos a seleccionar ***Crear un nuevo proyecto*** (Figura 5.2).

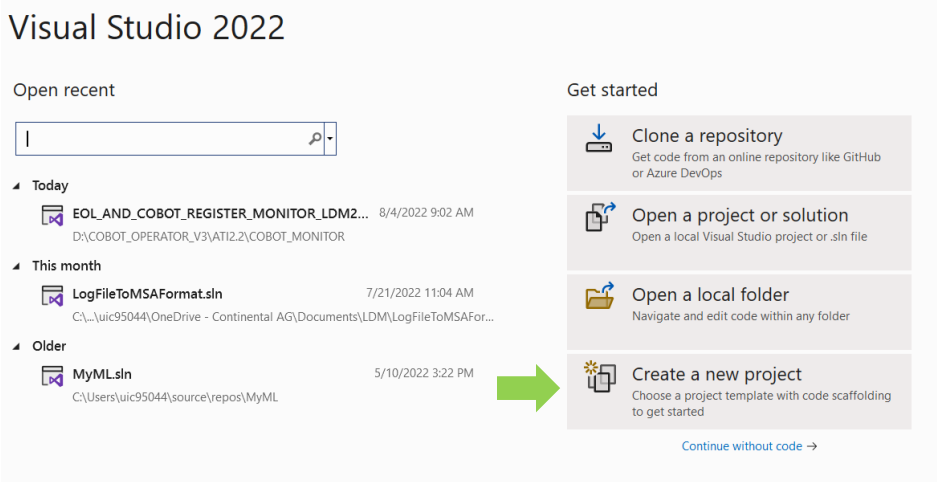

Figura 5.2 Inicio de la aplicación Visual Studio

Habiendo seleccionado la opción de la Figura 5.1, aparecerán varios templates de proyectos que se pueden utilizar, en este caso se estará utilizando la opción de **C# Console App**, y se presiona siguiente (Next), como se muestra a continuación.

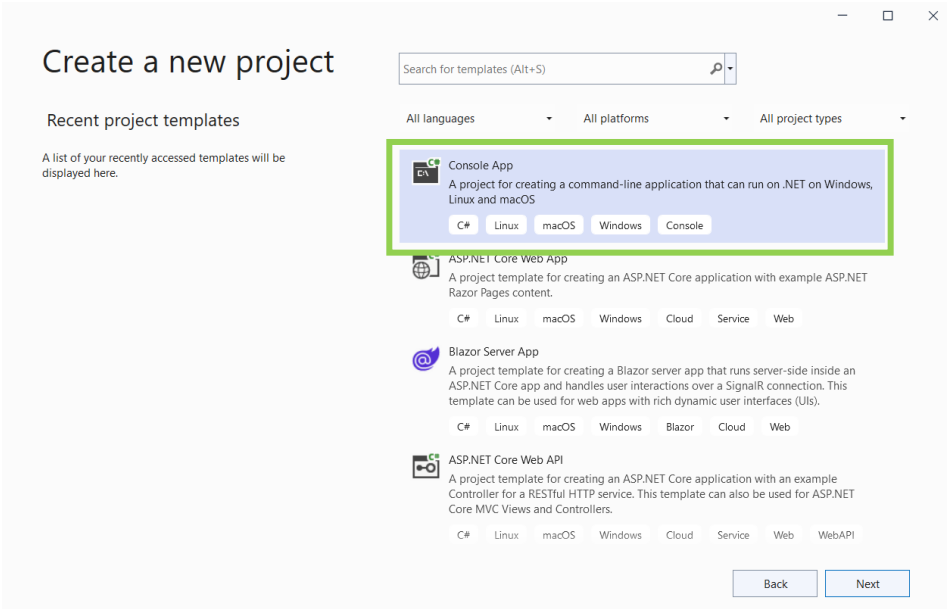

Figura 5.3 Seleccionamos C# Console App

En la siguiente ventana de configuración del proyecto (Figura 5.4) escribimos el nombre del nuestro proyecto en este caso "**myMLApp**", tenemos la opción de cambiar la ubicación del proyecto, en este caso se deja la que viene por defecto, de igual manera es recomendable seleccionar que el nombre del proyecto sea igual a la de la solución (modelado en ML.NET) y ubicarlos en la misma dirección, para ello se selecciona "Place solution and project in the same directory", por último damos clic en Siguiente. Como información adicional seleccionamos **.NET 6.0** (Long-term support) como Framework (Figura 5.5).

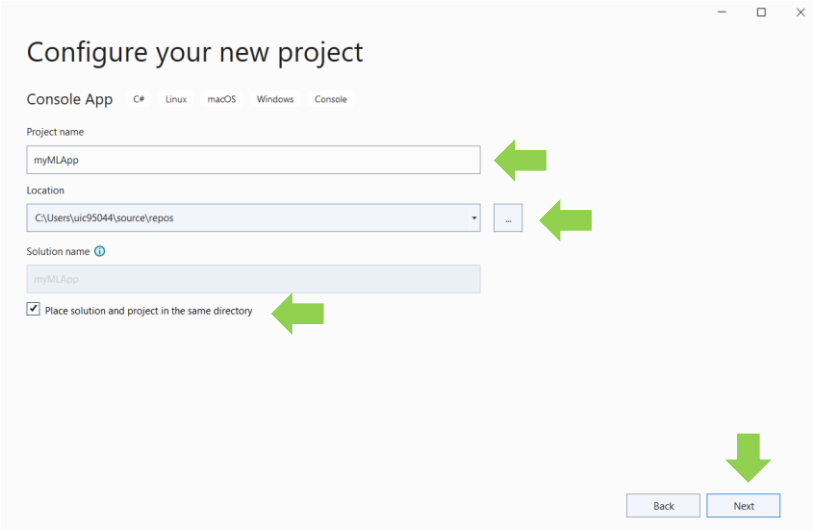

Figura 5.4 Configuración de nuevo proyecto

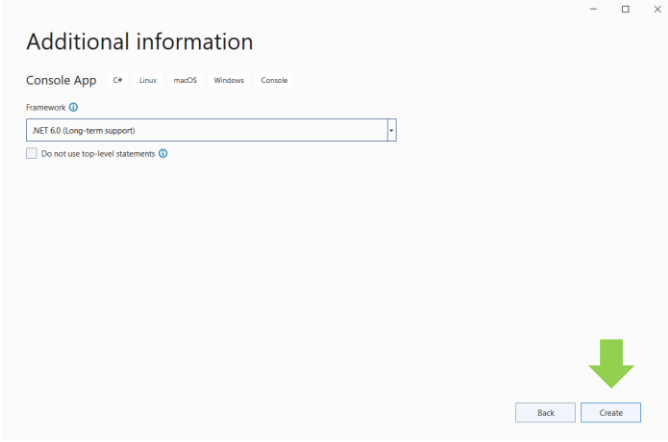

Figura 5.5 Información adicional de .NET

Al dar clic en **Create** se comenzará a crear nuestro proyecto y al finalizar la carga del proyecto se observará de la siguiente manera (Figura 5.6).

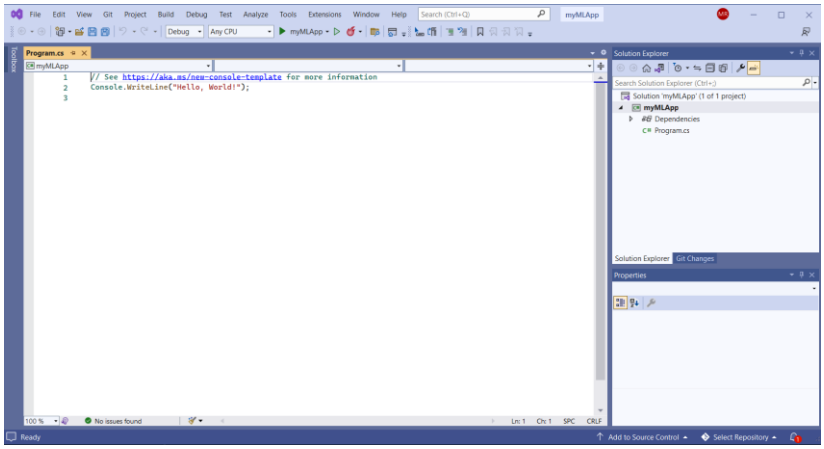

Figura 5.6 Proyecto iniciado

Se creo una pestaña por defecto en el proyecto, **Program.cs**, en la cual se agregará la solución del código para el modelo de clasificación de imágenes.

5.1.1.- Agregar machine learning

Hacemos clic con el botón derecho del mouse en el proyecto myMLApp en el Solution Explorer, situado en la parte superior derecha de la ventana principal, seleccionamos Add, y Machine Learning Model (Figura 5.7). Por subsiguiente aparecerá una ventana como la que se observa en la Figura 5.8, que se llama Add New Item, en la que debemos cerciorarnos de que esté seleccionado el Modelo de aprendizaje automático (ML.NET). En la ventana de Add New Item tenemos opción de cambiar el nombre del modelo de Machine Learning (ML.Net), para esto se puso un nombre con el que podamos identificar la solución para el proyecto, el nombre se encontrara como “ML_Evaluate_Main_Connector_Model.mbconfig”, con la extensión mbconfig.

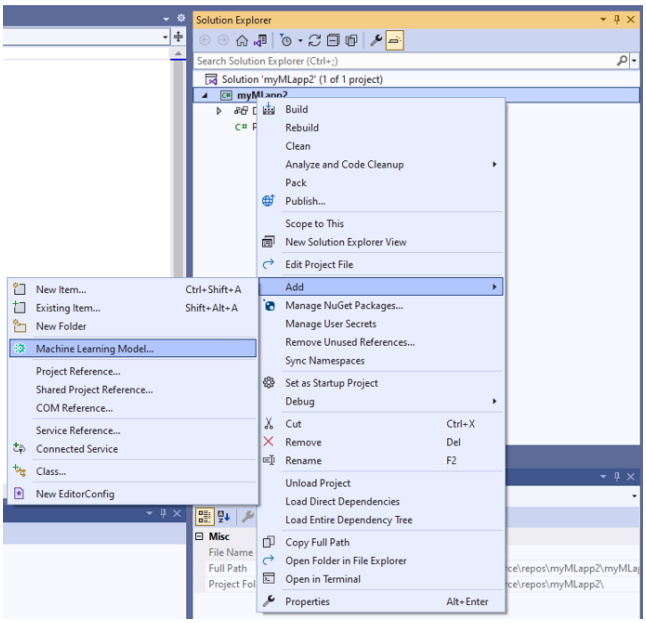

Figura 5.7 Agregar machine learning

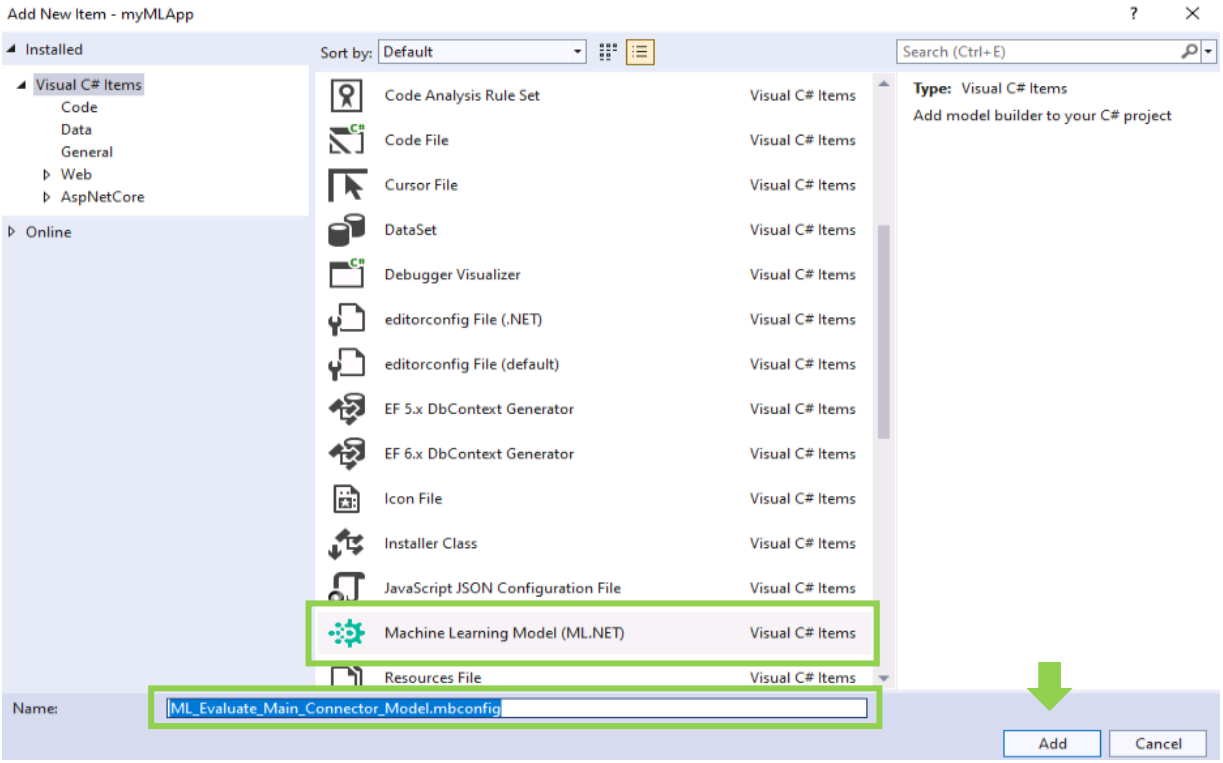

Figura 5.8 Agregar nuevo elemento de ML.NET

Al seleccionar Add (Figura 5.8) se agrega un nuevo archivo llamado ML_Evaluate_Main_Connector_Model.mbconfig a la solución y la interfaz de usuario de Model Builder (Figura 5.9) se abre en una nueva ventana de herramientas acoplada en Visual Studio. El archivo mbconfig es simplemente un archivo JSON que realiza un seguimiento del estado de la interfaz de usuario. Model Builder nos guiará a través del proceso de creación de un modelo de aprendizaje automático.

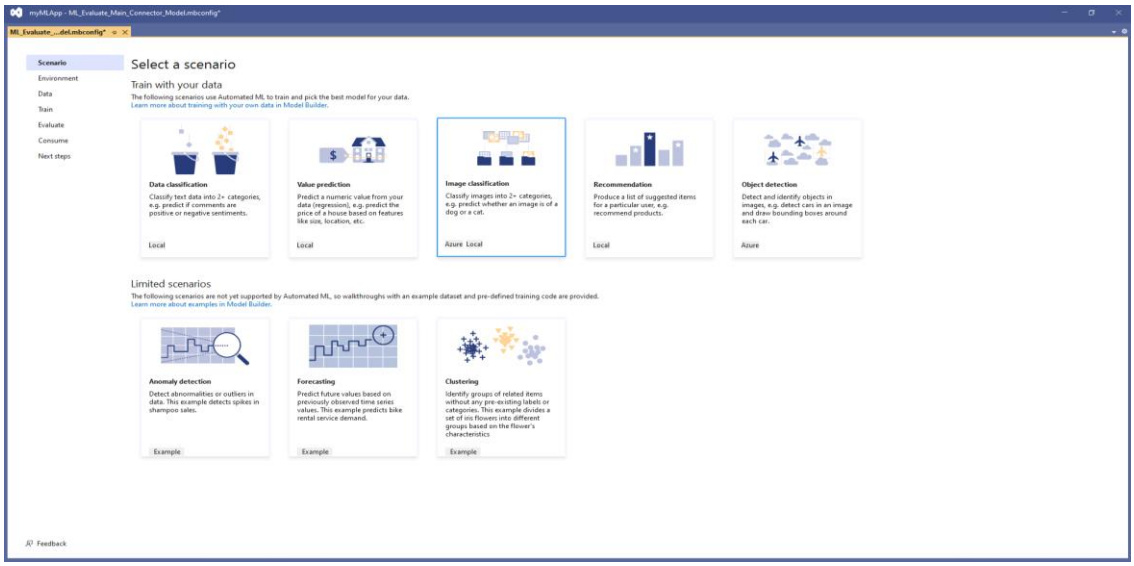

Figura 5.9 Interfaz de usuario de Model Builder

5.1.2.- Seleccionar un escenario en Model Builder

En la Figura 5.9 se muestra los tipos de escenarios que podemos seleccionar de acuerdo con la solución que estemos buscando, estas soluciones son las mencionadas en el CAPÍTULO II, página 37 donde se habla del tipo de predicciones que podemos realizar con ML.Net. Por consiguiente, puesto que buscamos una solución relacionada a la clasificación de imágenes de un conector principal, seleccionaremos el escenario de clasificación de imágenes (Image Classification), en el que solo estaremos utilizando dos categorías (Good_Main_Connector y Bad_Main_Connector).

La Figura 5.10 indica el icono que aparecerá en la interfaz de model builder, y que estaremos seleccionando para la solución.

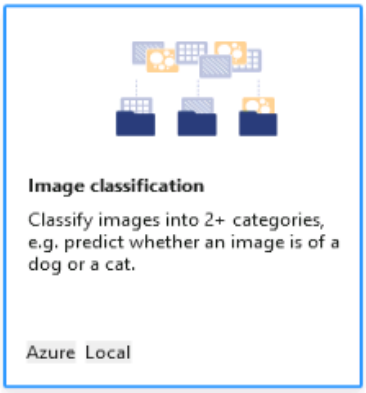

Figura 5.10 Escenario de Image Classification

Después de seleccionar el escenario de clasificación de imágenes, debe elegir su ambiente de entrenamiento. Algunos escenarios admiten el entrenamiento del modelo en Azure, entorno Local de la PC, o en la nube. Para seleccionar esta opción se nos presentará una pantalla como la que podemos ver en la Figura 5.11.

Debido a que las imágenes que se estarán utilizando para el entrenamiento del modelo de clasificación de imágenes están en una PC local, seleccionaremos Local (CPU), y continuaremos al siguiente paso dando Next step.

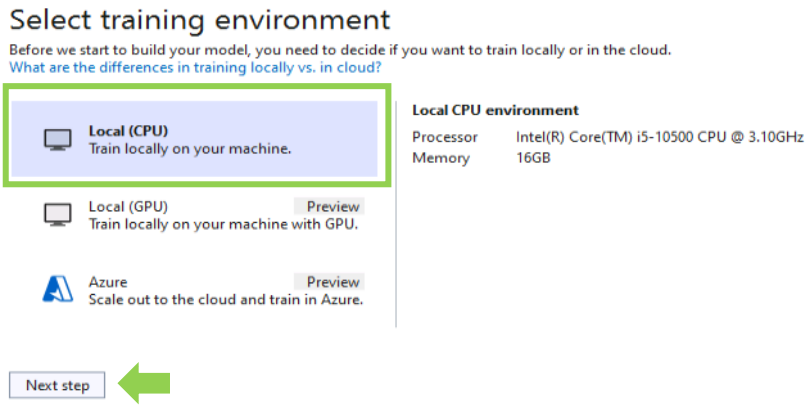

Figura 5.11 Ventana de ambiente de entrenamiento

5.1.3.- Agregar Datos

Como siguiente paso debemos seleccionar los datos que se estarán utilizando en el modelo de clasificación de imágenes. En Model Builder, puede agregar datos desde un archivo local o conectarse a una base de datos de SQL Server en dado caso tengamos datos en texto. Para nuestro caso seleccionaremos la dirección que contiene todas las imágenes que seleccionamos previamente para el entrenamiento. El modelo de clasificación de imágenes soporta formatos de png, jpg, jpeg, y gif. Las imágenes a utilizar están en formato pgn., y se localizan en la siguiente dirección "C:\Users\tbench\Documents\Images\Conectors_images_data_set". La Figura 5.12 nos muestra la ventana en la que agregan las imágenes.

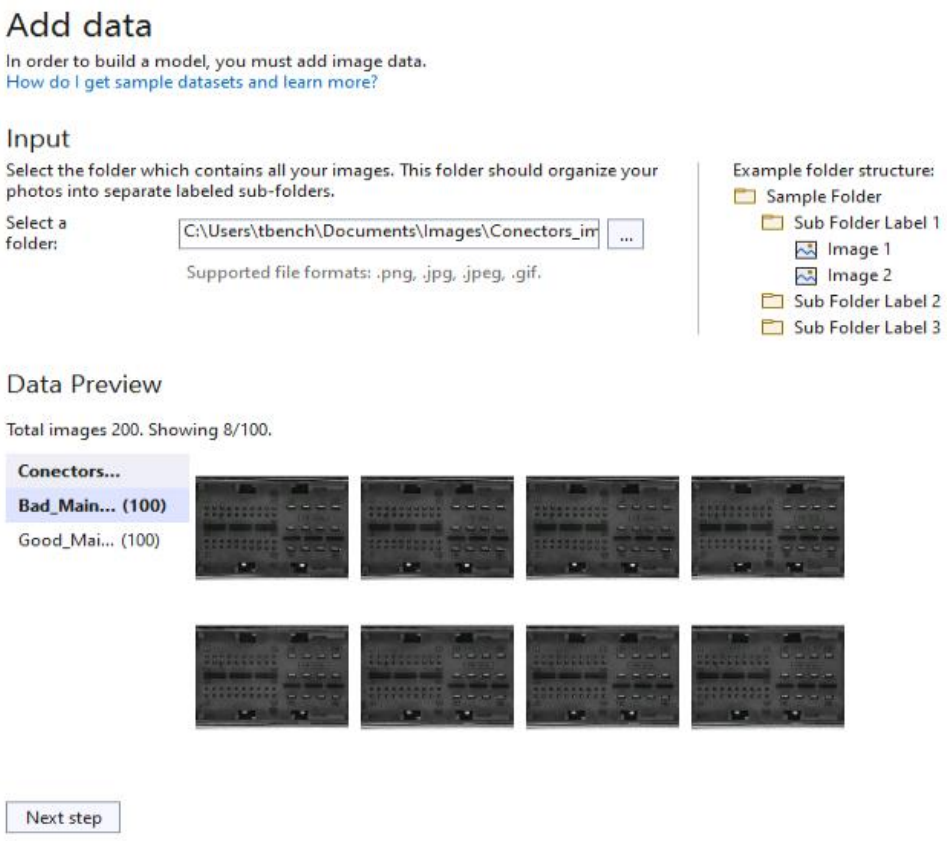

Figura 5.12 Agregar imágenes a Model Builder

Cabe mencionar que al momento de seleccionar la dirección donde se encuentran localizadas las imágenes, es necesario considerar la estructura presentada en el ejemplo (Figura 5.13), ya que esto evitará tener problemas al momento de cargar las imágenes al modelo de clasificación.

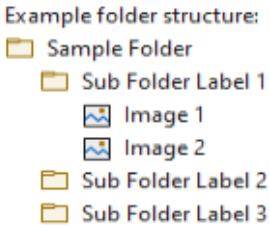

Figura 5.13 Ejemplo de estructura de folder para localizar imágenes.

Para este punto se seleccionaron 100 imágenes de conectores sin opacidad en los pines, y 100 de conectores con opacidad en los pines. Se obtuvieron las imágenes de radios que fueron probados anteriormente en los meses de febrero y marzo del 2022. Las carpetas donde se localizaron las imágenes fueron renombradas como “Bad_Main_Connector” (Figura 5.14) y “Good_Main_Connector” (Figura 5.15).

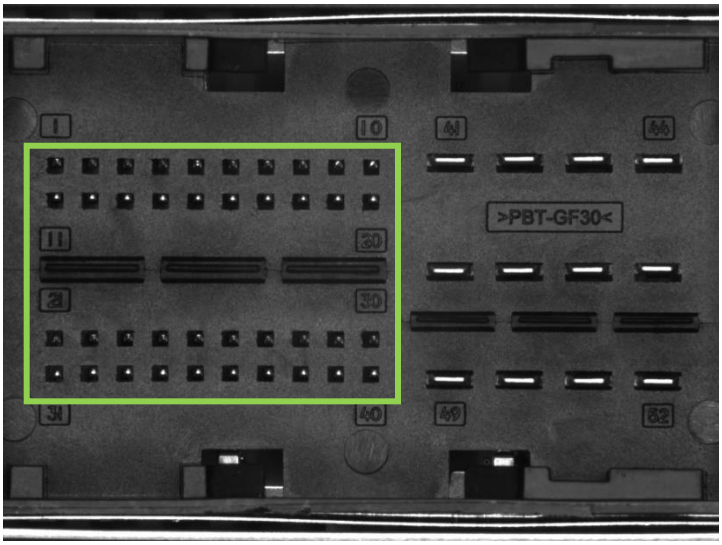

Figura 5.14 Bad_Main_Connector

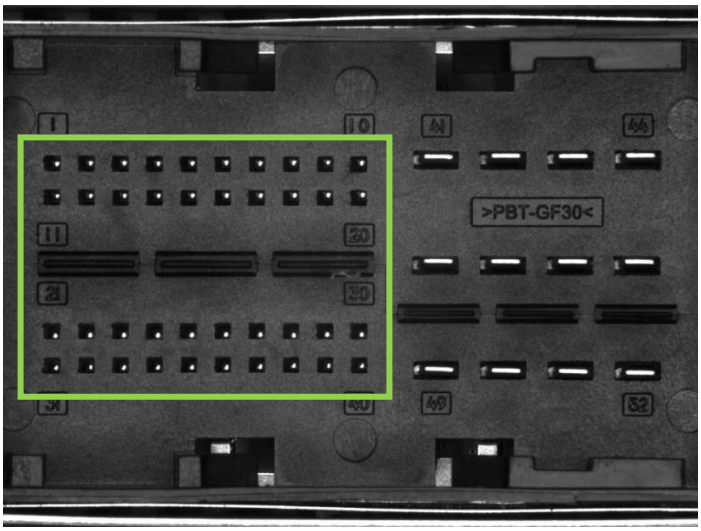

Figura 5.15 Good_Main_Connector

5.1.4.- Entrenar modelo

En este paso se entrena el modelo con el set de imágenes que hemos seleccionado anteriormente. Model Builder evalúa muchos modelos con diferentes algoritmos y configuraciones en función de la cantidad de tiempo de entrenamiento dado para construir el modelo de mejor rendimiento. En la Figura 5.16 podemos ver la ventana de entrenamiento, cuando ya se ha seleccionado “Train”. Si se ha realizado un entrenamiento previo, aparece “Train again”, lo que nos permite volver a entrenar nuestro modelo, en caso de que sea requerido (Figura 5.17).

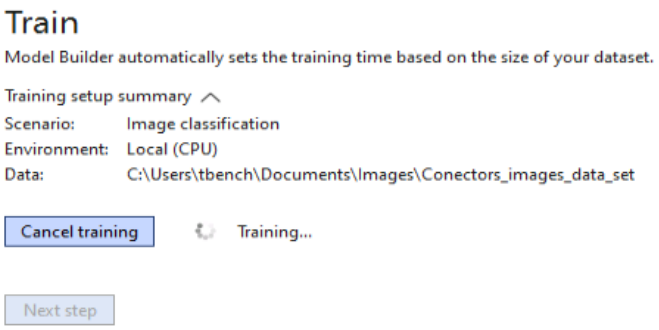

Figura 5.16 Model Train

Justo después de terminar el entrenamiento del modelo seleccionado, aparecerá una ventana con información y resultados importantes del proyecto. Hace mención del tipo de escenario seleccionado, el data set que se está utilizando y la localización de la data set, así como el mejor modelo encontrado, en este caso una DNN + ResNet50.

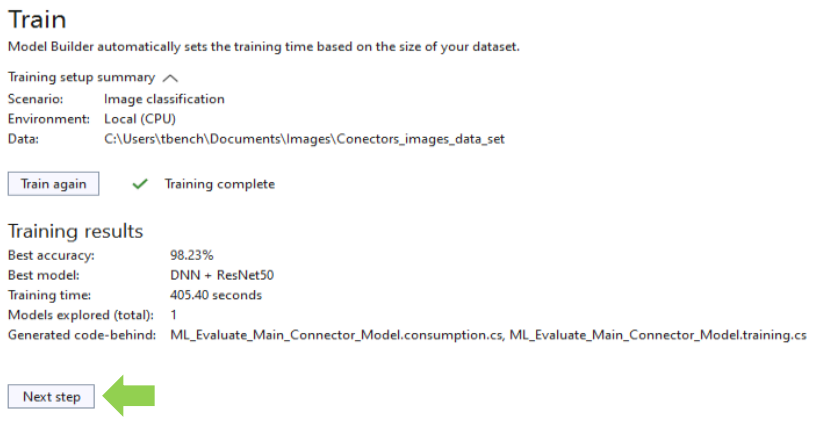

Figura 5.17 Entrenamiento de modelo completado

5.1.5.- Evaluar el modelo

En este paso se nos ofrece la oportunidad de evaluar de manera practica el modelo que se está desarrollando antes de implementarlo en código y generar las librerías adecuadas para su funcionamiento. Evaluar el modelo nos muestra el algoritmo de mejor rendimiento y la mejor precisión, permitiendo probar el modelo en la interfaz de usuario.

En la Figura 5.18 podemos ver la ventana que permitirá evaluar el modelo. En tales circunstancias debemos seleccionar una imagen, para que así el modelo pueda realizar la predicción en la clasificación de la imagen. Una vez seleccionada la imagen de prueba, arroja un resultado en porcentaje de asertividad en la clasificación, la Figura 5.19 nos muestra una evaluación previa con una imagen de un conector que tiene suficiente brillo en los pines.

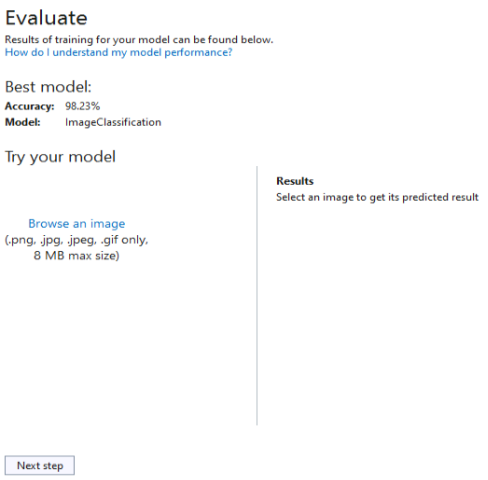

Figura 5.18 Evaluar modelo

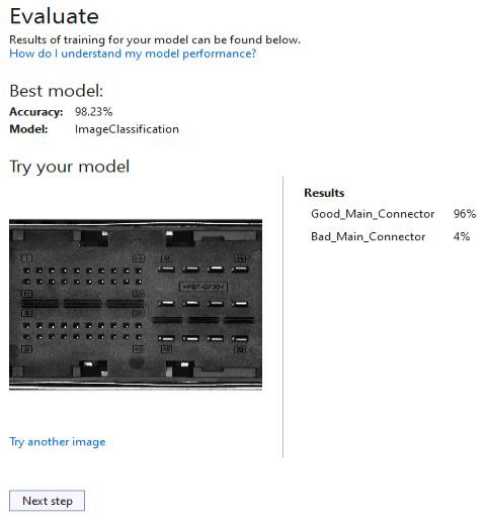

Figura 5.19 Modelo evaluado con una imagen

Si el modelo no está funcionando adecuadamente en este paso, es necesario considerar aumentar el número de imágenes en el data set, pero que conllevaría a un aumento en el tiempo de entrenamiento de la DNN + ResNet50. El tiempo aproximado que llevo entrenar el modelo con 200 imágenes (100 imágenes buenas y 100 imágenes malas) fue de 377.8 segundos.

5.1.6.- Generar Código

Una vez completado el entrenamiento, se agregan automáticamente tres archivos como código subyacente al ML_Evaluate_Main_Connector_Model.mbconfig. Los tres archivos generados son los siguientes:

- ML_Evaluate_Main_Connector_Model.zip: Este archivo es el modelo ML.NET entrenado, que es un archivo zip serializado.
- ML_Evaluate_Main_Connector_Model.consumption.cs: Este archivo contiene las clases de entrada y salida del modelo y un método Predict que se puede usar para el consumo del modelo.
- ML_Evaluate_Main_Connector_Model.training.cs: Este archivo contiene el pipeline de entrenamiento (transformación de datos, algoritmo y parámetros de algoritmo) utilizada para entrenar el modelo final.

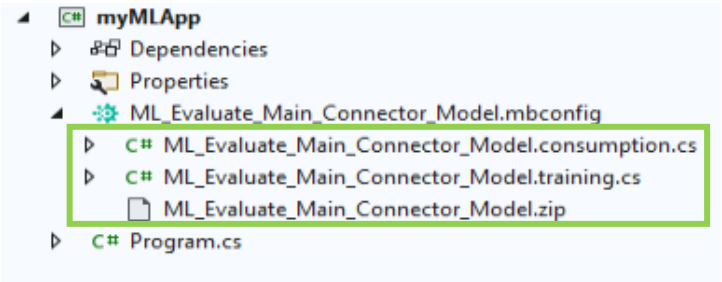

Figura 5.20 Archivos generados para la solución del modelo

La Figura 5.20 hace referencia a los tres archivos mencionados al momento de generar el código para solución del modelo y como se verían en el Solution Explorer en Visual Studio.

5.1.7.- Consumir el modelo

En el paso Consumir en Model Builder, se proporciona un fragmento de código que crea una entrada de muestra para el modelo y usa el modelo para hacer una predicción sobre esa entrada. También Model Builder nos da la opción de utilizar

plantillas de proyecto que podemos agregar opcionalmente la solución. Hay dos plantillas de proyecto (una aplicación de consola y una API web), ambas ejecutan el modelo entrenado. La Figura 5.21 muestra tanto el fragmento de código generado, como los Project templates que podemos agregar a la solución del modelo.

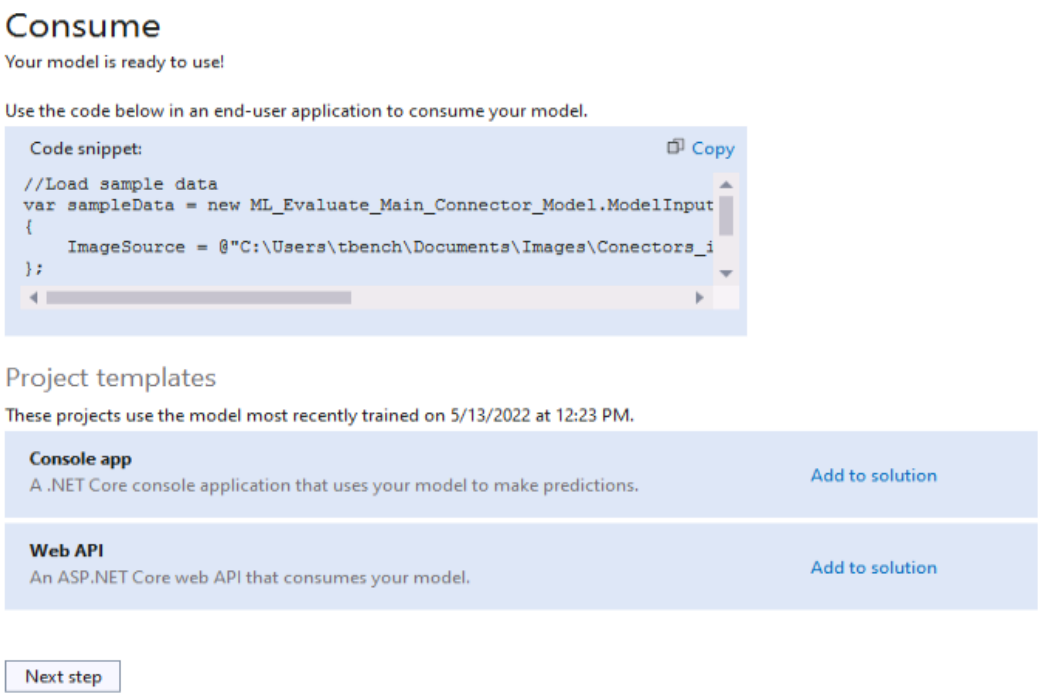

Figura 5.21 Imagen agregada

5.1.7.1- Reemplazar Código

Teniendo el código generado para el modelo de la solución de clasificación de imágenes, debemos reemplazarlo en el Program.cs, como se muestra a continuación (Figura 5.22).

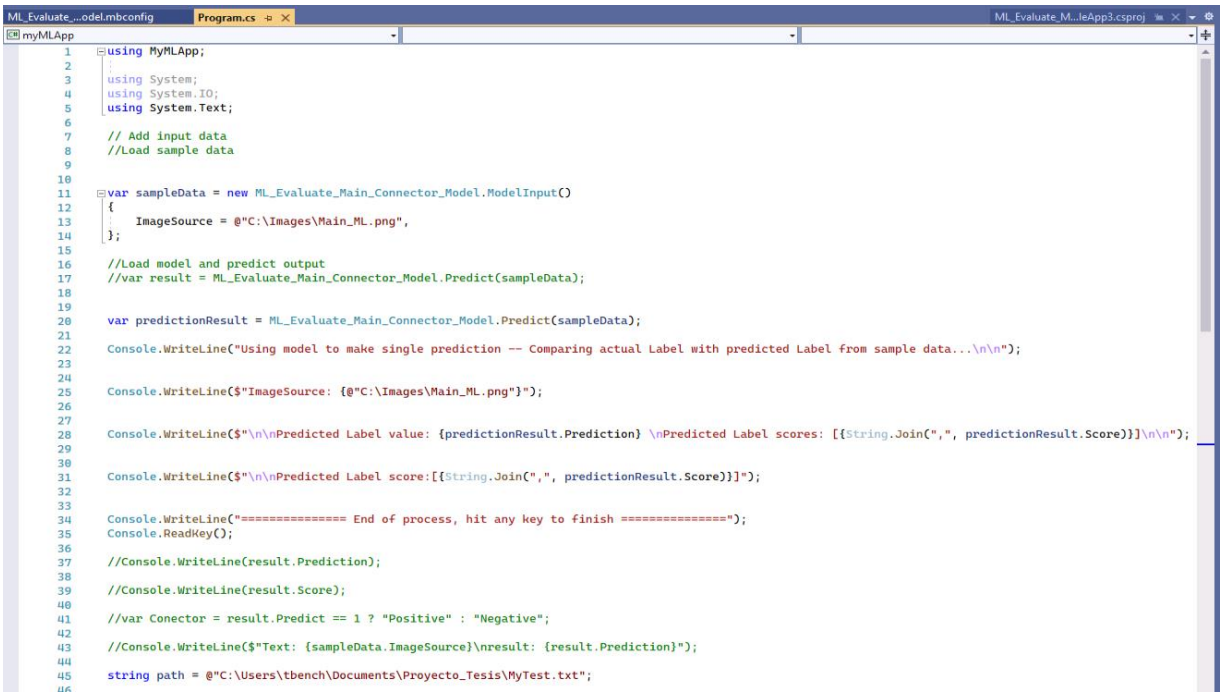

Figura 5.22 Reemplazo de código

En la sección de código de la Figura 5.22 se manda llamar la clase `ML_Evaluate_Main_Connector_Model`, en la que se encuentra el modelo de clasificación previamente entrenado. En esta sección se define la dirección de la cual se estará tomando la imagen para ser evaluada, la dirección que se definió es la siguiente: `"C:\Images\Main_ML.png"`.

Al final de la sección del código presentado en la Figura 5.22 se crea un archivo de texto, en dicho archivo se estarán almacenando los resultados de score de clasificación entregados por el modelo de solución. Para el proyecto, el archivo de texto se localizó en la dirección "C:\Users\tbench\Documents\Proyecto_Tesis\MyTest.txt", con el nombre “MyTest.txt”.

5.1.7.2- Ejecutar el modelo como Debug Console

Podemos ejecutar myMLApp (o seleccione Ctrl+F5 o Depurar > Iniciar sin depurar), lo cual debería arrojarnos el resultado de la Figura 5.23, prediciendo si la clasificación de la imagen de entrada es un Bad main connector o un Good main connector.

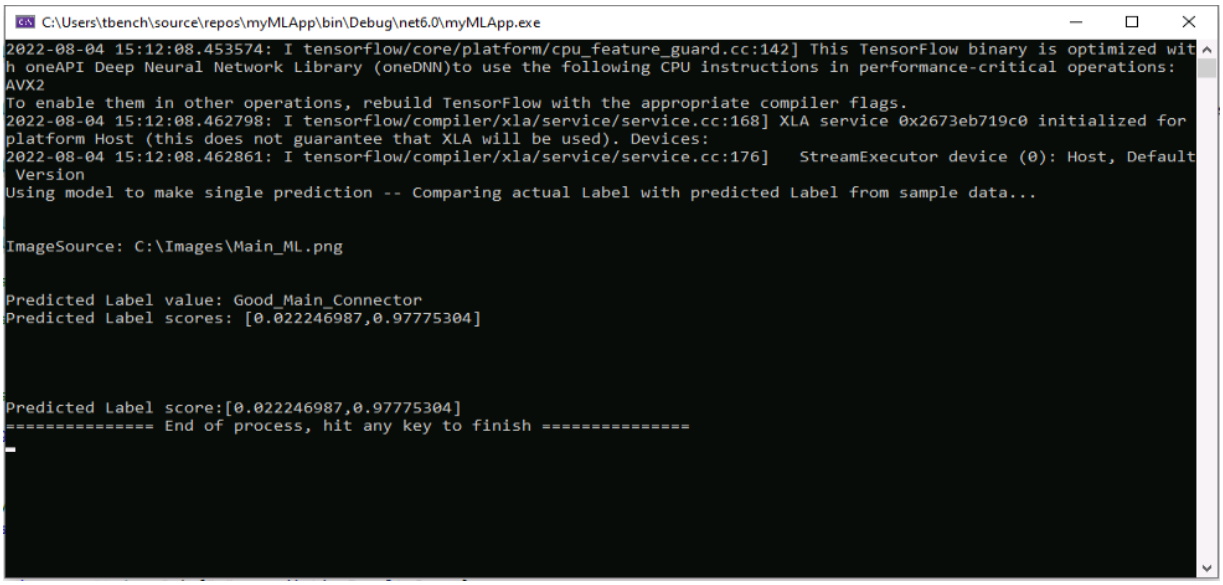

Figura 5.23 Modelo ejecutado en Consola

Para el caso presentado en la Figura 5.23 se muestra la evaluación con una imagen buena, lo cual aparece con el indicador “Good_Main_Connector”, debajo de eso aparece el score de clasificación, presentado como: [0.022246987,0.97775304]. El score del lado izquierdo indica el puntaje que se tiene de un mal conector, y el punto del lado derecho presenta el puntaje de un buen conector.

El siguiente paso en el desarrollo tiene que ver con el set up del hardware utilizado para la adquisición de la imagen, así como configuraciones que debemos tomar en cuenta para la obtención de una imagen nítida y sin distorsión.

5.2.- Set Up de Hardware

La Figura 5.24 muestra el hardware utilizado para la adquisición de la imagen. Debemos recordar que de acuerdo con las especificaciones de la cámara y del lente que se está utilizando, debemos posicionar el objeto a inspeccionar, en este caso los conectores de un radio de FCA, a una distancia de trabajo de 181.1 mm, esto permite enfocar adecuadamente el punto de inspección, logrando a su vez una imagen nítida. Además, en la misma Figura 4.24 se muestra colocada la lampara HPR2-100SW.

Figura 5.24 Set Up de Hardware

Por consiguiente, logrando ajustar la distancia de trabajo adecuada, debemos utilizar un goniómetro o angulometro, en este caso uno digital, de tal manera que dejemos el ángulo de la cámara a 0° o lo más cercano que podamos a 0°, como se observa en la Figura 5.25., de lo contrario, se corre el riesgo de tener una imagen distorsionada, complicando la obtención de resultados óptimos.

Figura 5.25 Cámara con ángulo de 0°

Posterior a la colocación de la cámara y del lente, se hicieron las conexiones necesarias a la SeaLevel 420U y la XP Power Supply para el encendido de la lampara. Ver Figura 5.26.

Figura 5.26 Conexiones de SeaLevel 420U y XP Power Supply

Teniendo conectada la mini PC e instalado el controlador de la SeaLevel 420U, podemos ver el puerto de comunicación de la SeaLevel en Device Manager, el cual aparecerá con el nombre “SeaIO USB Expansion Interface” (Figura 5.27). El COM de comunicación que se estará utilizando es el COM4.

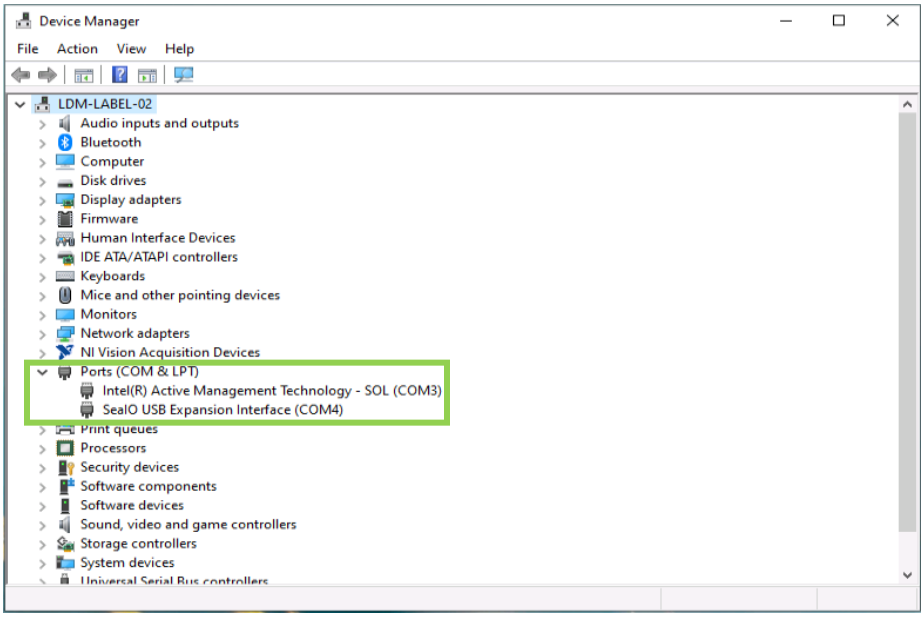

Figura 5.27 COM de comunicación para SeaLevel 420U

Teniendo la detección del COM4 de la SeaLevel podemos manipular manualmente el encendido de la lampara por medio del software SeaMax (Figura 5.28).

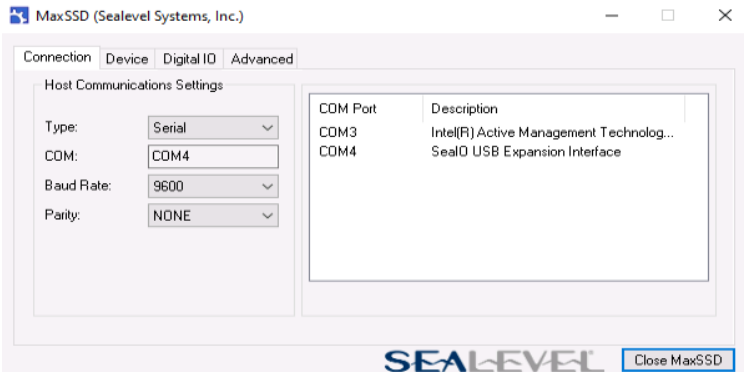

Figura 5.28 Ventana de controlador de SeaLevel 420U

Seleccionando Digital IO, en las opciones de la ventana del software de SeaMax podremos encontrar las entradas y salidas que estarán conectadas a la SeaLevel (Figura 5.29), en este caso solo tenemos conectada la lampara a uno de los 8 relevadores con los que cuenta la 420U. El relevador que se utilizó para el encendido y apagado de la lámpara fue el Relay2

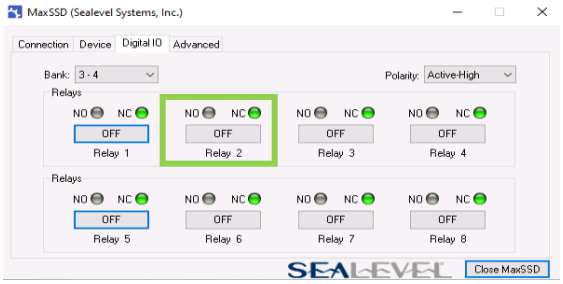

Figura 5.29 Digital IO de SeaLevel 420U

5.2.1- Configuración de la cámara Mako en NI Max

Habiendo conectado la cámara Mako G-319 por medio del cable de ethernet GigE al Ethernet Switch y este a la mini PC, debemos abrir el programa NI Max para configurar unas opciones de la cámara. Al ejecutar el programa NI Max aparecerá la interfaz de la Figura 5.30.

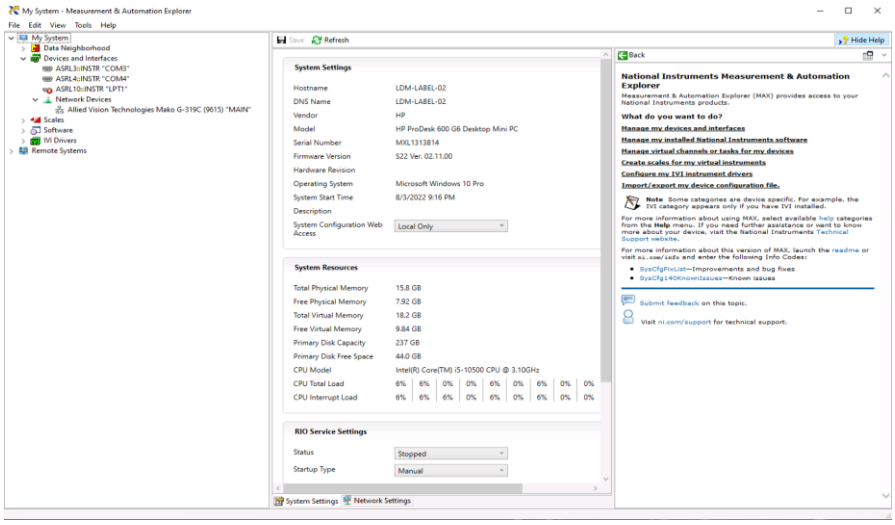

Figura 5.30 Interfaz de NI Max

En la esquina superior derecha de la interfaz de la Figura 5.30 encontramos la opción de “Devices and Interfaces”, dentro de esta opción seleccionaremos “Network Devices” en la que se encontrara la cámara Mako, nombrada como Allied Vision Technologies Mako G-319C (Figura 5.31).

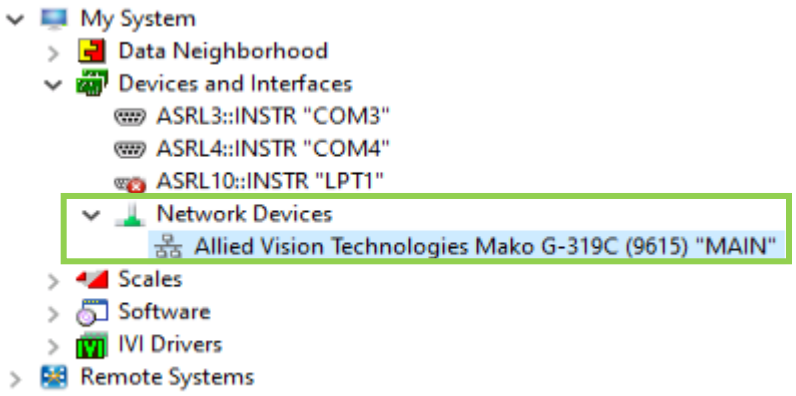

Figura 5.31 Network Devices

Al dar doble clic en Allied Vision Technologies Mako G-319C aparecerán las opciones de ajuste, atributos de adquisición, atributos de la cámara, entre otras. (Figura 5.32)

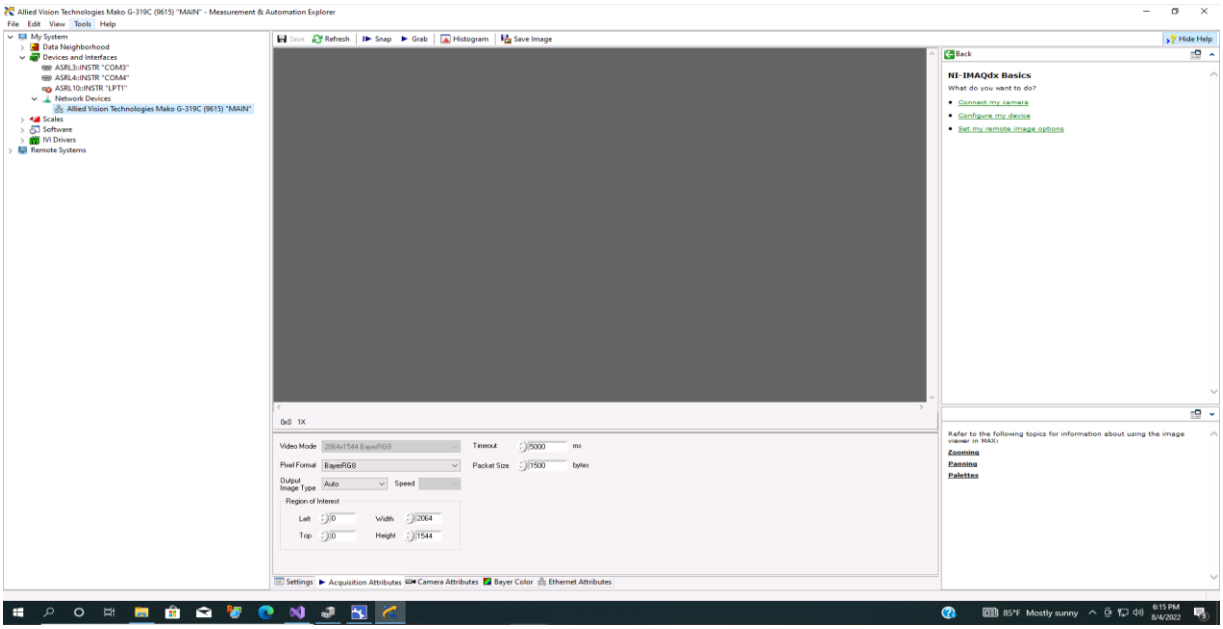

Figura 5.32 Pantalla de atributos de cámara

Los atributos que nos interesa resaltar son los atributos de adquisición (Figura 5.33), ya que se recomienda poner el Pixel Format en BayernRGB, el Packet Size en 1500 bytes, la región de interés en 2064 para Width, y 1544 para Height. Esto por el hecho que al momento de conectar la cámara no se lograba adquirir la imagen, con la configuración presentada en la Figura 5.33 se logró la adquisición de la imagen. En la imagen de la Figura 5.34 se observa la adquisición de la imagen al momento de seleccionar Grab. Cabe mencionar que el Field of View para el lente seleccionado es de 51.67 x 38.62 mm, ya que el sensor de la cámara Mako es de 1/1.8".

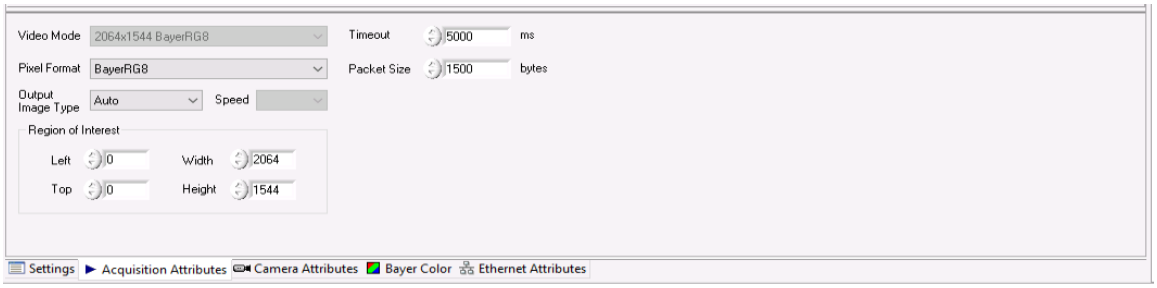

Figura 5.33 Atributos de adquisición

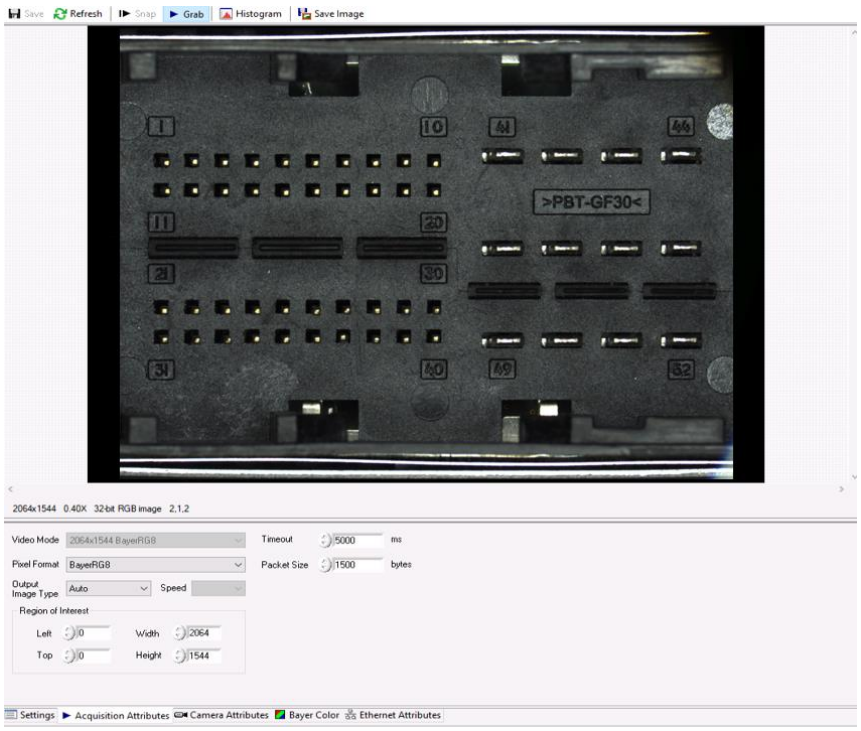

Figura 5.34 Adquisición de la Imagen en NI Max

5.3.- Script de Vision Assistant

Después de haber configurado la cámara para la adquisición de la imagen, se procede a generar el script o secuencia para la detección de los pines de la parte B del conector principal. Al ejecutar Vision Assistant 2013 SPI aparece la pantalla de inicio como se puede observar en la Figura 5.35, en esta ventana tenemos la opción de abrir una imagen para procesarla, de adquirir una imagen, o seleccionar un asistente de solución.

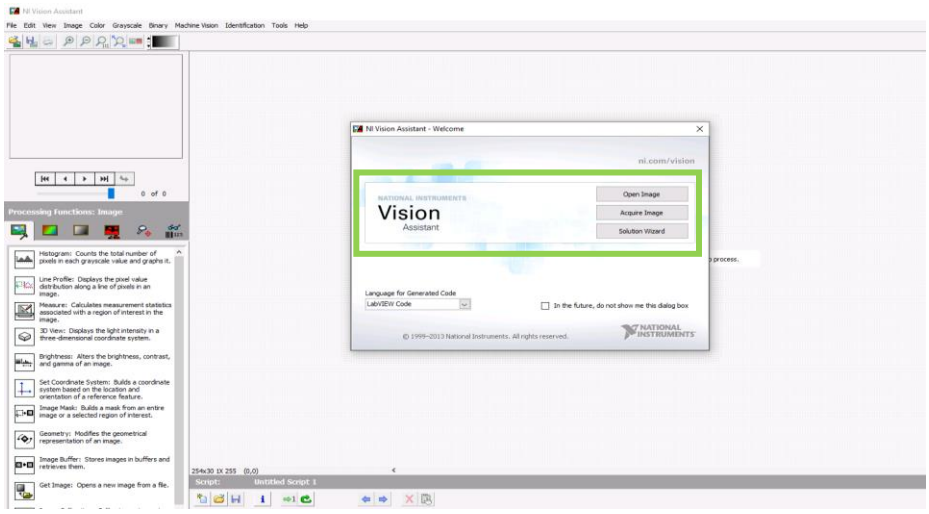

Figura 5.35 Interfaz de Vision Assistant

Para crear un nuevo script, solo hay que irnos a File, y New Script.

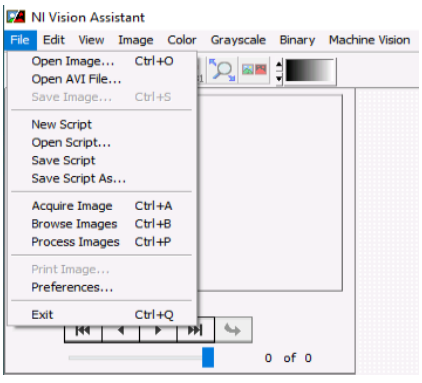

Figura 5.36 Nuevo script

5.3.1- Procesamiento de la imagen en el script

Después de cargar la imagen a procesar y de haber creado un nuevo script, se empezará con el procesamiento de la imagen. Para ello es necesario poner atención en las funciones de procesamiento que se nos presentan en la parte izquierda del panel de Vision Assitant (Figura 5.37). De todas las funciones con las que cuenta este software, solo se mencionarán las necesarias para la solución que buscamos.

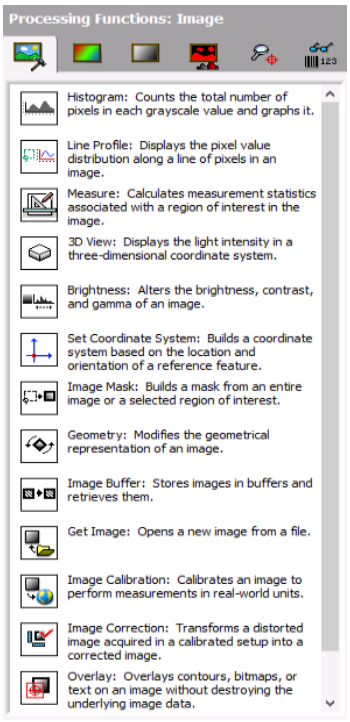

Figura 5.37 Funciones de procesamiento de imagen

En el centro de la interfaz observaremos la imagen original, debajo de ella el script con cada de una de las funciones de procesamiento que se hayan agregado. El script siempre iniciara con la imagen original, las primeras 4 funciones de procesamiento que agregan serán:

1. Color Plane Extraction. - Extrae los tres planos de color (RGB, HSV o HSL) de una imagen.

2. Pattern Matching. - Comprueba la presencia de un template en toda la imagen o en una región de interés en función de su intensidad.
3. Set Coordinate system. - Crea un sistema de coordenadas basado en la ubicación y orientación de una característica de referencia.
4. Image Mask. - Esta función se utiliza para crear una máscara a partir de una imagen completa o una región seleccionada. Puede usar máscaras de imagen para enfocar su procesamiento o inspección en regiones particulares de la imagen.

En la Figura 5.38 podemos ver las primeras 4 funciones utilizadas en el script de procesamiento de la imagen. La Figura 5.39 muestra el resultado de aplicar el Color Plane Extraction, al cual aplicamos el panel de extracción HSV – Value Plane (del inglés Hue, Saturation, Value – Matiz, Saturación, Valor).

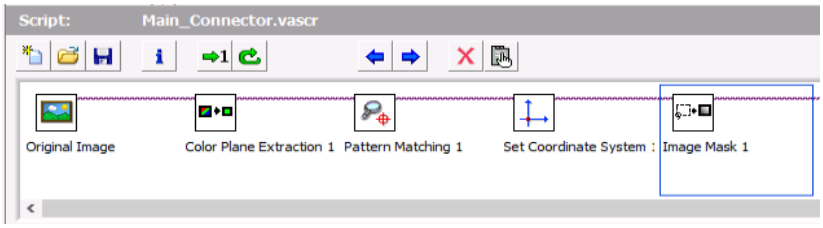

Figura 5.38 Cuatro funciones de procesamiento de imagen

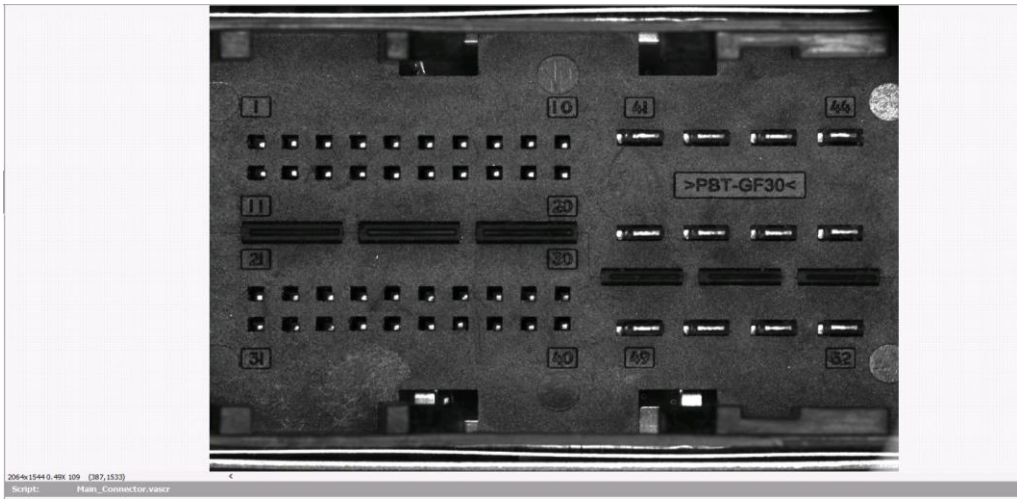

Figura 5.39 Aplicación de Color Plane Extraction

El siguiente paso es la utilización de Pattern Matching (Figura 5.40), en el recuadro verde de la imagen de la figura 5.40 podemos ver que se refiere a la región de búsqueda del patrón, el recuadro rojo seria el patrón que se ha seleccionado, en este caso dejamos como patrón de búsqueda la leyenda “>PBT-GF30<”, debido a que todos los conectores cuentan con esta leyenda, y no tendría movimiento alguno, ya que si seleccionamos por ejemplo uno de los pines como patrón, corremos el riesgo que el pin este desalineado por pasar algún proceso, y eso puede ocasionar que el resto de herramientas o funciones de búsqueda de patrón o mascaras puedan moverse de lugar si tenemos un set de coordenadas en ese patrón.

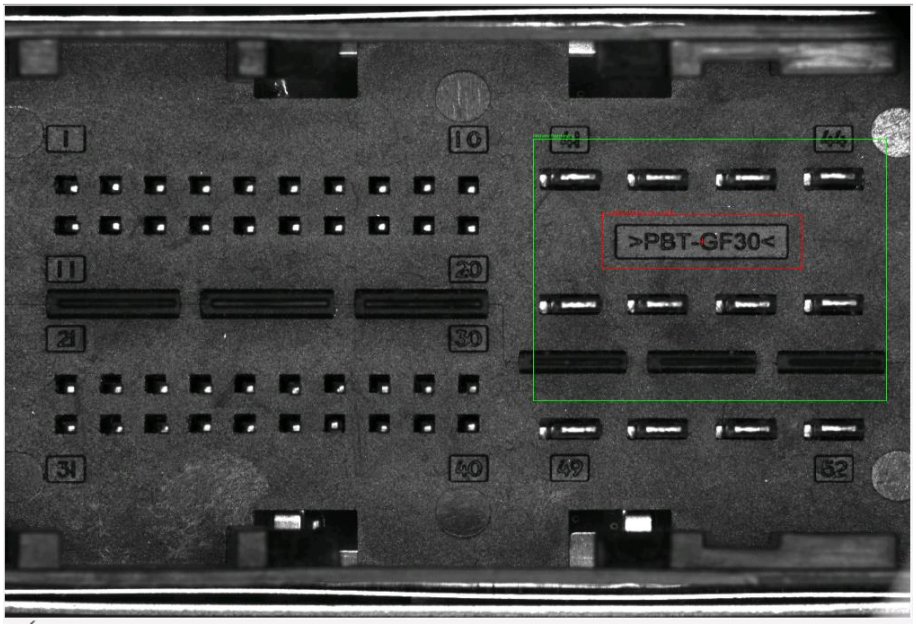

Figura 5.40 Pattern Matching

Para definir el patrón de búsqueda es necesario elegir de la imagen que es lo que se desea buscar, y guardarla en una dirección como una pequeña imagen. La dirección en la que tenemos localizada la imagen del patrón es:

C:\Users\tbench\Documents\Proyecto_Tesis\Main\Pattern\Pattern_1.png

Teniendo establecido el patrón de búsqueda, creamos un set de coordenadas en ese patrón, esto con la finalidad de poder establecer otras de búsqueda que se muevan basadas en el set de coordenadas que se ha establecido. Para el set de coordenadas aparecerán dos flechas dentro del patrón previamente definido (Figura 5.41).

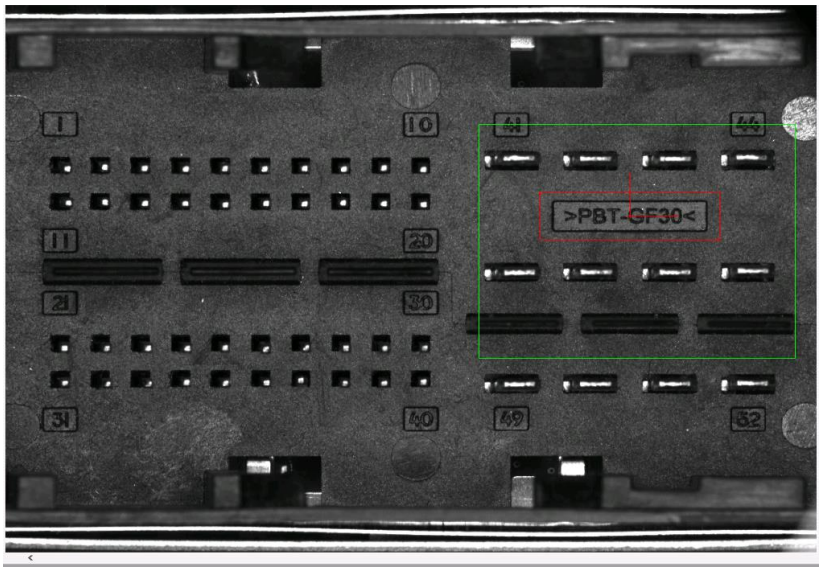

Figura 5.41 Set Coordinate system

Se puede definir el set de coordenadas con movimiento vertical, horizontal o angular. En este caso se definió con los 3 movimientos mencionados. (Figura 5.42)

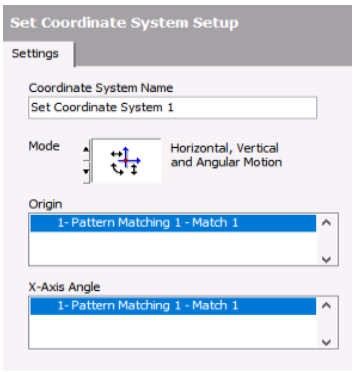

Figura 5.41 Set coordinate system setup

Habiendo estableció el sistema de coordenadas, el siguiente paso es crear mascara en cada una de las regiones que nos interesan (ROI), en este caso la región que nos interesa inspeccionar es 40 pines de la Parte B del Main Connector, para ello se crean mascaras individuales en forma de cuadro en cada uno de los pines. Las máscaras aparecerán en un cuadro verde, mientras que las regiones que no nos interesan estarán en color azul. (Figura 5.42)

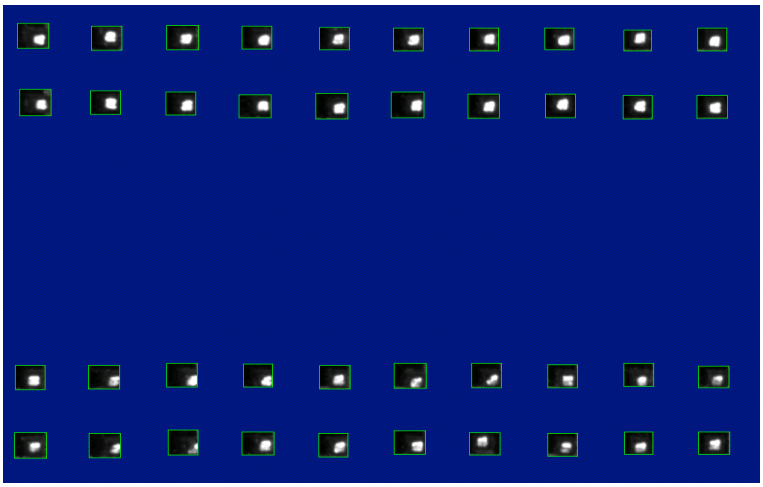

Figura 5.42 Image Mask

También la función de Image Mask nos permite la opción de seleccionar algunas figuras geométricas para crear las máscaras, o dibujarlas dependiendo de la necesidad que se tenga. Así como hacer mascara de los pixeles fuera del ROI o dentro del ROI (Figura5.43).

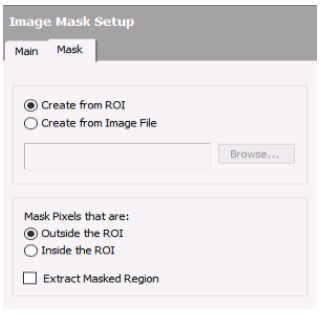

Figura 5.43 Image Mask Setup

Las cuatro funciones siguientes de procesamiento de la imagen serán:

5. Brightness. - Se utiliza esta función para modificar el brillo, el contraste y la gama de una imagen.
6. Threshold. – Su función es separar estructuras significativas en una imagen del resto de la imagen. El umbral (Threshold) establece todos los píxeles dentro del rango de umbral en 1 y establece todos los demás píxeles de la imagen en 0. La imagen resultante es una imagen binaria.
7. Particle Filter. - Elimina o mantiene partículas en una imagen según lo especificado por los criterios de filtro.
8. Particle Analysis. – Esta función ayuda a realizar mediciones de forma en partículas en la imagen.

La imagen de la Figura 5.44 nos muestra el complemento de la secuencia del script, siguiendo con el brillo, hasta el análisis de partículas.

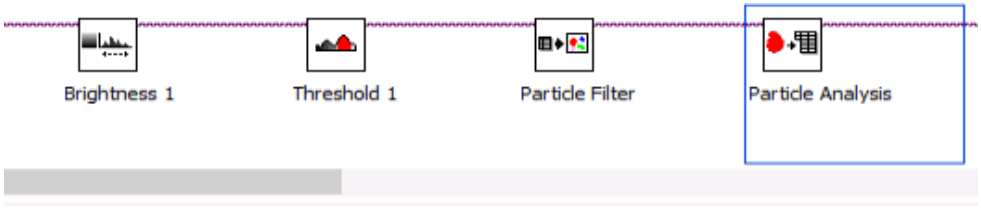

Figura 5.44 Funciones de procesamiento finales

Las funciones de la Figura 5.44 ayudaran en la detección de los 40 pines de la parte B del conector principal, puesto que las primeras 4 funciones ayudaron a establecer la región que se busca inspeccionar.

Dejando definidas las máscaras en la imagen que se está procesando, se le aplica la función de brillo a la imagen. En la Figura 5.45 podemos diferenciar el brillo de la imagen original (lado izquierdo) y la imagen procesada (lado derecho) al aplicar la función Brightness.

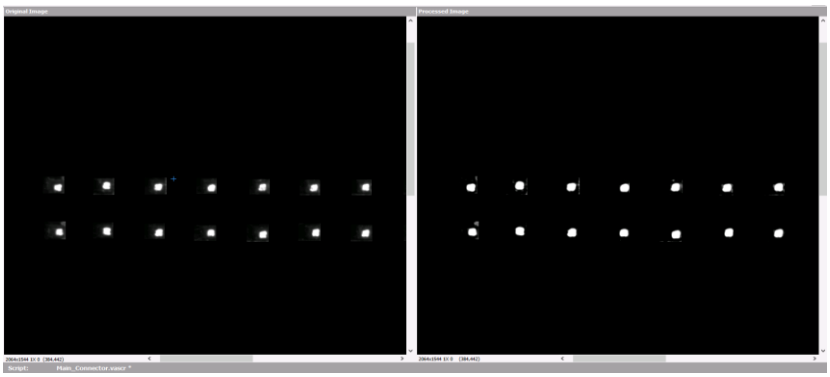

Figura 5.45 Imagen procesada con la función

Esta función ofrece tres valores, los cuales pueden ser modificados dependiendo de lo que sea más conveniente. Los valores son Brillo, Contraste y Gamma. Estos valores se modifican con un scroll, o ingresando el valor manualmente. Cuando procesa una imagen en color, puede aplicar modificaciones de brillo, contraste y gamma a cada plano de color por separado, o puede aplicar la misma transformación a todos los planos de color al mismo tiempo. Los valores seleccionados convenientes para la imagen procesada son brillo 220, contraste 70.00 y gamma 0.85. (Figura 5.46)

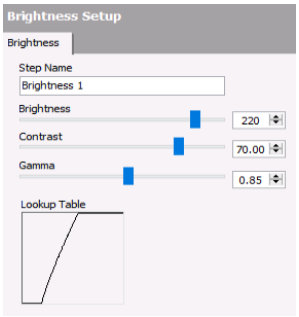

Figura 5.46 Brightness Setup

Al aplicar la función Threshold cada una de las partículas aparecerán en color rojo (Figura 5.47). En este caso podemos seleccionar si queremos que la función busque objetos brillantes, oscuros o grises, puesto que hemos agregado un filtro de brillo anteriormente, seleccionamos que la función busque objetos brillantes (Figura 5.48), así como el tipo de umbral que se desea realizar. El Threshold seleccionado es

manual, con un valor de 144. Los valores del umbral van de 0 a 255. Como nota, no se pueden ajustar los niveles de Threshold de los umbrales locales y automáticos. Si ninguna de las opciones local o automática funciona para la imagen, se debe seleccionar Umbral manual y ajustar hasta que todas las partículas de interés se resalten en rojo.

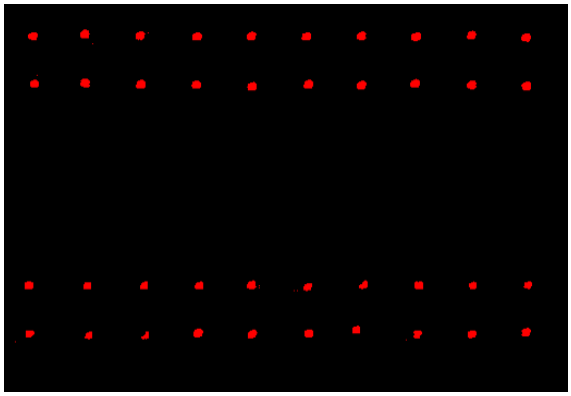

Figura 5.47 Threshold

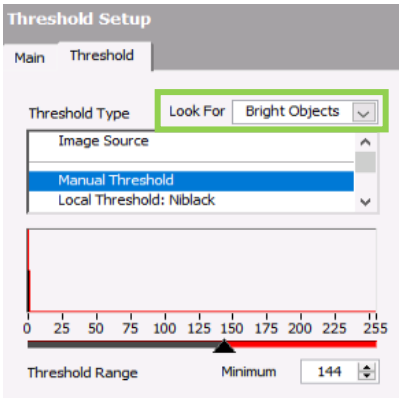

Figura 5.48 Threshold Look for

Al dejar establecido el nivel de Threshold conveniente, se agregará un filtro de partículas, el cual eliminará todas aquellas partículas basura que puedan causar ruido en el análisis posterior, para lograrlo utilizamos la función Particle Filter (Figura 5.49). Del lado derecho de la figura 5.49 podemos observar el resultado al aplicar el filtro de partículas.

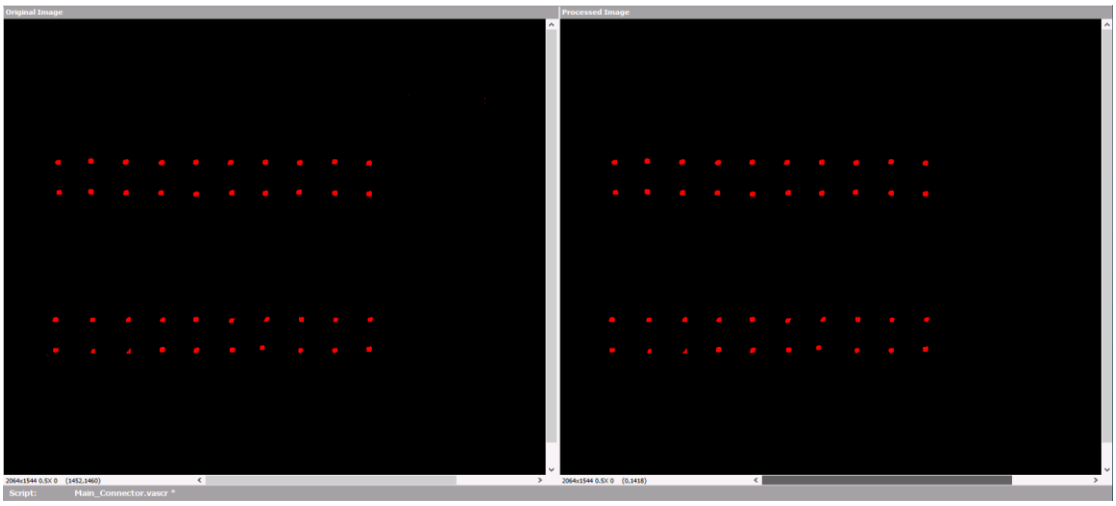

Figura 5.49 Filtro de partículas

La configuración para el filtro de partículas (Figura 5.50) se agrega por área, con un rango de parámetros de 0 a 9 en pixeles, que será la delimitación para remover las partículas encontradas en esos rangos.

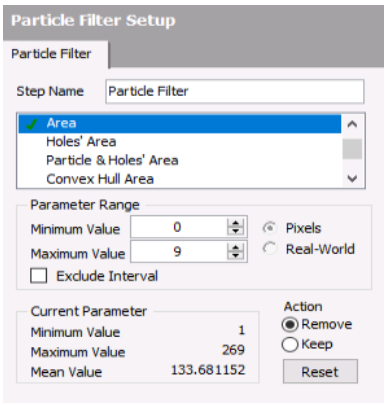

Figura 5.50 Particle Filter Setup

Por último, en el procesamiento de la imagen, se utilizará la función análisis de partículas (Figura 5.51). Vision Assistant calcula las medidas de cada partícula en la imagen y muestra los resultados, también asigna etiquetas numéricas a todas las partículas analizadas. Del lado izquierdo de la Figura 5.51 se puede observar la cantidad de partículas encontradas por la función, en este caso son 40 (Figura 5.52).

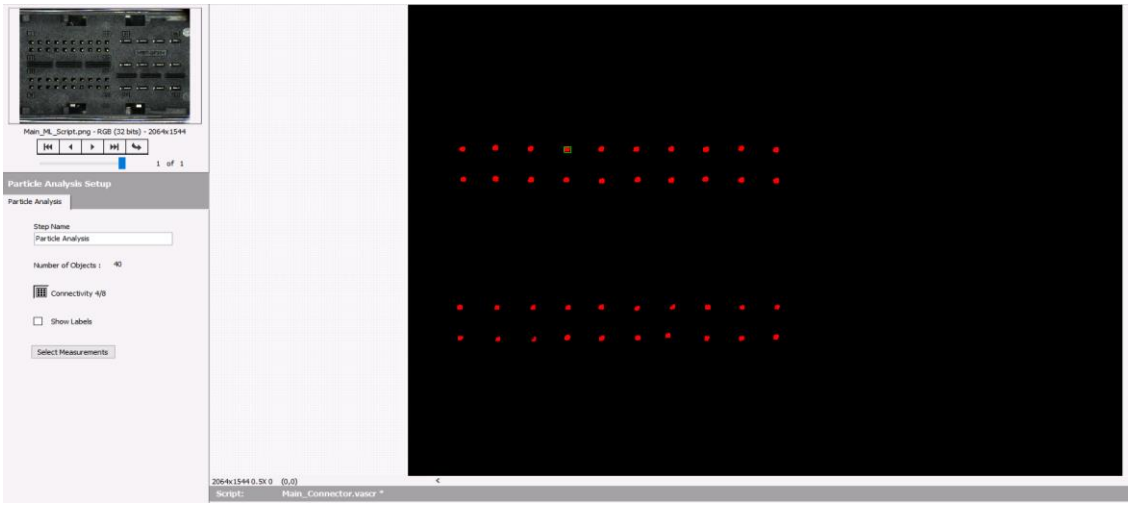

Figura 5.51 Análisis de partículas

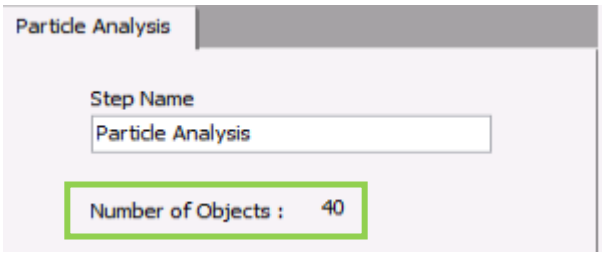

Figura 5.52 Número de objetos encontrados

Los resultados de este análisis los podemos ver en las Tablas B.1 y B.2 del Anexo B.

5.3.2- Generar código en C del script

Al finalizar la secuencia de procesamiento de la imagen en el script, Vision Assistant cuenta con 3 opciones para generar la solución de código:

1. LabVIEW VI

2. C Code
3. .NET Code

La opción que tomaremos sera crear C Code. Para ello es necesario ir a la opción de Tools en la interfaz principal de Vision Assistant y seleccionar Create C code. (Figura 5.53)

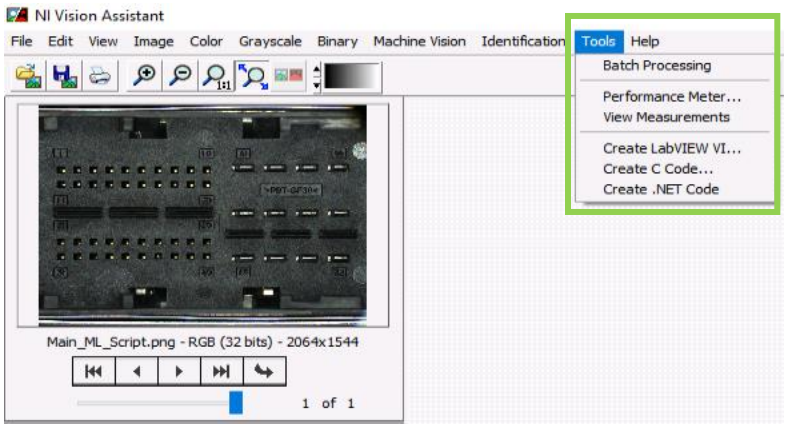

Figura 5.53 Crear Código C

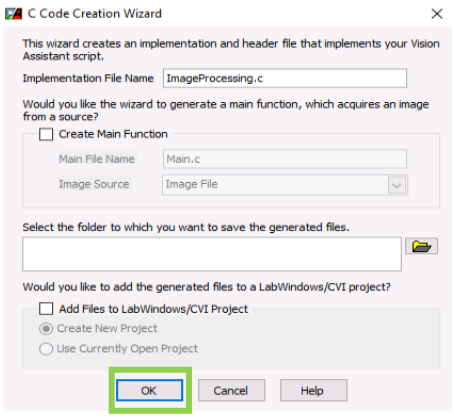

Figura 5.54 Opciones al Crear Código C

Al instante de presionar Create C Code aparece una ventana como se observa en la Figura 5.54, donde podemos cambiar el nombre del archivo que se creara con extensión .C, y elegir la ruta en donde se guardara el archivo creado. Habiendo seleccionado la ruta de guardado, solo dar “ok” (Figura 5.54) y aparecerá un mensaje

indicando que la creación del código en C fue exitosa, mostrándonos la dirección previamente elegida para localizar el código. (Figura 5.55)

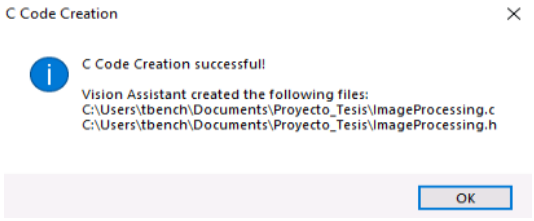

Figura 5.55 Creación del código C exitosa

La Figura 5.56 muestra parte del código creado por Vision Assitant. En esta sección de código se puede observar tres de las 8 funciones mencionadas, la máscara para la región de interés, el brillo con los valores establecidos, y el threshold manual con el valor definido.

```
//---------------------------------------------------------------------//
//                          Mask from ROI                              //
//---------------------------------------------------------------------//

VisionErrChk(IVA_MaskFromROI(image, roi1, FALSE, FALSE));

// Cleans up resources associated with the object
imaqDispose(roi1);

//---------------------------------------------------------------------//
//                            Brightness                               //
//---------------------------------------------------------------------//

bcgOptions.brightness = 220;
bcgOptions.contrast = 70.0000010281801;
bcgOptions.gamma = 0.85;

// Applies brightness, contrast, and gamma correction to the image
VisionErrChk(imaqBCGTransform(image, image, &bcgOptions, NULL));

//---------------------------------------------------------------------//
//                         Manual Threshold                            //
//---------------------------------------------------------------------//

// Thresholds the image.
VisionErrChk(imaqThreshold(image, image, 144, 255, TRUE, 1));

VisionErrChk(IVA_ParticleFilter(image, pParameter, plower, pUpper,
    pCalibrated, pExclude, 1, TRUE, TRUE));

VisionErrChk(IVA_Particle(image, TRUE, pPixelMeasurements, 9,
    pCalibratedMeasurements, 0, ivaData, 7));

// Releases the memory allocated in the IVA_Data structure.
IVA_DisposeData(ivaData);
```

Figura 5.56 Parte del código C de funciones de procesamiento

5.4.- Iniciar la aplicación para el proyecto en LabWindows/CVI

Al ejecutar LabWindows/CVI se observa una pantalla como en la Figura 5.57.

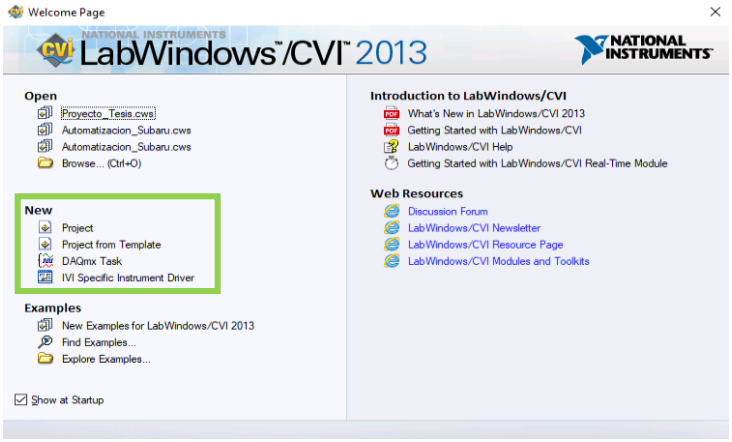

Figura 5.57 Inicio de LabWindows/CVI

Aquí se elige la opción Project from Template (Figura 5.57). En seguida se selecciona la opción User Interface Application (Figura 5.58), puesto que con este software se desarrolla la interfaz diseñada en el CAPÍTULO IV.

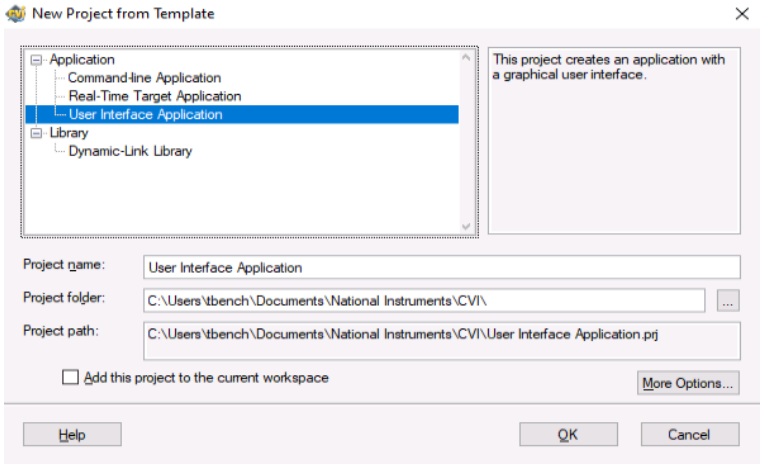

Figura 5.58 Aplicación de User Interface

Teniendo el template de la interfaz de usuario (Figura 5.59) podremos empezar a trabajar en el desarrollo, agregando cada uno de los elementos en el panel llamado “Untitled Panel” (Figura 5.59, cuadro verde), al que podemos cambiar el nombre, así como a cada una de las figuras que se estén agregando al panel. Del lado derecho de la interfaz de LabWindows/CVI tendremos los archivos que se usarán para ejecutar la interfaz de usuario (Figura 5.59 en amarillo), al dar clic derecho al panel de desarrollo (en verde Figura 5.59) se abrirán todos los elementos que podemos agregar a la interfaz de usuario (Figura 5.59 en azul). Debajo de la sección en amarillo de la Figura 5.59 encontramos las librerías (Figura 5.59 en rojo) de ayuda que brinda LabWindows/CVI para cada uno de los elementos que se deseen agregar y que sean necesarios utilizar.

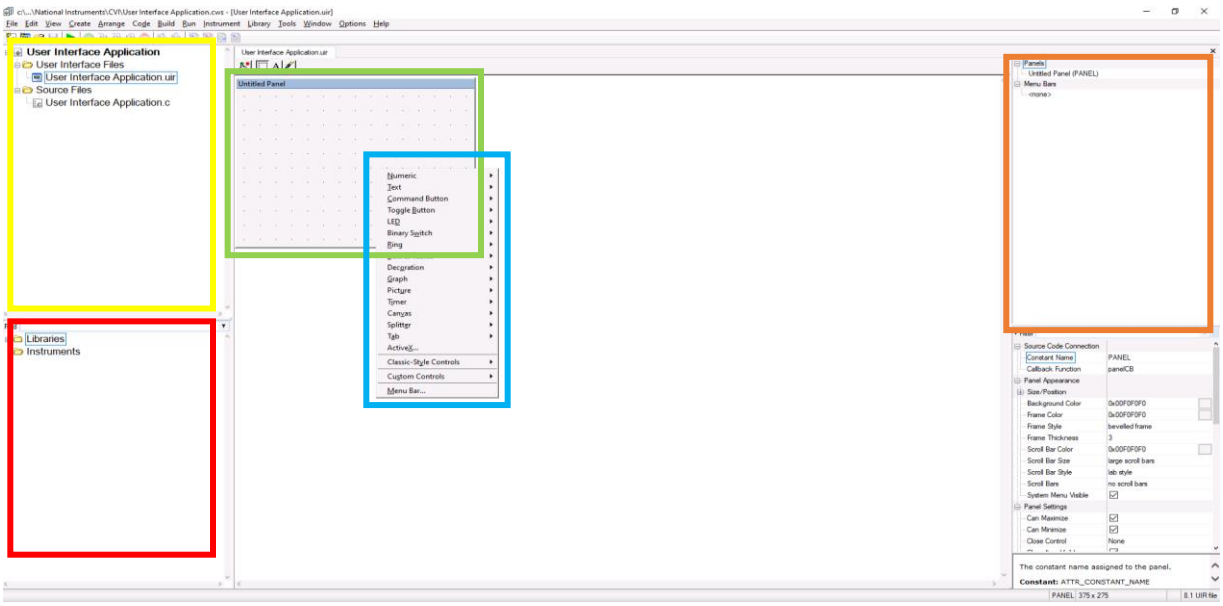

Figura 5.59 Template de User Interface

En la parte superior derecha de la interfaz de LabWindows/CVI (Figura 5.59 en naranja) se agregan cada uno de los elementos seleccionados en el panel de desarrollo de la interfaz de usuario diseñada.

5.4.1.- Elementos agregados y características del panel principal y secundario

El tamaño del panel de desarrollo será de alto:793 y ancho:1314. Los elementos agregados para el panel principal en cantidad y nomenclatura de LabWindows/CVI serán:

1. 5 – COMMANDBUTTON.
2. 1 – QUITBUTTON.
3. 9 – DECORATION.
4. 11 – TEXTMSG.
5. 1 – IMAGECTRL
6. 3 – TEXTBOX.

Los elementos anteriores se observarán en el panel principal de la siguiente manera:

Figura 5.60 Panel principal de la interfaz de usuario

Para los elementos del panel secundario (ventana de warning, Figura 5.61), solo necesitaremos 3 TEXTMSG.

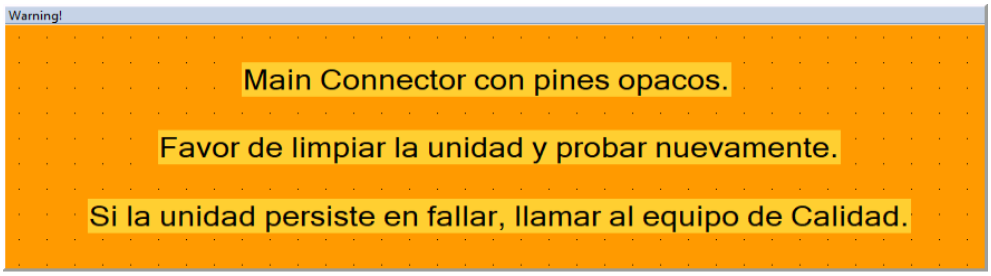

Figura 5.61 Panel secundario de la interfaz de usuario

5.4.2.- Librerías necesarias

Las librerías necesarias para la ejecución del proyecto son principalmente dos, la librería de SeaMAX.lib y la librería de niimaqdx.lib. La librería de SeaMax nos permitirá tener acceso a todas las funciones necesarias para controlar la SeaLevel 420U mediante nuestra interfaz. Y con la librería niimaqdx podremos controlar mediante funciones de código la cámara Mako G-319.

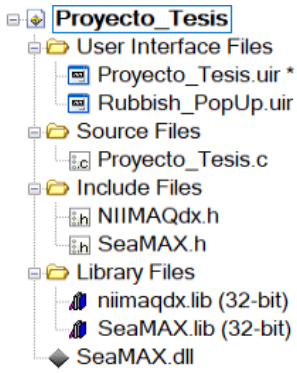

Figura 5.62 Librerías SeaMax y niimaqdx

En el caso de la librería niimaqdx se recomienda instalar la que viene en el paquete de software 18.5 de NI Vision Acquisition, de mayo del 2018. Esto para evitar un error de compilación del programa que no permita el control de la cámara.

5.4.3.- Código de LabWindows/CVI

En este apartado se presentará parte del código ejecutado en la interfaz de usuario, tanto para el control de la lámpara por medio de la SeaLevel 420U, el control de la cámara por medio de la librería mencionada anteriormente, así como los códigos de la solución de ML.Net y del generado por Vision Assistant para el conteo de los pines.

5.4.3.1.- Código para inicializar cámara y SeaLevel 420U

Para la inicialización de la SeaLevel que controla la lámpara, se creó una función (Figura 5.63) en la que se abre el “COM4” de comunicación por medio la función SM_Open (&seaMAXHandle [1]). En dado caso se tenga algún problema para inicializar la SeaLevel se mostrará un mensaje en pantalla (Figura 5.64). Este error puede salir debido a que no esté conectada la SeaLevel, o que el puerto de comunicación no esté configurado en el número de puerto de comunicación correcto, o debido a que se esté utilizando con el software para controlar la SeaLevel manualmente.

```
int Initialize (void)
{
    int error = 0;
    unsigned char ReadOutput[1];
    int SeaLevel_Model = 0 ,SeaLevel_commType = 0,SeaLevel_baudrate = 0,SeaLevel_parity = 0;

    /****************************************** Initialize SeaLevel ************************************************/
    if(!error)
    {
        error = SM_Open(&seaMAXHandle[1], "COM4");//Abre puerto de SeaLevel
        if(error)
        {
            MessagePopup("Error", "No se pudo inicializar SeaLevel COM4");
        }
    }
    if(!error)
    {
        error = SM_SelectDevice(seaMAXHandle[1],INPUTSOUTPUTS_SEALEVEL_ID);//Selecciona la Dirección de SeaLevel1 entradas
        error = SM_GetConfig(seaMAXHandle[1], &SeaLevel_Model ,&SeaLevel_commType ,&SeaLevel_baudrate ,&SeaLevel_parity);
        if(error)
        {
            MessagePopup("Error", "No se pudo inicializar SeaLevel 420U");
        }
    }
    SM_WriteDigitalOutputs(seaMAXHandle[1], Lampara [0], 1, Off);

    /****************************************** Initializing Cameras ************************************************/
    if(!error)
    {

        OpenCameras();     //Comentado para debug
         MessagePopup("Camara Mako G-319C", "                Inicializada");
    }
    /*****************************************************************************************************************/

    return error;
}
```

Figura 5.63 Código para inicializar SeaLevel y Cámara

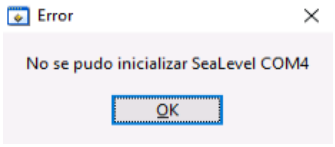

Figura 5.64 Error de inicialización de SeaLevel

En la Figura 5.63 también se puede observar el código para la inicialización de la cámara Mako G-319, en el cual se utiliza la función OpenCameras. Cuando la cámara se haya inicializado y la interfaz tome control de ella aparecerá un mensaje como el de la Figura 5.65.

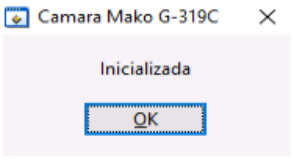

Figura 5.65 Mensaje de cámara Mako G-319 inicializada

Cabe mencionar que estas funciones se llamaran al momento de ejecutar el proyecto en LabWindows/CVI, como medida de contención para asegurarnos que el hardware esté conectado correctamente. Teniendo la inicialización exitosa podemos tomar control de la lámpara y de la cámara por medio del panel principal de la interfaz de usuario por medio de la sección ‘Image Tools” (Figura 5.66)

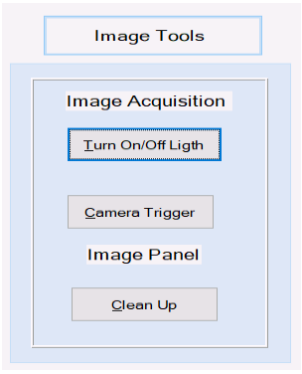

Figura 5.66 Herramientas de imagen en la interfaz principal

5.4.3.2.- Código de On/Off de la lámpara y de adquisición de imagen

Por medio del código de la Figura 5.67 se enciende y se apaga la lámpara mediante una función que utiliza los controladores de las salidas de la SeaLevel, por medio de la sentencia SM_WriteDigitalOutputs.

```
int Turn_On_Off_Lamp(int FIXTURE)
{
    unsigned char ReadOutput[1];
    int status = 0;

    SM_SelectDevice(seaMAXHandle[1], INPUTSOUTPUTS_SEALEVEL_ID);//Selecciona la direccion de SeaLevel Salidas
    SM_ReadDigitalOutputs(seaMAXHandle[1], Lampara [FIXTURE - 1], 1, ReadOutput);

    //Sleep(10);
    if(ReadOutput[0] == 0)
    {
        SM_WriteDigitalOutputs(seaMAXHandle[1], Lampara [FIXTURE - 1], 1, On);// Enciende la Lampara
        status = 1;
    }
    else
    {
    SM_WriteDigitalOutputs(seaMAXHandle[1], Lampara [FIXTURE - 1], 1, Off);// Apaga la Lampara
        status = 0;

    }
    return status;


}
```

Figura 5.67 Código de On/Off de la lámpara

Mediante la función IMAQdxSnap es que podemos adquirir la imagen utilizando LabWindows/CVI (Figura 5.68). La imagen obtenida se estará mandando al panel de imagen de la interfaz de usuario y se guardará en la dirección C:\\Images\\Main_ML.png, la cual será la dirección utilizada por el modelo de clasificación de imágenes de ML.Net para procesar la imagen.

```
int CVICALLBACK Image_Trigger (int panel, int control, int event,
                               void *callbackData, int eventData1, int eventData2)
{
    float gamma=0;
    float Exp_Cam=0;

    switch (event)
    {
        case EVENT_COMMIT:


            image = imaqCreateImage(IMAQ_IMAGE_U8, 7);

            IMAQdxSnap (session_MAIN, image);


            ImageControl_ConvertFromDecoration (panelHandle, PANEL_IMAGECTRL, image);

            ImageControl_SetZoomToFit(panelHandle, PANEL_IMAGECTRL, image);


            MessagePopup("Trigger", "Imagen Obtenida");

            imaqWritePNGFile2(image,"C:\\Images\\Main_ML.png",750,NULL,FALSE);


            break;
    }
    return 0;
}
```

Figura 5.68 Código Trigger de Cámara

5.4.3.3.- Evaluar imagen mediante ML y conteo de pines

Al momento de presionar el botón “Classify” (Figura 5.69) en la sección de pruebas del panel principal, éste ejecutará el modelo que se generó de clasificación de imágenes en ML.Net. También abrirá el archivo de texto en el que se guardaron los scores de clasificación para así desplegarlos en la ventana de resultados del panel principal de la interfaz. En la Figura 5.70 podemos observar el código utilizado para la sección de evaluar la imagen por medio de ML.

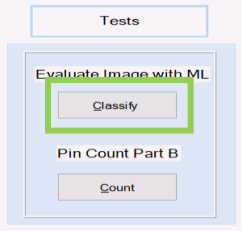

Figura 5.69 Sección de tests del panel principal

```
system("C:\\Users\\tbench\\source\\repos\\myMLApp.exe");
file = fopen("C:\\Users\\tbench\\\Documents\\Proyecto_Tesis\\MyTest.txt", "r");
fgets(buffer_file,80, file);
k = strlen(buffer_file);
badScore = strtok(buffer_file,",");
goodScore = strtok(NULL,",");

Real_Bad_Score = (atof(badScore));
Real_Bad_Score = Real_Bad_Score *100;
if (Real_Bad_Score > 70)
    {
        DisplayPanel (PANEL_2);
    }

else
Real_Good_Score = (atof(goodScore));
Real_Good_Score = Real_Good_Score*100;

    sprintf(badScore,"%f",Real_Bad_Score);  //Convertir numero real a string
    sprintf(goodScore,"%f",Real_Good_Score);
    strcat(badScore,"%");
    strcat(goodScore,"%");
```

Figura 5.70 Código para evaluar la imagen

Posterior a la evaluación de la imagen, se contarán los pines por medio de la función Pin_Count_Part_B, esta función ejecutará el código generado por Vision Assistant, utilizando las funciones del procesamiento de imagen en código C (Figura 5.71).

```
//-------------------------------------------------------------------------//
//                          Extract Color Plane                            //
//-------------------------------------------------------------------------//

// Creates an 8 bit image that contains the extracted plane.
VisionErrChk(plane = imaqCreateImage(IMAQ_IMAGE_U8, 7));

// Extracts the value plane
VisionErrChk(imaqExtractColorPlanes(image, IMAQ_HSV, NULL, NULL, plane));

// Copies the color plane in the main image.
VisionErrChk(imaqDuplicate(image, plane));

 //-------------------------------------------------------------------------//
//                           Pattern Matching                              //
//-------------------------------------------------------------------------//

// Creates and read the image template.
VisionErrChk(imageTemplate = imaqCreateImage(IMAQ_IMAGE_U8, 7));
VisionErrChk(imaqReadVisionFile(imageTemplate, templatePath, NULL, NULL));

// Set the angle range.
for (i = 0 ; i < 2 ; i++)
{
    angleRange[i].lower = angleRangeMin[i];
    angleRange[i].upper = angleRangeMax[i];
}

advancedOptions = (PMMatchAdvancedSetupDataOption*)malloc(numAdvOptions * sizeof(PMMatchAdvancedSetupDataOption));
for (i = 0 ; i < numAdvOptions ; i++)
{
    advancedOptions[i].matchSetupOption = advOptionsItems[i];
    advancedOptions[i].value = advOptionsValues[i];
}
```

Figura 5.71 Sección de código para el procesamiento de la imagen

5.4.4.- Sección de resultados en el panel principal de la interfaz

Del lado izquierdo de la interfaz de usuario en LabWindows/CVI (Figura 5.60) se encontrará la sección de resultados. En esta sección se agregaron tres cajas de texto, una para presentar el score de un buen main connector, otra para score de un mal main connector, y otra para la cantidad de pines contados. Estos resultados son arrojados por la evaluación de la imagen y el conteo de pines en la sección de pruebas.

El resultado del Score se tradujo a porcentaje para agregarlo a los textbox. En la Figura 5.72 podemos ver que se clasificó la imagen como un Good main connector, con un score del 84.37%, teniendo el score de un main connector con opacidad en 15.62%. También notamos que en la parte de Pin Count, aparece el número de pines en 40.

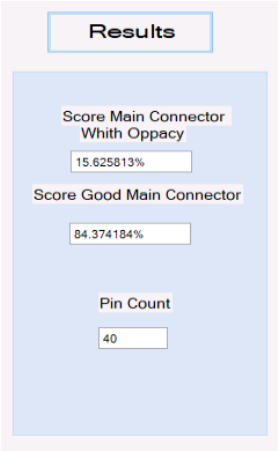

Figura 5.72 Sección de resultados en la interfaz de usuario en LabWindows/CVI

Para esta sección se toman los resultados y se almacenan en las variables Real_Good_Score, y Real_Bad_Score, se convierte el número real a string después de haberlo multiplicado por 100 para tener el valor en porcentaje, y se envía a las textbox por medio del control SetCtrlVal. (Figura 5.73)

```
Real_Good_Score = (atof(goodScore));
Real_Good_Score = Real_Good_Score*100;

    sprintf(badScore,"%f",Real_Bad_Score);  //Convertir numero real a string
    sprintf(goodScore,"%f",Real_Good_Score);

    strcat(badScore,"%");
    strcat(goodScore,"%");


ResetTextBox (panel, PANEL_TEXTBOX_BadScore, "");
ResetTextBox (panel, PANEL_TEXTBOX_GoodScore, "");
SetCtrlVal (panelHandle,PANEL_TEXTBOX_BadScore,badScore );
SetCtrlVal (panelHandle,PANEL_TEXTBOX_GoodScore,goodScore );
```

Figura 5.73 Sección de código para resultados de score

Como resultado final en el desarrollo, se obtuvieron las librerías necesarias del modelo de clasificación de imágenes para utilizarlas en la clasificación del *main connector* en la interfaz de LabWindows/CVI. De igual manera se generó la secuencia del conteo de pines por medio del script de Vision Assistant, en el que se obtuvo el código en C para ser implementado en la misma interfaz de LabWindows/CVI.

CAPÍTULO VI. PRUEBAS, RESULTADOS Y CONCLUSIONES

En este capítulo se muestran las pruebas realizadas para el funcionamiento de la interfaz, el modelo de clasificación de imágenes realizado en ML.Net, así como pruebas al script desarrollado en Vision Assistant. Se agrega una sección de pruebas de iluminación hechas previamente para el set up del hardware, en el que se exploran 4 diferentes colores de iluminación. Las pruebas para el modelo de clasificación de imágenes y el script de Vision Assistant, mediante la interfaz de usuario desarrollada en el CAPÍTULO V, se realizaron con 50 imágenes de *main connector*, previamente seleccionadas, esto debido a la falta de material para el ensamblado de radios en el Back – End de producción de MP Refresh.

6.1.- Pruebas de iluminación

Para las pruebas de iluminación se seleccionaron 4 tipos diferentes de luz, siendo los más comunes. Luz roja, luz verde, luz azul y luz blanca (Figura 6.1) fueron las luces elegidas. Esta prueba es con el fin de seleccionar la luz más adecuada para cumplir el objetivo.

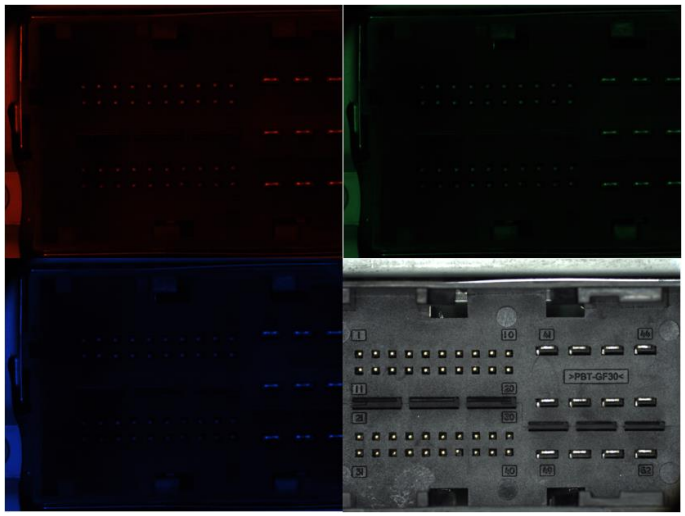

Figura 6.1 Pruebas de iluminación

La luz blanca es la más conveniente para el tipo de aplicación, como se muestra en la cuarta imagen de la Figura 6.1. Con la luz blanca resalta el brillo de los pines, lo que hace más fácil su detección, además que con alguna de las otras luces sería difícil crear un patrón como el creado en el apartado 5.3.1. Estas pruebas fueron realizadas previas al desarrollo de las aplicaciones de Visual Studio y de Vision Assistant. Se realizaron durante el Set Up del hardware, encendiendo la lámpara de manera manual, y obteniendo la imagen en el software de NI Max.

6.2.- Realización de las primeras pruebas del modelo de Clasificación de Imágenes

Durante el desarrollo del modelo de clasificación de imágenes por medio de ML.Net, se probó previamente en la interfaz de model builder, antes de utilizarlo con la interfaz de LabWindows/CVI. Para estas pruebas se utilizaron 40 imágenes a parte de las 50 imágenes para la prueba del modelo en la interfaz de LabWindows/CVI.

6.2.1.- Resultados de la prueba en model builder

En la Tabla 6.1 se muestran los resultados de las 20 imágenes buenas probadas previamente en model builder en el desarrollo del modelo de clasificación de imágenes en Visual Studio. Con un promedio de detección de 94% como un buen *main connector* y con 6% en detección de un mal *main connector*. Así mismo, en la Tabla 6.1 encontramos los resultados de las 20 imágenes malas probadas en la misma interfaz de model builder de Visual Studio, con lo cual tenemos un promedio de detección para imágenes malas del 92%, mientras que las clasifica como buenas en un promedio del 8%.

Tabla 6.1 Resultados de 20 imágenes buenas en model builder

Imagen Buena	Good_Main_Connector	Bad_Main_Connector
1	95%	5%
2	98%	2%
3	93%	7%
4	91%	9%
5	88%	12%
6	81%	19%
7	88%	12%
8	98%	2%
9	94%	6%
10	95%	5%
11	87%	13%
12	96%	4%
13	97%	3%
14	98%	2%
15	94%	6%
16	97%	3%
17	95%	5%
18	94%	6%
19	96%	4%
20	97%	3%
Promedio	**94%**	**6%**
Imagen Mala	**Good_Main_Connector**	**Bad_Main_Connector**
1	4%	96%
2	3%	97%
3	25%	75%
4	4%	96%
5	3%	97%
6	4%	96%
7	5%	95%
8	23%	77%
9	2%	98%
10	3%	97%
11	12%	88%
12	6%	94%
13	18%	82%
14	2%	98%
15	3%	97%
16	2%	98%
17	17%	83%
18	2%	98%
19	13%	87%
20	10%	90%
Promedio	**8%**	**92%**

6.3.- Pruebas del modelo de clasificación de imágenes y conteo de pines en la interfaz de LabWindows/CVI

Estas pruebas se realizaron con la interfaz de LabWindows/CVI y con 50 imágenes al azar tomadas de piezas de producción probadas en el mes de mayo del 2022. Las pruebas se realizaron de la siguiente manera:

Se ejecutó la interfaz de LabWindows/CVI en modo release. Con modo release nos referimos que no tendremos acceso al código fuente de la interfaz. Se evaluó cada imagen por medio del modelo de clasificación de imágenes y se contaron los pines del conector B del *main connector*. Los datos del conteo de pines y del porcentaje de clasificación se guardaron en un archivo de texto, para posteriormente ser analizados.

6.3.1.- Resultados del modelo de clasificación de imágenes

La Tabla 6.2 presenta los resultados de las 50 imágenes para el modelo de clasificación de imágenes y conteo de pines.

Tabla 6.2 Resultados de evaluación de 50 imágenes

Image	Unit-ID	Good_Main Connector	Bad_Main Connector	Pin_Count_CON_B
1	'TVPQN1602H000C'	97%	3%	40
2	'TVPQN1602H000D'	95%	5%	40
3	'TVPQN1602H000E'	92%	8%	40
4	'TVPQN1602H000F'	94%	6%	40
5	'TVPQN1602H00A0'	91%	9%	39
6	'TVPQN1602H00A1'	88%	12%	40
7	'TVPQN1602H00A2'	70%	30%	40
8	'TVPQN1602H00A3'	93%	7%	39
9	'TVPQN1602H00A5'	75%	25%	40
10	'TVPQN1602H00A6'	92%	8%	39
11	'TVPQN1602H00A7'	94%	6%	40
12	'TVPQN1602H00A8'	98%	2%	40
13	'TVPQN1602H00A9'	94%	6%	40

14	'TVPQN1602H00AA'	97%	3%	40
15	'TVPQN1602H00AB'	98%	2%	40
16	'TVPQN1602H00AC'	96%	4%	40
17	'TVPQN1602H00AF'	98%	2%	40
18	'TVPQN1602H00B0'	95%	5%	40
19	'TVPQN1602H00B1'	96%	4%	40
20	'TVPQN1602H00B2'	97%	3%	40
21	'TVPQN1602H00B3'	88%	12%	40
22	'TVPQN1602H00B4'	90%	10%	40
23	'TVPQN1602H00B5'	88%	12%	40
24	'TVPQN1602H00B6'	87%	13%	40
25	'TVPQN1602H00B7'	12%	88%	40
26	'TVPQN1602H00B8'	32%	68%	40
27	'TVPQN1602H00BA'	87%	13%	39
28	'TVPQN1602H00BB'	90%	10%	40
29	'TVPQN1602H00BC'	93%	7%	40
30	'TVPQN1602H00BD'	95%	5%	40
31	'TVPQN1602H00BF'	87%	13%	40
32	'TVPQN1602H00C0'	89%	11%	40
33	'TVPQN1602H00C1'	93%	7%	40
34	'TVPQN1602H00C2'	98%	2%	40
35	'TVPQN1602H00C4'	73%	27%	39
36	'TVPQN1602H00C5'	88%	12%	40
37	'TVPQN1602H00C6'	92%	8%	40
38	'TVPQN1602H00C7'	96%	4%	40
39	'TVPQN1602H00C8'	98%	2%	40
40	'TVPQN1602H00CA'	89%	11%	40
41	'TVPQN1602H00CB'	92%	8%	40
42	'TVPQN1602H00CC'	91%	9%	40
43	'TVPQN1602H00CD'	98%	2%	40
44	'TVPQN1602H00CE'	95%	5%	40
45	'TVPQN1602H00CF'	96%	4%	40
46	'TVPQN1602H00D0'	92%	8%	40
47	'TVPQN1602H00D1'	35%	65%	40
48	'TVPQN1602H00D2'	56%	44%	40
49	'TVPQN1602H00D3'	94%	6%	40
50	'TVPQN1602H00D4'	23%	77%	40

6.4.- Análisis de datos

Al analizar los datos de la Tabla 6.2, podemos notar que las imágenes clasificadas en un porcentaje de 75 a 100 % por el modelo de ML.Net son 42. Siendo 6 imágenes las clasificadas con un porcentaje menor (Tabla 6.3). En la Figura 6.2

podemos observar que esas 42 imágenes representan el 84% del total, teniendo buen funcionamiento el modelo de clasificación de imágenes desarrollado.

Tabla 6.3 Resultados de evaluación de 50 imágenes

% de Clasificación	Unidades (Imágenes)
0-25 %	2
25-50 %	2
50-75 %	4
75-100%	42

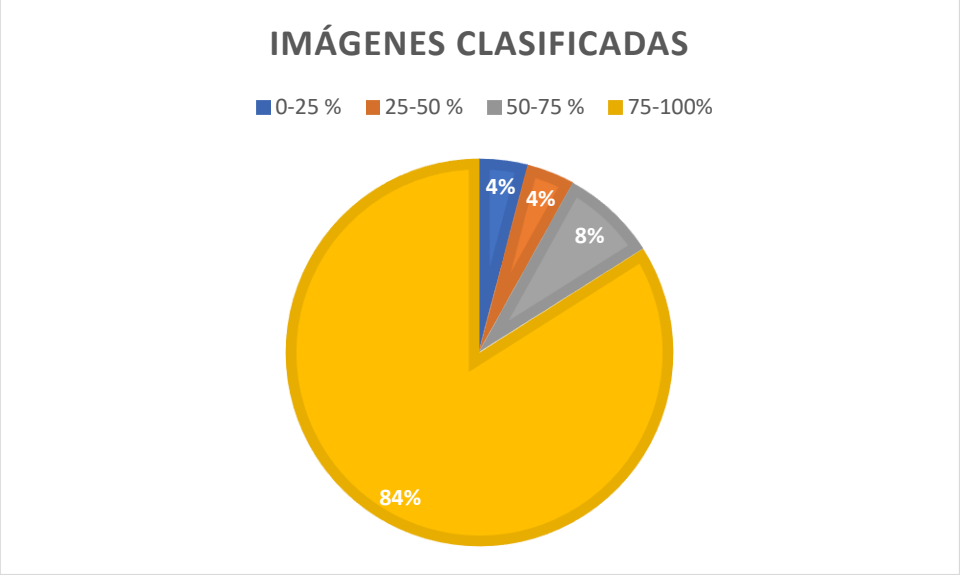

Figura 6.2 Porcentaje de las imágenes clasificadas

Así mismo, viendo los datos del conteo de pines de la parte B en la Tabla 6.2, podemos concluir que 45 unidades fueron contadas con los 40 pines, resultando en el 90%, mientras que el resto de las unidades (imágenes) se contaron con 39 pines (Tabla 6.4).

Tabla 6.4 Resultados del conteo de pines

	Conteo de Pines	
	Correcto	Incorrecto
Imágenes	45	5

6.4.1.- Matriz de confusión del modelo de clasificación de imágenes

La matriz de confusión es una herramienta que nos permite visualizar el desempeño de un algoritmo de aprendizaje automático. Esto permite ver qué errores y aciertos está teniendo el modelo a la hora de ser implementado.

Mediante una matriz de confusión binaria, ya que solamente tenemos dos posibilidades, una imagen clasificada como buena, con pines con el suficiente brillo para ser detectados y una imagen clasificada como mala, con pines opacos. Para esto surgen cuatro opciones:

- Unidades (imágenes de conector) que tienen suficiente brillo en los pines, y que el modelo las clasificó como buenas. Esto sería un verdadero positivo. (VP)
- Unidades con pines opacos (sin suficiente brillo en los pines) y el modelo las clasificó como malas. Esto sería un verdadero negativo. (VN)
- Unidades que tienen suficiente brillo en los pines y que el modelo las clasificó como malas, sería un falso negativo (FN).
- Unidades con pines opacos (sin suficiente brillo), clasificadas como buenas, sería un falso positivo (FP).

Las opciones anteriores se pueden resumir en la Figura 6.3.

Figura 6.3 Matriz de confusión binaria

En base a las opciones anteriores podemos calcular la exactitud que tiene el modelo de clasificación de una medición del valor verdadero. En forma practica la exactitud es la cantidad de predicciones positivas que fueron correctas. La fórmula siguiente permite calcular la exactitud (accuracy) del modelo de clasificación de imágenes:

$$Accuracy = \frac{(VP + VN)}{(VP + FP + FN + VN)}$$

Tabla 6.5 Valores para matriz de confusión de las 50 imágenes probadas

Verdadero Positivo	**45**
Verdadero Negativo	3
Falso Negativo	1
Falso Positivo	1

Con los valores de la Tabla 6.5 podemos calcular la exactitud del modelo de clasificación con las pruebas que se han hecho con 50 imágenes, para definir estos valores, se revisaron las imágenes mediante el script de Vision Assistant. Estos valores nos dan como resultado una exactitud del 96%, que comparado con el valor de exactitud otorgado por model builder de Visual Studio de un 98% solo tenemos un error del 2%.

6.5.- Comparación

Para comprender mejor el impacto de los resultados obtenidos en la Tabla 6.4 se ha realizado una comparación, entre los datos obtenidos por el sistema desarrollado vs el sistema de producción actual en la línea de MP Refresh. En datos obtenidos en el mes de mayo podemos notar que, de 70 unidades probadas, 19 presentaron falla en 21_Pin_Count_CON_B. Al. Por medio de una regla de proporcionalidad (regla de

3) podemos deducir que, de 50 unidades probadas, 14 fallarían en Pin_Count_CON_B (Tabla 6.6).

Tabla 6.6 Conteo de Pines en estación P&F de Producción

	Conteo de Pines	
	Correcto	Incorrecto
Imágenes	36	14

En la Figura 6.3 podemos ver que las fallas en Pin_Count_CON_B para 50 piezas probadas están impactando en un 28%, mientras que en el sistema desarrollado en este proyecto estarían impactando en un 10% (Figura 6.4). Lo que quiere decir que la reducción hecha sería en un 18% de las fallas. De 14 unidades que estarían fallando actualmente, implementando la mejora del sistema, solo estarían fallando 5.

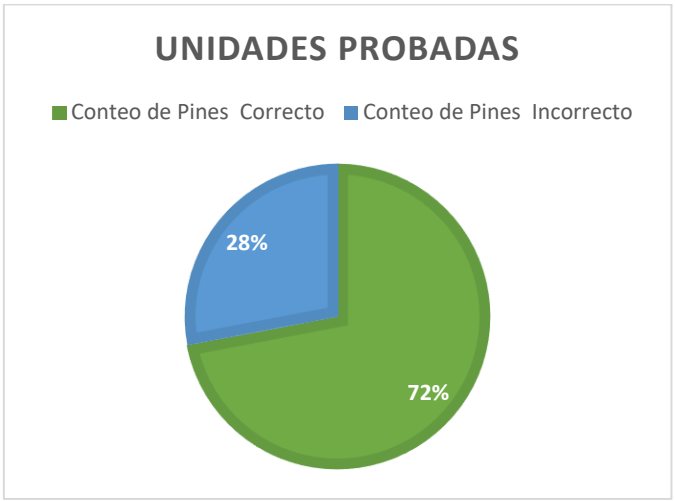

Figura 6.3 Porcentaje de las imágenes con falla en producción

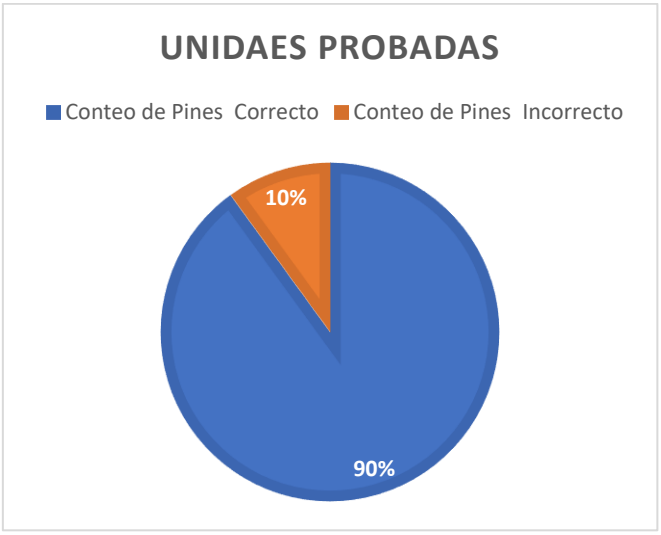

Figura 6.4 Porcentaje de las imágenes con falla en sistema desarrollado

6.6.- Conclusiones

En el presente trabajo, se muestra una solución a la falla descrita en el planteamiento del problema mediante la implementación de un modelo de clasificación de imágenes utilizando un algoritmo de Machine Learning. De igual forma se utiliza una técnica de procesamiento de imágenes para la inspección de la parte B del conector principal. Un algoritmo de Machine Learning con modelo de clasificación de imágenes tiene la capacidad de clasificar diferentes tipos de conectores, siendo una posible solución a la clasificación de uno o más conectores. Por medio del desarrollo de un sistema de visión artificial con un modelo de Machine Learning se tiene la utilidad de tener procesos mejor monitoreados, más robustos en la inspección, y detección de fallas con mejor precisión.

La hipótesis planteada al inicio del documento da respuesta a la evaluación del sistema que se ha tomado como solución, cumpliendo de manera satisfactoria la investigación y desarrollo de éste. La implementación de un sistema por medio de visión artificial con ayuda de un modelo de clasificación de imágenes en el área de Back End de MP Refresh (VP2 Refresh) contribuiría a una reducción del 18% de las fallas obtenidas en la prueba de Pin_Count_CON_B, siendo 18% más de la mitad de la reducción propuesta de 30.6% en la hipótesis, logrando una mejora en el rendimiento de la estación Pin&Fakra en cuanto a la detección de unidades con pines opacos, siendo esta del 90% de las 50 unidades con las que se realizó la prueba. Al tener un mensaje de advertencia en la interfaz ayuda a que el personal de producción, de ingeniería de calidad y de ingeniería de pruebas este al tanto de estas fallas en el momento de su detección, mejorando de esta manera el flujo del material. Al dar respuesta a la hipótesis planteada en este proyecto, se puede considerar que el objetivo se cumplió satisfactoriamente, ya que fue posible realizar el análisis, el diseño y desarrollo de un sistema de visión agregando un modelo de clasificación de imágenes por medio de Machine Learning, con capacidad de clasificación con una exactitud del 96% para *main connector* con pines opacos y *main connector* sin pines opacos.

6.6.1.- Trabajos futuros

Con el desarrollo del sistema de visión con el modelo de clasificación de imágenes, se pretendió lograr los mayores alcances posibles. Sin embargo, debido a que el set up mencionado en el desarrollo solo nos permite la obtención de la imagen del *main connector*, debido a que si queremos inspeccionar otro conector del radio de MP Refresh tendría que cambiar el set up de la obtención de la imagen por estar los conectores en diferente posición (ver Figura 1.1 en página 4), durante el desarrollo nos podemos percatar de trabajos que pueden ser implementados en un futuro. El primero sería que con esta base de conocimientos se puede verificar alguno de los otros conectores con los que cuenta el radio de MP Refresh, que son los siguientes:

1. USB, en sus dos colores.
2. Antena Fakra, en 5 diferentes colores, correspondientes a 5 diferentes antenas, FM/AM1, FM/AM2, DAB, SDAR y GPS.

Como segundo trabajo a futuro, el modelo de clasificación de imágenes podría explorar la solución de detección de housing (alojamiento plástico donde se colocan los pines, ver Anexo B) dañado en alguno de los conectores mencionados anteriormente, o en el mismo *main connector*. Ya que éste, también es un problema común en la línea de producción VP2 Refresh.

El tercer trabajo que se puede desarrollar a futuro es la verificación de la alineación de los pines respecto a los dibujos presentados en el Anexo A con la utilización de algún otro modelo de machine learning.

Como cuarto trabajo a futuro seria desarrollar el modelo de clasificación de imágenes a otras celdas de producción que tengan problemas similares presentados en este documento.

ANEXO A

Dibujos del Main Connector

Los dibujos presentados a continuación son los del housing del *main connector*, y de los pines:

En la Figura A.1 podemos ver las dimensiones de los pines utilizados para el ensamblado del *main connector*. Estos pines corresponden a la parte B.

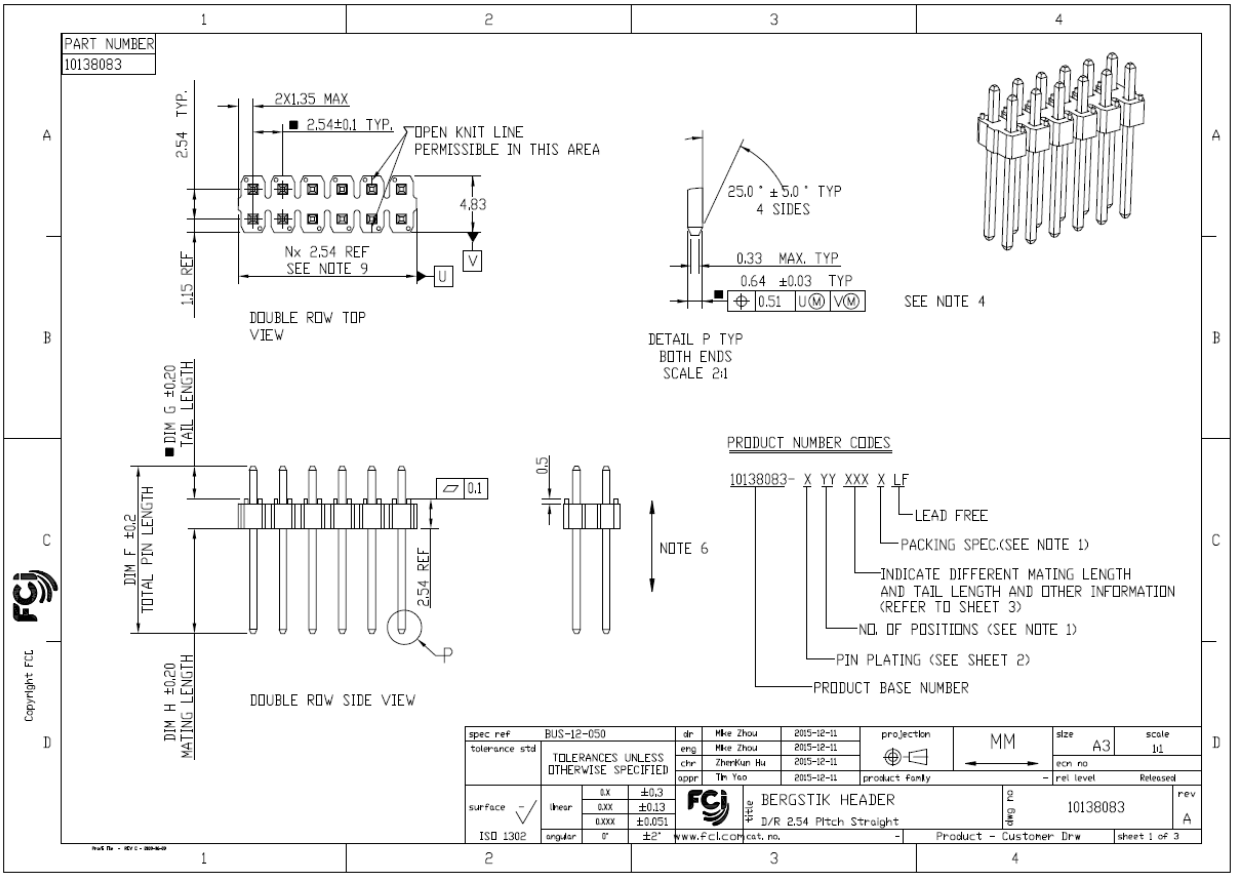

Figura A.1 Dibujo de los Pines del Main Connector

En la Figura A.2 podemos ver las dimensiones del housing (alojamiento) para el *main connector*. En este housing es donde estarán insertados los pines.

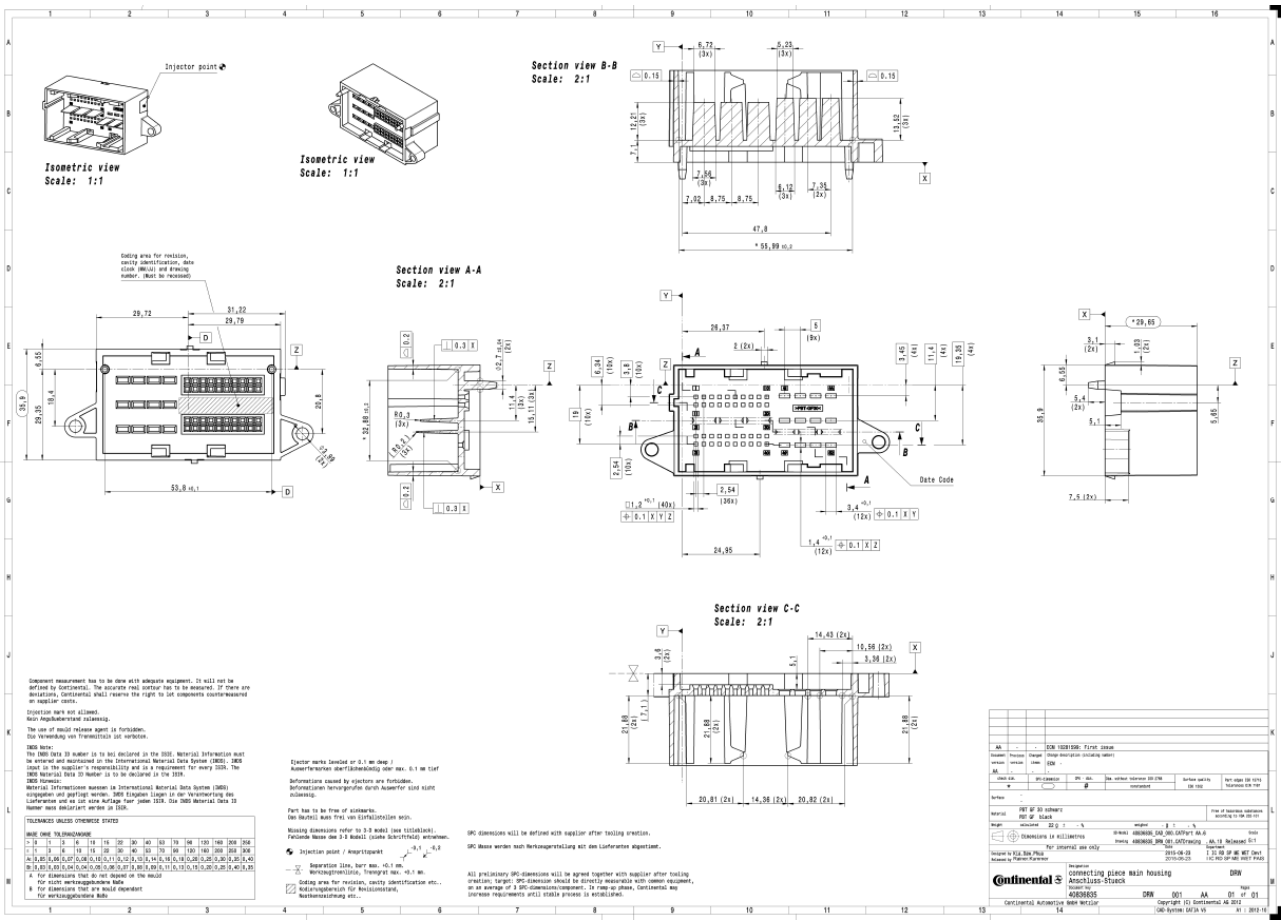

Figura A.2. Dibujo del Housing del Main Connector

ANEXO B

En las Tablas B.1 y B.2 podemos ver los resultados arrojados por Vision Assistant al momento de realizar el análisis de partículas. Estos resultados se presentan como trabajo de análisis a futuro. Recordando que son 40 pines en la parte B del *main connector*, la Tabla B.1 presenta 20 pines, mientras que la Tabla B.2 el resto de los pines.

Tabla B.1 Resultados arrojados por Vision Assistant 40 Pines

Object #	Center of Mass X	Center of Mass Y	Perimeter	Area	Elongation Factor	Hu Moment 4	Hu Moment 5	Hu Moment 6	Hu Moment 7
1	252.88	456.93	57.94	255	1.57	1.11E-07	1.41E-13	-1.23E-09	1.78E-13
2	962.94	459.07	57.25	244	1.74	3.33E-07	2.31E-13	-1.29E-09	-8.46E-13
3	354.30	459.84	57.41	250	1.75	6.51E-08	5.39E-14	1.56E-09	-7.20E-15
4	558.60	460.55	56.59	247	1.82	1.54E-08	6.90E-15	2.05E-10	-3.93E-16
5	760.95	460.53	57.51	248	1.67	3.75E-07	-1.97E-12	6.13E-10	-9.31E-13
6	157.44	460.98	56.18	236	1.67	5.63E-07	6.26E-12	4.17E-09	-2.09E-12
7	458.99	461.75	58.40	258	1.74	1.65E-07	7.55E-14	3.31E-09	1.18E-13
8	659.63	461.69	57.68	241	1.88	6.06E-07	2.05E-12	4.98E-09	-2.42E-12
9	861.23	462.11	59.35	260	1.76	6.10E-08	4.96E-14	2.82E-10	3.34E-14
10	1063.25	463.63	57.44	253	1.68	1.16E-07	-2.70E-13	1.63E-09	2.14E-13
11	253.32	555.90	60.69	269	1.68	1.51E-06	9.31E-12	1.27E-08	1.31E-11
12	859.39	557.24	59.09	254	1.70	2.02E-07	7.75E-13	3.61E-09	-4.52E-13
13	160.27	557.77	55.03	224	1.63	1.15E-08	2.38E-15	1.17E-11	-5.49E-15
14	355.51	559.10	60.86	261	1.75	3.86E-08	6.85E-14	-2.25E-10	-4.72E-14
15	663.32	559.00	58.11	259	1.77	3.84E-08	-1.45E-14	-7.79E-10	-5.57E-14
16	963.33	560.00	57.33	250	1.68	7.05E-07	-3.79E-12	3.55E-09	-6.45E-12
17	457.37	559.68	56.69	238	1.58	3.06E-07	-3.58E-13	-2.16E-09	-8.68E-13
18	760.91	560.58	59.08	255	1.75	2.14E-08	1.37E-14	-4.03E-10	-3.55E-16
19	559.69	563.15	59.08	254	1.70	2.30E-07	3.97E-13	3.59E-09	-5.50E-13
20	1063.64	562.52	59.19	261	1.63	6.73E-07	-2.31E-12	-1.40E-09	-6.04E-12

Tabla B.2 Complemento de Tabla B.1

Object #	Center of Mass X	Center of Mass Y	Perimeter	Area	Elongation Factor	Hu Moment 4	Hu Moment 5	Hu Moment 6	Hu Moment 7
21	558.91	966.59	55.86	235	1.79	1.59E-07	9.37E-14	2.81E-09	-6.12E-13
22	765.22	966.64	54.24	192	2.18	1.13E-05	-6.71E-10	-8.78E-07	-1.35E-10
23	362.16	967.96	53.10	199	2.14	7.67E-06	3.06E-10	3.22E-07	-3.05E-10
24	150.18	967.64	57.46	242	1.74	1.04E-07	8.39E-14	1.09E-09	-4.44E-14
25	462.54	968.23	53.12	209	1.80	5.02E-07	3.33E-12	1.22E-08	-5.06E-12
26	866.21	967.35	57.80	232	1.82	6.04E-07	2.80E-12	-2.86E-09	9.77E-13
27	965.51	968.01	51.53	184	1.72	9.62E-08	-9.33E-14	-1.35E-09	-1.04E-13
28	1066.69	966.93	50.92	188	1.81	1.90E-07	5.10E-13	5.50E-09	-2.90E-13
29	257.63	968.17	51.80	196	2.01	4.38E-07	8.98E-13	1.04E-08	-1.38E-12
30	662.45	970.38	52.61	201	2.11	2.94E-06	3.22E-11	1.56E-07	-1.27E-11
31	751.62	1057.99	52.04	198	1.83	4.09E-07	1.68E-12	8.30E-09	4.46E-13
32	1063.35	1062.47	57.51	236	1.77	2.01E-07	2.83E-13	-3.41E-09	-2.25E-13
33	460.82	1064.11	57.78	249	1.73	8.30E-08	-3.44E-14	7.67E-11	-2.06E-14
34	664.46	1065.17	56.36	235	1.60	3.42E-07	1.33E-12	2.59E-09	3.68E-13
35	560.27	1066.34	58.94	243	1.97	8.18E-07	-1.89E-12	-1.99E-08	2.37E-12
36	151.01	1065.66	52.29	206	1.79	2.11E-07	5.26E-13	5.41E-09	1.30E-12
37	863.56	1066.36	51.94	200	1.80	2.16E-06	3.49E-11	6.63E-08	-8.09E-12
38	963.68	1066.47	54.29	214	1.76	5.01E-07	1.26E-12	9.52E-09	1.36E-12
39	259.84	1068.95	49.03	164	2.21	9.66E-06	5.64E-10	4.88E-07	1.31E-10
40	364.43	1069.93	49.93	136	2.72	0.000233	-1.02E-08	-5.55E-06	2.18E-07

REFERENCIAS

Continental. (2022). *User Experience.* Obtenido de Continental: https://www.continental.com/en/press/fairs-events/techshow-around-the-world/live-event/user-experience/

Continental, M. (2022). *Compañia.* Obtenido de Continental Mexico: https://www.continental.com/es-mx/compania/continental-mexico/

Durwad K. Sobek II and Art Smalley. (2008). Understanding A3 Thinking. New York: Taylor & Francis Group.

Education, I. C. (3 de June de 2020). *Artificial Intelligence. Recuperado el 10 de Junio del 2022*. Obtenido de IBM: https://www.ibm.com/cloud/learn/education

Hao Gong, Xinjian Deng, Jianhua Liu, Jiayu Huang. (19 de October de 2021). Quantitative loosening detection of threaded fasteners using vision-based. *Elsevier B.V.*, págs. 1-15.

INSTRUMENTS, N. (2022). *¿Qué es el Módulo Vision Development?* Obtenido de ni.com: https://www.ni.com/es-mx/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module.html

INSTRUMENTS, N. (2022). *¿Qué es LabWindows™/CVI?* Obtenido de ni.com: https://www.ni.com/es-mx/shop/electronic-test-instrumentation/programming-environments-for-electronic-test-and-instrumentation/what-is-labwindows-cvi.html

INSTRUMENTS, N. (24 de May de 2022). *What is NI Measurement & Automation Explorer (NI MAX)*. Obtenido de ni.com: https://www.ni.com/es-mx/support/documentation/supplemental/21/what-is-ni-measurement---automation-explorer--ni-max--.html

Jhinson Edgar Coyago Tomalo, Jonathan Javier Coyago Tomalo. (2019). *Implementación de un sistemas prototipode luces frontales con segmentación para automoviles empleando Técnicas de visión artificial difusas.* Riobamba-Ecuador: Escuela Superior Politécnica de Chimborazo.

Jones, Paul Viola and Michael. (2001). Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade. *Mistubishi Electric Research Lab, Cambridge, MA*, 8.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. (2015). *Deep Residual Learning for Image Recognition.* arXiv.

Luis Arturo Medina Muñoz, Samuel González-López, Jesús Antonio Mayorquín Robles, Gabriel Antonio López Valencia. (2019). Diseño de interfaces de inspección de calidad utilizando redes neuronales y visión artificial para la industria de manufactura. *Research in Computing Science*.

Luis Quintanilla, David Coulter, Nick Schonning. (15 de September de 2021). *What is ML.NET and how does it work?* Obtenido de docs.microsoft.com: https://docs.microsoft.com/en-us/dotnet/machine-learning/how-does-mldotnet-work?WT.mc_id=dotnet-35129-website

McCarthy, J. (24 de November de Revised 2004). WHAT IS ARTIFICIAL INTELLIGENCE? *Stanford University, Computer Science Department* , pág. 2.

Microsoft. (2022). *.NET Machine Learning & AI*. Obtenido de dotnet.microsoft.com: https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet

Microsoft. (2022). *Visual Studio 2022*. Obtenido de visualstudio.microsoft.com: https://visualstudio.microsoft.com/es/vs/

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (11 de April de 2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, págs. 211-252.

Peña Lorenzo, J. M. (2020). *Aplicación de técnicas de aprendizaje profundo (deep learning) para la detección de objetos en Industria 4.0 : Tesis.* Valladolid: Universidad de Valladolid.

Pressman, R. S. (2010). Diseño de la interfaz de usuario. En *Ingeniería del software. Un enfoque práctico* (págs. 265 - 294). Mc Graw Hill.

Rodarte Leyva, E. (2021). *Desarrollo de un sistema basado en visión artificial para evaluación de la clidad en el ensamble de conectores en bolsas de aire automotrices.* Hermosillo, Sonora, Mex: Universidad de Sonora.

Shook, J. (2008). Managing to Learn Using the A3 management process to solve problems, gain agreement, mentor, and lead. Cambridge, MA, USA: The Lean Enterprise Institute.

Stuart Russell and Peter Norvig. (1995). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Terry G. Lee, Gordon Hogenson. (29 de Abril de 2022). *IDE de Visual Studio.* Obtenido de Documentación técnica de Microsoft: https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*.