



**DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO
PARA PIEZAS FALTANTES DE MOLDES EN LA
INDUSTRIA DE ALIMENTOS MEDIANTE EL
PROCESAMIENTO DE IMÁGENES CON MATLAB**

Daniel Camilo Buitrago Lopez

10561522311

Universidad Antonio Nariño

Programa Ingeniería Biomédica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Bogota, Colombia

2022

**DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO PARA PIEZAS
FALTANTES DE MOLDES EN LA INDUSTRIA DE ALIMENTOS MEDIANTE
EL PROCESAMIENTO DE IMÁGENES CON MATLAB**

Daniel Camilo Buitrago Lopez

Proyecto de grado presentado como requisito parcial para optar al título de:

Ingeniero Biomédico

Ascensor:

MSc. Carlos Arturo Garcia Gomez

Línea de Investigación:

Procesamiento de imágenes, automatización en la industria

Universidad Antonio Nariño

Programa Ingeniería Biomédica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Bogota, Colombia

2022

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____.

Cumple con los requisitos para optar

Al título de _____.

Firma del Tutor

Firma Jurado

Firma Jurado

TABLA DE CONTENIDOS

Agradecimientos	IX
Resumen	XI
Abstract	XIII
Lista de Tablas	XV
Lista de Figuras	XV
1 Introducción	1
1.1 Planteamiento del problema	3
1.2 Justificación	4
1.3 Objetivos	5
1.3.1 Objetivo general	5
1.3.2 Objetivo específico	5
1.4 Alcance	6
2. Marco teórico	6
2.1 Imágenes en escala de grises	7
2.2 Imágenes RGB	8
2.3 Convolución	9
2.4 Convolución de imágenes en escala de grises	10
2.5 Convolución de imágenes RGB	10
2.6 Redes convolucionales	12
3.7 Aprendizaje de características, capas y clasificación	13
2.8 Valores de peso y sesgo compartidos	14

2.9 Clasificación de moldes utilizando la red neuronal convolucional de googlenet (cnn)	14
3. ESTADO DEL ARTE	16
3.1 Norma Haccp	17
4. METODOLOGIA	18
4.1 Análisis del molde	19
4.2 Toma de imágenes	22
4.3 Procesamiento del algoritmo	26
5. PROTOCOLO DE PRUEBAS	32
6. RESULTADOS Y CONCLUSIONES	38
7. BIBLIOGRAFIA	41
ANEXOS	43

LISTA DE FIGURAS

Figura 1. Super Cavemil 800, área de inyección de chocolate, Fuente del autor.	2
Figura 1.1. Sistema de depositadora MLP600 Fuente del autor	2
Figura 1.2. Vista interna en cava (sistema de enfriamiento) y solidificación del chocolate, Fuente del autor.	3
Figura 1.3. Moldes con fisuras dentro de la línea de producción, Fuente del autor	4
Figura 2. Ejemplo de imagen digital - Fuente: bibliografía [7]	7
Figura 2.1 - Imágenes en escala de grises - Fuente: bibliografía [8].	7
Figura 2.2 Imagen RGB - Fuente: bibliografía [8]	8
Figura 2.3 Convolución de imágenes - Fuente: bibliografía [9]	10
Figura 2.4 Convolución de imágenes a escala de grises - Fuente: bibliografía [10].	10
Figura 2.5 Convolución de imágenes RGB - Fuente: bibliografía [11]	10
Figura 2.6 - Flujo de trabajo de deep learning. Las imágenes se envían a CNN - Fuente: bibliografía [12]	12
Figura 2.7 Capas y clasificación - Fuente: bibliografía [12]	13
Figura 2.8 Capas y clasificación - Fuente: bibliografía [15]	14
Figura 2.9 Clasificación usando google net - Fuente: bibliografía [20]	16
Figura 4 – Diagrama de bloques- Fuente del autor	19
Figura 4.1 – Vista Superior- Fuente del autor.	20
Figura 4.2 – Betas de ruptura- Fuente del autor	20
Figura 4.3 – Vista posterior- Fuente del autor	20
Figura 4.4 – Producto contaminado con pieza de molde - Fuente del autor	21

Figura 4.5 – Vista lateral derecha- Fuente del autor	21
Figura 4.6 – Ruptura de molde en área de sujeción- Fuente del autor	21
Figura 4.7 Figura de representación del sistema de captura de imágenes. Fuente del autor. .	23
Figura 4.8 Posición de montaje de cámaras e laminación en línea de producción Imagen del autor.	24
Figura 4.9 Imágenes capturadas mediante el sistema propuesto. Fuente del autor	25
Figura 4.9.1 Imágenes descartadas durante la creación de la base de datos. Fuente del autor..	26
Figura 4.9.2 Base de datos para las imágenes. Fuente del autor..	26
Figura 4.9.3 Arquitectura de la red neuronal durante el primer procesamiento. Fuente autor. .	27
Figura 4.9.4 Arquitectura de la red neuronal durante el segundo procesamiento. Fuente autor.	32
Figura 4.9.5 Arquitectura de entrenamiento de la red neuronal. Fuente del autor.	31
Figura 5. Imágenes de resultados. Fuente del autor.	32
Figura 5.1. Top de predicciones de clasificación. Fuente del autor.	32
Figura 5.2. Matriz de confusión. Fuente del autor	32
Figura 5.3. Visualización de la anomalía mediante el uso del algoritmo de aumento. Fuente del autor.	36
Figura 5.4. Visualización de la anomalía. Fuente del autor.	39

LISTA DE TABLAS

Grafico 1. Numero de eventos registrados de moldes rotos según años	18
Tabla 1. Tabla de referencias utilizadas en produccion.	18
Tabla 2. Clasificación de Imágenes en base de datos.	22
Tabla 3. Resolución de imágenes en base de datos	23
Tabla 2. Resultados de validación con la puesta en marcha del algoritmo. Fuente autor. .	35
Grafico 2. Porcentaje de validación con la puesta en marcha del algoritmo. Fuente autor	35
Grafico 3. Protocolo para pada de produccion cuando se identifica una alerta del algoritmo. Fuente del autor.	37

Dedico con todo mi corazón este trabajo de grado a mi familia, su amor y apoyo hicieron esto posible.

Si he logrado ver más lejos, ha sido porque he subido sobre hombros de gigantes.

Isaac Newton

AGRADECIMIENTOS

Agradezco a Dios por permitirme vivir esta experiencia que cambio mi vida, a toda mi familia por su apoyo y oraciones para que culminase con éxito esta etapa, ustedes siempre han sido el motor que impulsara mis sueños y anhelos, siempre han sido los mejores guias de mi carrera.

Al Ingeniero Carlos Garcia, su paciencia y constancia este trabajo fueron claves en la realización de este. A todos los docentes que tuve durante mi carrera, sus palabras fueron sabias y me formaron en incontables aspectos, donde quiera que la vida me lleve sus conocimientos rigurosos, llevaré conmigo. Gracias por compartir sus conocimientos por su paciencia, tolerancia y virtudes.

A Casa luker, a mis jefes y compañeros de trabajo, por trasformarme personal y profesionalmente, por tanta paciencia y apoyo para que estudiara y culminara con éxito cada semestre. En ello estare agradecido eternamente.

A mis compañeros y amigos que de una u otra forma influyeron en mi proceso de formacion.

A ti trompitas que me has apoyado en todo momento, gracias por tu paciencia, dedicacion y apoyo en tantas noches dedicadas en este trayeto educativo.

A todos infinitas gracias

RESUMEN

La industria chocolatera transforma la producción cacaotera en productos terminados que luego se exportan. Durante la producción existe el riesgo de que los moldes separtan puesto que en los procesos de fabricación intervienen partes mecánicas de los equipos que hacen que los moldes se deterioren, la probabilidad de que un trozo de molde caiga dentro de una tableta y llegue a un consumidor final es alta, lo que puede generar sanciones legales y pérdidas económicas para la compañía al generar un riesgo para el consumidor, para disminuir esto, el presente proyecto hace un análisis de las variables que intervienen en el proceso de inyección de chocolate en la línea súper cavemil 800 e implementa un sistema de captura de imágenes de los moldes que componen la máquina de producción y mediante el procesamiento de patrones de la red neuronal googlenet de Matlab permite el reconocimiento de piezas faltantes en los moldes pudiendo así alertar al operador de la máquina la presencia de una anomalía en ellos activando un protocolo de alerta para la producción.

PALABRAS CLAVE: Procesamiento de imágenes, chocolate, Deep learning, súper cavemil 800, Molde roto, reconocimiento de patrones, Googlenet.

ABSTRACT

The chocolate industry transforms cocoa production into finished products that are then exported. During production there is a risk of moulds breaking as mechanical parts of the equipment that cause moulds to deteriorate are involved in the manufacturing process, the probability of a piece of mould falling into a tablet and reaching a final consumer is high, which can generate legal penalties and economic losses for the company by creating a risk for the consumer, to reduce this, the present project makes an analysis of the variables involved in the chocolate injection process in the line super cavemil 800 and implements a system of image capture of the molds that make up the production machine and through the Matlab googlenet neural network image processing allows the recognition of missing parts in the molds can thus alert the operator of the machine the presence of an anomaly in them activating an alert protocol for production.

KEYWORDS: Image processing, chocolate, deep learning, super cavemil 800, broken mold, pattern recognition, Googlenet.

1. INTRODUCCION

La industria del cacao y sus derivados ha venido en aumento en el país gracias al aporte de las empresas para la restitución de cultivos ilícitos para que estos sean reemplazados por siembra de cacao, este aporte ha hecho que las hectáreas hayan crecido un 24,2% desde el 2018 a hoy en el neto de las exportaciones en toneladas nacionales. [1]

Hoy Colombia exporta diferentes variaciones de chocolates, cocoas, manteca de cacao, licor de cacao, cacao en grano entre otros, a más de 74 países. Departamentos como Antioquia, Huila, Valle del Cauca, Magdalena, Nariño, Caldas, Tolima son los principales cultivadores y gracias al trabajo de miles de personas que componen la industria desde el cultivador hasta la producción final es posible la obtención de estos productos. [2].

La transformación de la semilla de cacao a un rico chocolate es un proceso complejo, en el intervienen muchas variables para su elaboración: la selección de las mejores semillas, el cuidado en la tostión y triturado, la mezcla con los mejores ingredientes, el refinado para que sea muy suave al paladar, el conchado para la adquisición de sabores y aromas, por último su moldeado

Para la fabricación de estos productos existe en la industria una línea o máquina de producción llamada Super cavemil 800 (Figura 1). Es una máquina desarrollada por CARLE & MONTANARI con capacidad de 10 a 25 moldes/min, entregando 2500 Kg/h. La línea está compuesta por un dispositivo giratorio para vaciado de los moldes, un precalentador de moldes, el depositador se compone de 7 inyectores que incorporan el chocolate en el molde, luego es llevado a una cinta transportadora vibratoria para dar forma y posterior a un túnel de enfriamiento para luego ser desmoldado y empacado



Figura 1. Super Cavemil 800, área de inyección de chocolate, imagen del autor

La zona de moldeo consta de una depositadora MLP600 la cual es encargada de inyectar el chocolate en el molde como se muestra en la figura 1.1. La línea posee una cinta vibratoria y rascador de los excesos de chocolate para pasar al túnel de enfriamiento y una estación de desmóldelo. La máquina de producción posee un compresor de refrigeración e incluye también juego completo de moldes y placas de inyección correspondientes para tabletas de 500 g. [3].

Antes de pasar por la dosificación los moldes ya deben de estar revisados de que todas sus partes se encuentren en estado óptimo para inyección.



Figura 1.1. Sistema de depositadora MLP600 Imagen del autor.

1.1 PLANTEAMIENTO DEL PROBLEMA

Dentro del proceso de producción y teniendo en cuenta que la línea súper cavemil 800 es una línea industrial en donde el proceso productivo no se detiene, siendo así los moldes donde se inyecta el producto tienden a sufrir un desgaste por su uso y se parten por las condiciones de la máquina no es posible que el operador de la misma pueda detectar esta pieza faltante en el momento que ocurre el evento.



Figura 1.2. Vista interna en cava, (sistema de enfriamiento) y solidificación del chocolate, Imagen del autor.

¿por qué se rompen los moldes?

Dentro de la cava, (Parte interna de la máquina donde se solidifica el chocolate) ocurre un proceso natural de desgaste, en donde los moldes tienden a sufrir fricción con las partes metálicas de la máquina y con la cadena de transporte de moldes.

Los moldes sufren también un cambio de temperaturas constante dentro del proceso, pasando por la inyección de un chocolate caliente a unos 30°C y posterior al ingreso a una zona de enfriamiento que oscila entre los 4°C y 7°C, lo que hace que el material del molde se debilite, así mismo un sistema neumático golpea el molde para que el chocolate ya solidificado se desprenda del molde para ser envuelto y empacado.

Este proceso hace que las betas que tienen el molde para su apoyo en el traslado por la línea de producción se partan y generen una posible contaminación al producto, lo que puede generar un evento de inocuidad pudiendo llegar al cliente final y generar un riesgo para la salud del consumidor y posibles reclamos y sanciones legales para la compañía. Todos los moldes para chocolate utilizados se realizan con un proceso de moldeo por inyección, utilizando única y exclusivamente materiales de primera elección certificados.

Dependiendo de las diferentes exigencias y del tipo de rendimiento requerido se deben esforzar moldes a diferentes desgastes, para el Super cavemil 800 los moldes están hechos de policarbonato estándar, estos son desarrollados por la empresa italiana Cabrellon, una empresa que maneja altos estándares de calidad en sus productos [5]. A pesar de esto y sabiendo que mecánicamente la línea de inyección es tan extensa, los materiales del molde no pueden ser desarrollados en materiales más rígidos o anti choque, ya que si por alguna falla mecánica la línea se bloquea o se estrella se debe retirar más fácilmente el molde y evitar que una parte mecánica de la línea se dañe o se desajuste. [6]



Figura 1.3. Moldes con fisuras dentro de la línea de producción, Imagen del autor.

Para cumplir con los programas de inocuidad y que la compañía pueda exportar sus productos, la empresa debe de disponer de dos personas para que realicen la revisión en la máquina con una frecuencia de revisión de 4 horas para revisar cada uno de los 530 moldes que la línea contiene.

Durante la operación normal de producción una persona está encargada de la revisión de los moldes de la línea una vez cada 4 horas, si esta detecta un molde con algún faltante se debe detener la línea inmediatamente, se debe activar un protocolo de búsqueda de la pieza faltante el cual pueden ser varias horas, lo que genera pérdidas millonarias por la no disponibilidad de la línea y todo el producto comprometido desde la última revisión hasta la hora del hallazgo queda rechazado.

1.2 JUSTIFICACION

Este trabajo se desarrolló en las instalaciones una empresa que proporcionará el ambiente propicio al ser una empresa productiva en donde su nivel de ventas las posiciona dentro de las principales empresas exportadoras y generadoras de empleo del país, dentro de su plan de producción abarca fabricación mensual de 2400 toneladas. El Super cavemil 800 es la línea más importante de la compañía, en el recae el 38% de la producción, una sola hora de parada genera la pérdida de 2500kg de producto. En un evento de inocuidad por molde roto las pérdidas pueden llegar a ser de más de 10 toneladas de producto, unos 77 millones de pesos en tan solo por la parada para la búsqueda de la pieza faltante.

Dentro del protocolo de inocuidad se debe detener la línea e iniciar la búsqueda de la pieza faltante del molde, se deben desmontar todos los moldes e iniciar a revisar toda la máquina parte por parte hasta verificarla completamente, si la pieza se encuentra el molde es desechado por el área de calidad como molde no apto para uso en línea y se retorna nuevamente el proceso, si por el contrario la pieza no se encuentra, se debe reprocesar todo el producto empacado desde la última revisión hasta el momento en que se detecta el faltante de molde, lo que incurre en pérdidas de producción y sobre costos en la mano de obra.

Para evitar esto se hace indispensable la inversión tecnológica y la adaptación de un sistema de monitoreo continuo, en donde los moldes que hacen parte de la línea sean monitoreados antes del contacto con el producto. Para ello se propone un sistema de visualización basado

en el procesamiento de imágenes que genere una alerta al operador de la línea y esta pueda ser detenida. Como la monitorización de los moldes será constante se reducirá el riesgo de retener grandes cantidades de producto y será mucho más fácil encontrar la pieza faltante que se debió desprender segundos antes.

1.3 OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar e implementar un sistema de reconocimiento de piezas faltantes en moldes para la industria de alimentos, mediante el procesamiento de imágenes que permita la monitorización de los moldes que componen la máquina de producción y pueda alertar al operador de la máquina la presencia de un molde roto.

1.3.2. OBJETIVOS ESPECÍFICOS

Analizar las variables (Velocidad de recorrido del molde, área del molde, textura del molde, color del molde) que intervienen en el proceso de fabricación de chocolate para estimar las características de resolución necesarias en la toma de imágenes.

Diseñar por medio de Matlab y con el uso de una base de datos de imágenes con anomalías generadas por eventos de inocuidad con moldes rotos, junto con imágenes de moldes sin anomalías para contrastarlas mediante el procesamiento de imágenes, a través de un algoritmo que permita procesar la captura imágenes para compararla por medio de contrastes detectado así las anomalías presentadas en los moldes.

Implementar un sistema de operación continua que permita la captura de hasta 10 moldes por minuto para la detección y alerta de la presencia de un molde con anomalía.

Evaluar la funcionalidad del sistema por medio del número moldes por minuto que el programa es capaz de procesar y realizar un análisis estadístico para comprobar la exactitud de los moldes rotos detectados por el sistema en función del tiempo.

Establecer un protocolo de detención de la maquina al momento de detectar un molde roto.

1.4 ALCANCE.

El alcance de este proyecto está limitado al presupuesto en la adquisición de los equipos y resoluciones de cámaras que se utilizaron, así mismo el proyecto se implementó en las instalaciones de una compañía chocolatera y no se tendrá apoyo económico de ella, dado que se busca comercializar el producto por parte del proponente. El proyecto se desarrollará en la plataforma Matlab sin embargo una vez se concluya el proyecto y se vea la viabilidad comercial se podrá migrar a otra plataforma de desarrollo.

2. MARCO TEORICO

Podemos entender que el concepto de imagen según varios criterios, como la etimología, la fotografía y el arte. En este caso, la definición simple es que la imagen es una representación visual del objeto [6].

La fotografía es un arte o técnica que captura una imagen permanente cuando se expone a la luz. Para tomar una fotografía, necesita un medio químico (imagen analógica) y un medio electrónico que pueda convertir la luz en una señal electrónica (imagen digital). Independientemente de cómo se tomen las imágenes, el procesamiento asistido por computadora necesita digitalizar estas imágenes. Durante el proceso de digitalización de una foto, los dispositivos electrónicos realizan la discretización de la imagen. Es decir, el dispositivo electrónico divide la imagen en unidades de información más pequeñas. Cada unidad almacena información sobre la intensidad de cada uno de los tres colores primarios. Cada unidad de información se denomina píxel. Los sistemas electrónicos digitales, como las computadoras, requieren un proceso de discretización porque codifican información en un estado discreto que solo puede tomar dos valores, 0 o 1. De esta forma, una imagen digital puede interpretarse como una matriz en la que cada celda almacena un número (foto en escala de grises, izquierda en la Figura 2), o un conjunto de tres tablas en las que cada celda

almacena un valor como su propio número. De los tres colores principales en ese punto en particular (foto a color, a la derecha en la Figura 2) [6]

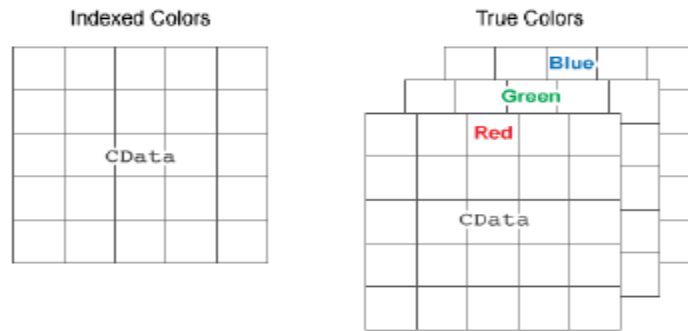


Figura 2. Ejemplo de imagen digital - Fuente: bibliografía [7]

2.1 IMÁGENES EN ESCALA DE GRISES

Una imagen en escala de grises, también conocida como imagen de intensidad, es un arreglo de datos que representa una identificación donde sus valores son la representación de un rango dentro de la matriz de la imagen. Las imágenes en escala de grises se representan como una única matriz, en el que cada elemento de la matriz corresponde a un píxel de la imagen. Las imágenes en escala de grises generalmente no se almacenan en la tabla de colores, En consecuencia, las imágenes en escala de grises se consideran imágenes indexadas. [8]

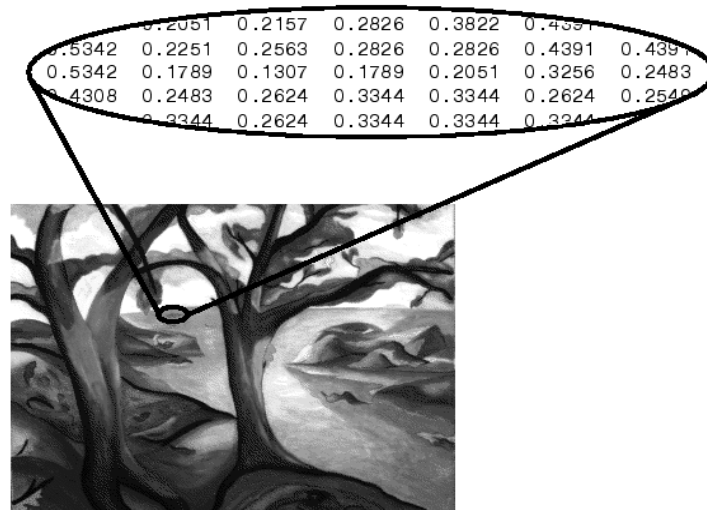


Figura 2.1 - Imágenes en escala de grises - Fuente: bibliografía [8]

2.2 IMÁGENES RGB

Las imágenes RGB, también conocidas como imágenes de color verdadero, se almacenan como una matriz de datos $m \times n \times 3$ que define los componentes de color rojo, verde y azul de los píxeles individuales. Las imágenes RGB no utilizan paletas. El color de cada píxel está determinado por la combinación de intensidades de rojo, verde y azul almacenadas en cada plano de color en la ubicación del píxel. El formato de archivo gráfico guarda imágenes RGB como imágenes de 24 bits. En esta imagen, los componentes rojo, verde y azul tienen 8 bits cada uno. Esto ofrece 16 millones de posibilidades de color. La razón por la que se denomina "imagen en color verdadero" es que puede representar con precisión una imagen real. [8]

Las matrices RGB de MATLAB pueden ser de clase double, uint8 o uint16. En una matriz RGB de clase double, cada componente de color tiene un valor entre 0 y 1. Los píxeles con un componente de color de (0,0,0) se muestran en negro y los píxeles con un componente de color de (1,1,1) se muestran en blanco. Los tres componentes de color de cada píxel se almacenan en tres dimensiones de la matriz de datos. Por ejemplo, los componentes de color rojo, verde y azul de un píxel (10,5) se almacenan en RGB (10,5,1), RGB (10,5,2) y RGB (10,5,3) para cada caso. [7] [8]

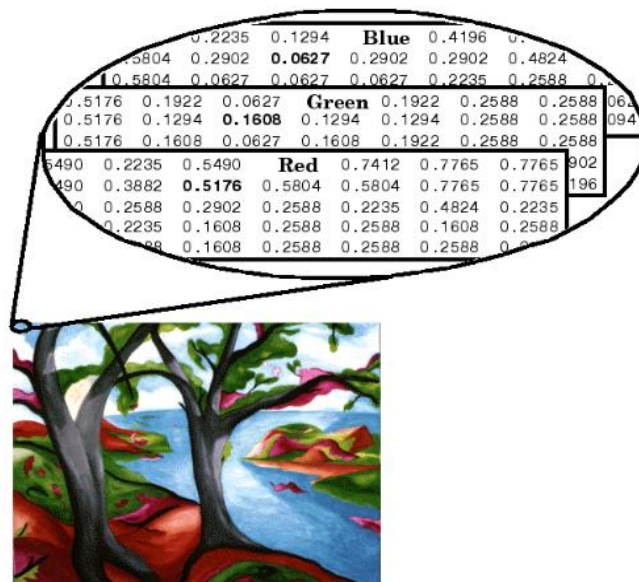


Figura 2.2 - Imagen RGB - Fuente: bibliografía [8]

2.3 CONVOLUCIÓN

Uno de los procesos más distintivos de estas redes son las convoluciones. El cual consiste en tomar un grupo de píxeles de la imagen de entrada e ir realizando un producto escalar con un kernel. El kernel en las redes convolucionales se considera como el filtro que se aplica a una imagen para extraer ciertas características importantes o patrones de esta, Entre las características importantes para lo que sirve el kernel son detectar bordes, enfoque, desenfoque, entre otros. Esto se logra al realizar la convolución entre la imagen y el kernel El kernel recorrerá todas las neuronas de entrada y obtendremos una nueva matriz. En el caso de que la imagen sea de color se tendrán 3 kernels del mismo tamaño que se sumarán para obtener una imagen de salida. [9]

Por ejemplo, si tiene dos matrices de 3 por 3 (primer núcleo, segundo segmento de imagen), la convolución es el proceso de invertir las filas y columnas del núcleo, multiplicarlas como entradas locales y al sumarlas, elemento en las coordenadas es [2, 2] en la imagen resultante (es decir, el elemento central) es una combinación ponderada de todas las entradas en la matriz de imágenes la convolución se define como la función $f(x)$ y $g(x)$ de la variable continua X , la convolución es:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x - \tau) d\tau$$

Donde $*$ representa la convolución y significa una multiplicación normal. Para la función de la variable discreta X , es decir, matriz por matriz, se define de la siguiente manera:

$$f[x] * g[x] = \sum_{k=-\infty}^{\infty} f[k] \cdot g[x - k]$$



Figura 2.3 – convolución de imágenes - Fuente: bibliografía [9]

2.4 CONVOLUCIÓN CON IMÁGENES EN ESCALA DE GRISES.

Para este proceso tenemos una imagen y el kernel, con la finalidad que el filtro o kernel recorra toda la imagen (pixel). Por lo general, el kernel es de menor tamaño que la imagen. La convolución permite multiplicar el kernel con la porción de imagen escogida, se realiza la multiplicación tal como indica la imagen y luego el kernel se va desplazando por esta razón es un proceso iterativo. [10]

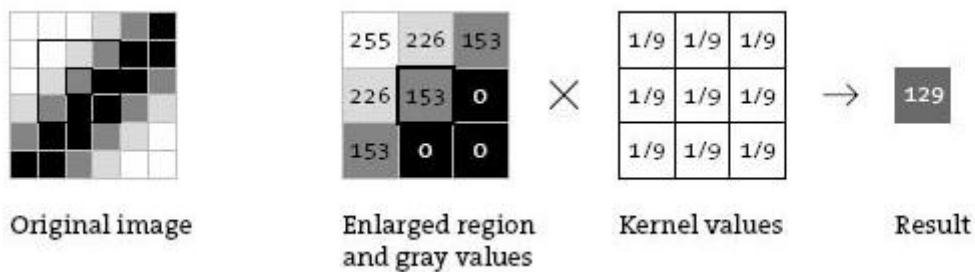


Figura 2.4 – convolución de imágenes a escala de grises - Fuente: bibliografía [10]

2.5 CONVOLUCIÓN CON IMÁGENES RGB.

Las imágenes de color se expresan en 3 planos R(red), G(Green), B(blue) y al combinarse se ven todas las mezclas de colores.

Por lo que ahora el filtro también será de 3 planos, la convolución suele ser similar al de escala de grises, pero ahora el filtro de 3 capas le corresponde a un pixel(capa) correspondiente. Aunque la imagen de entrada es de 3 planos, la imagen resultante será de 1 plano, (es decir una imagen a color), se hace el proceso iterativo mencionado anteriormente, hasta completar todo. [11]

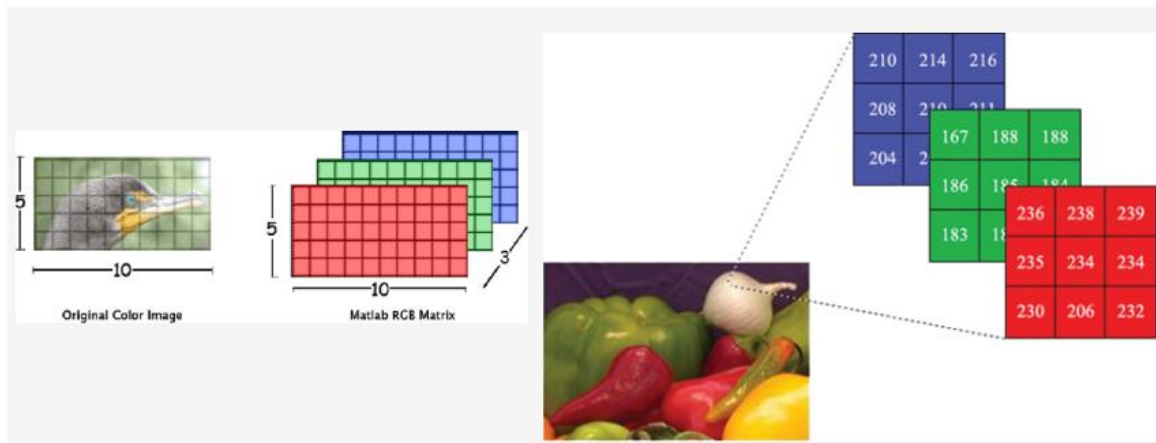


Figura 2.5 – convolución de imágenes RGB - Fuente: bibliografía [11]

El procesamiento digital de imágenes puede definirse como la operación sobre imágenes mediante el uso de ordenadores con el objetivo de imitar, hasta cierto punto, el proceso de visión biológico.

Un sistema de visión y procesamiento de imágenes se compone de una serie de subsistemas que operan sobre una imagen con el objetivo de interpretar alguna característica notable.

Hoy en día, prácticamente no existe ninguna área de la vida cotidiana en la que no se apliquen de una forma u otra algún tipo de tratamiento digital de imágenes. Desde procesos automatizados a gran escala hasta una simple publicación en una red social. El tratamiento digital de imágenes forma parte de la vida actual en mayor o menor medida.

Para cualquier proceso industrial es fundamental detectar las fallas que pueda tener un elemento de una cadena productiva, tanto de los elementos producidos como de los elementos productores, con el fin de minimizar las pérdidas, tanto económicas como logísticas. [11]

2.6 REDES CONVOLUCIONALES

Una red neuronal convolucional (CNN o ConvNet) es una arquitectura de red de aprendizaje profundo que aprende directamente de los datos sin necesidad de extraer funciones manualmente.

Estas redes son especialmente útiles para encontrar patrones en imágenes para reconocer objetos, rostros y escenas. También es útil para clasificar datos que no son imágenes, como audio, series temporales y señales. Las aplicaciones que utilizan el reconocimiento de objetos y la visión artificial, como las aplicaciones de vehículos autónomos y el reconocimiento facial, dependen en gran medida de las CNN. El uso de CNN con el aprendizaje profundo es popular debido a tres factores clave. CNN:

- Aprenden características directamente sin necesidad de extraerlas manualmente.
- Produce resultados de reconocimiento muy precisos.
- Se pueden volver a entrenar para nuevas tareas de reconocimiento, lo que permite aprovechar las redes preexistentes. [12]

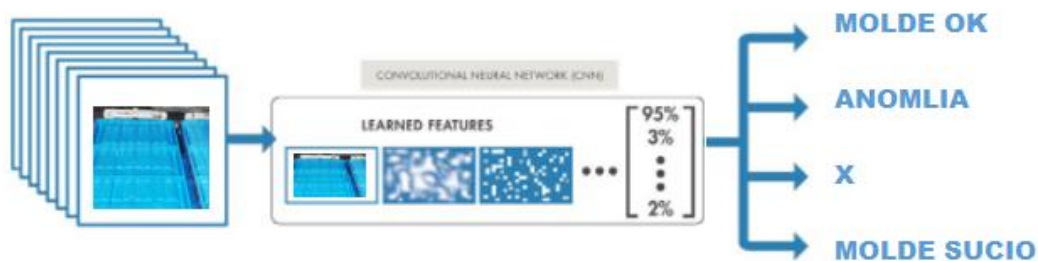


Figura 2.6 - Flujo de trabajo de deep learning. Las imágenes se envían a CNN - Fuente: bibliografía [12]

Las redes convolucionales proporcionan una construcción óptima para exteriorizar y labrar características principales en imágenes y datos de series temporales como se muestra en la figura 2.6. Las redes convolucionales son una tecnología top en aplicaciones tales como: Imágenes médicas, Procesamiento de audio, procesamiento de patrones.

Una red neuronal convolucional puede orientar de decenas o cientos de capas que aprenden a detectar diferentes características de una plancha. Se aplican filtros a cada plancha de aprendizaje con distintas resoluciones, y la posibilidad de cada plancha convolucionada se

emplea como puertas para la su ulterior capa. Los filtros pueden cambiar desde características muy simples, tales como la claridad y los bordes, hasta más complejas, como las características que definen la finalidad de forma única. [12]

2.7 APRENDIZAJE DE CARACTERÍSTICAS, CAPAS Y CLASIFICACIÓN

Al igual que otras redes neuronales, una red convolucional está compuesta por una capa de entrada, una capa de salida y muchas capas intermedias ocultas.

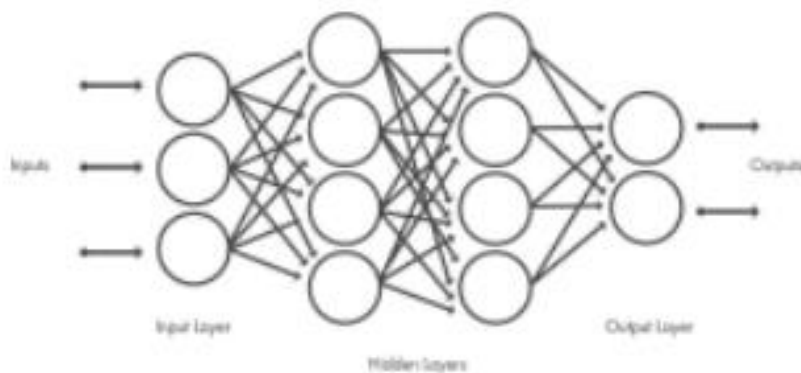


Figura 2.7 capas y clasificación - Fuente: bibliografía [12]

Estas capas realizan operaciones en los datos para aprender ciertas propiedades de los datos.

- **Convolución:** Se aplican una serie de filtros de convolución a la imagen de entrada. Cada filtro habilita ciertas propiedades de la imagen.
- **Agrupación:** simplifica la salida al reducir la tasa de muestreo no lineal y reducir la cantidad de parámetros que la red necesita aprender.

Estas operaciones se repiten en decenas o cientos de niveles, por lo que cada nivel aprende a identificar diferentes características. [13]. [14]

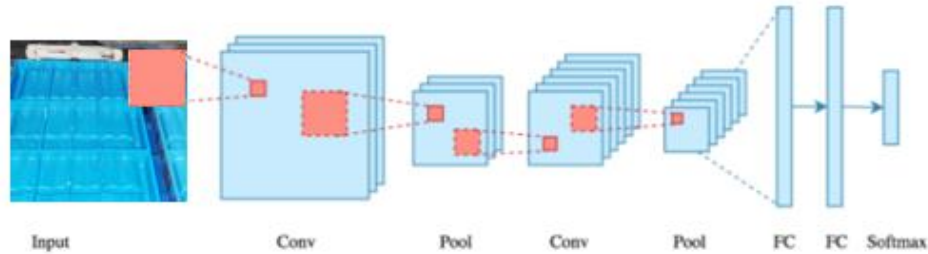


Figura 2.8 Capas y clasificación - Fuente: bibliografía [13]

2.8 VALORES DE PESO Y SESGO COMPARTIDOS

Al igual que las redes neuronales tradicionales, las redes convolucionales están formadas por neuronas con pesos y sesgos. El modelo aprende estos valores durante el proceso de entrenamiento y los actualiza continuamente con cada nuevo ejemplo de entrenamiento. Sin embargo, para las redes convolucionales los valores de ponderación y sesgo son los mismos para todas las neuronas ocultas en una capa en particular.

Esto significa que todas las neuronas ocultas en diferentes áreas de la imagen reconocerán las mismas características, como bordes y rayas. Esto hace que la red sea resistente a la transformación de objetos en la imagen. Por ejemplo, una red entrenada para reconocer un automóvil puede hacerlo dondequiera que esté el automóvil en el cuadro.

Capa de clasificación

Después de aprender las características en diferentes niveles, la arquitectura red pasa a la clasificación. La penúltima capa está completamente conectada y genera un vector de dimensión K . Donde K es el número de clases que la red puede predecir. Este vector contiene las probabilidades de cada clase de imágenes clasificadas. La capa final de la arquitectura CNN

utiliza una capa de clasificación como Softmax para proporcionar resultados de clasificación. [14]

2.9 CLASIFICACIÓN DE MOLDES UTILIZANDO LA RED NEURONAL CONVOLUCIONAL DE GOOGLNET (CNN)

GoogLeNet es una red neuronal convolucional que tiene 144 capas de profundidad. Estas redes han aprendido diferentes representaciones de características para una amplia gama de imágenes, este tipo de redes pre entrenadas tienen un tamaño de entrada de imagen de 224 por 224.

La red neuronal convolucional es un tipo de red neuronal profunda utilizada para la clasificación de imágenes. Este trabajo muestra un enfoque en la clasificación de moldes utilizando la red neuronal convolucional de GoogLeNet a través del aprendizaje por transferencia. [15].

El desarrollo más adelante es muestra tal vez la forma más sencilla posible de entrenar una red neuronal convolucional para clasificar imágenes. Comienza con la división del conjunto de datos en conjuntos de datos de entrenamiento y validación. La proporción del conjunto de datos de entrenamiento y validación es 70:30. Luego se redimensionan de acuerdo con el tamaño de entrada de GoogLeNet. Después de eso, se modifican la capa de aprendizaje de características y la capa de clasificador de salida. Finalmente, la capa de aprendizaje de características modificada se vuelve a entrenar con el conjunto de datos

La clasificación de moldes utilizando la red neuronal convolucional de GoogLeNet tiene un enfoque muy sencillo y exacto. En consecuencia, este es el propósito del aprendizaje por transferencia. Preferimos transferir el aprendizaje siempre que sea posible para una implementación más rápida en donde no se emplee un código complejo o robusto que presente trabas en la compilación del mismo. La utilización de esta red se divide en 7 pasos fundamentales.

1. Cargar el conjunto de imágenes,
2. Dividir el conjunto de imágenes en un conjunto de datos de entrenamiento y validación con una proporción de 70:30,
3. Cambiar el tamaño del conjunto de imágenes en base a el tamaño de la capa de entrada de la red neuronal convolucional.
4. Validar la red e identifique el aprendiz de características y la capa de clasificación.

5. Modificar la capa de aprendizaje de características y la capa que contiene la clasificación,
6. Definir opciones de formación deseada
7. Entrenar la red

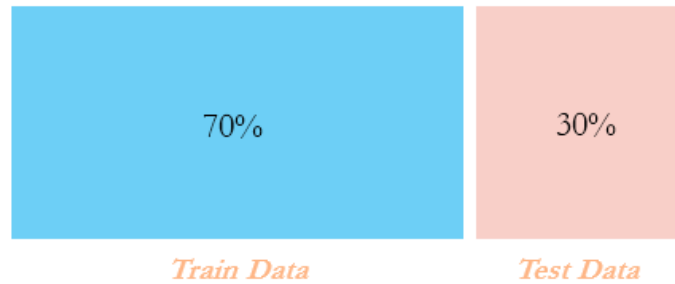


Figura 2.9 – Clasificación usando google net - Fuente: bibliografía [15]

3. ESTADO DEL ARTE

El desarrollo de métodos automatizados e interactivos para la manipulación de imágenes multidimensionales, la implementación de herramientas de medición para evaluación cuantitativa de las imágenes y el diseño y la validación de modelos que mejoren la interpretación y la toma de decisiones en las aplicaciones que desde la academia puedan ser aplicados en la industria. Este modelo ha sido utilizado en otro tipo de proyectos asociados al procesamiento de imágenes como trabajos de grado para el análisis de paneles solares, en donde busca encontrar las fallas superficiales por el uso y exposición al sol, el cual tiene como objetivo detectar contaminación en los paneles debido a la exposición con el ambiente que redujeran el rendimiento del panel [16]. El procesamiento de imágenes para la clasificación de café verde, el cual tiene como objetivo aumentar la calidad en la compra de café por la asociación de cafeteros de Colombia [17], el diseño de un sistema por visión artificial para determinar la calidad de mandarinas [18] o el sistema de detección de caracteres de placas de automóviles mediante técnicas de visión artificial [19].

3.1 NORMA HACCP

¿Qué es la HACCP?

HACCP significa Hazard Analysis and Critical Control Point (Análisis de Peligros y Puntos Críticos de Control) es un sistema de misión en el que se aborda la seguridad alimentaria a través del análisis y control de peligros químicos, físicos y biológicos en la producción de materia prima, así como en la adquisición y manipulación, hasta la fabricación, distribución y consumo del producto en el cliente final.

El Análisis de Peligros y Puntos Críticos de Control es una de las normas internacionales que se basan en la producción segura de alimentos con un acercamiento preventivo.

Dichos sistemas de gestión HACCP son un requisito legal en Estados Unidos y varios países europeos para todos los niveles de la cadena alimentaria. [20].

El HACCP se desarrolla en 7 principios:

- Conducir un análisis de peligros
- Identificar puntos críticos de control
- Establecer un máximo o mínimo de revisión para los puntos críticos de control como: moldes, imanes, filtros y temperaturas
- Establecer límites críticos
- Establecer procedimientos de seguimiento
- Establecer acciones correctivas
- Establecer procedimientos de mantenimiento de registros

3.2 INVESTIGACION

Para el desarrollo del presente trabajo fue necesario realizar una validación año tras año sobre la ocurrencia de los eventos e incidentes de inocuidad relacionado con moldes rotos en la línea súper cavemil reflejados en el gráfico 1.



Grafico 1. Numero de eventos registrados de moldes rotos según años, Fuente eventos de inocuidad casa luker.

4. METODOLOGIA

El primer objetivo tiene como propósito abarcar la monitorización de moldes utilizados en la inyección de chocolates de 500g y así poder cuantificar para realizar el análisis de las variables que intervienen en el proceso, siendo así el sistema propuesto estará en la capacidad de analizar los moldes para las referencias de pastillados

molde	referencia
p16	Molde referencia pastillaje 16
p20	Molde referencia pastillaje 20

Tabla 1. Tabla de referencias utilizadas en produccion

La ubicación de la cámara de monitorización se ubicará en la zona antes de inyección, esto para garantizar que los moldes que se dosifiquen con chocolate ya estén validados.

El sistema de dosificación del Super cavemil varía según la referencia que estamos trabajando, es decir según el tipo de sabor del chocolate esta velocidad varia. Para ello la maquina trabaja en su máxima capacidad a 750 RPM pasando 25 moldes por minuto.

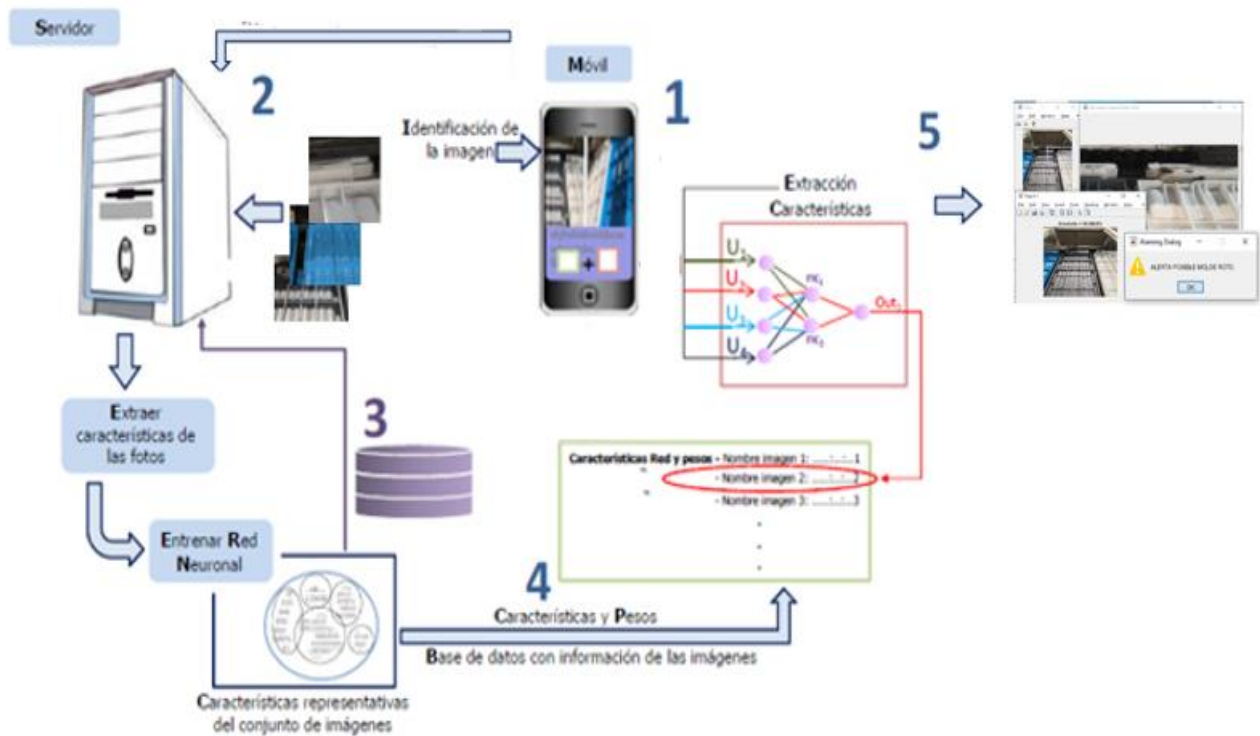


Figura 4 – Diagrama de bloques de la metodología a utilizar, Fuente del autor.

4.1 ANALIS DEL MOLDE

Para la fabricación de chocolate existe en la industria infinidad de estilos y diseños, para el desarrollo de este producto se emplearán los moldes de patillaje de 16 unidades de chocolate de meza en barra, como se observa en la figura 4.1 El molde posee un diseño especial según la línea de producción en la que se va a utilizar, en este caso podemos observar las líneas verdes las cuales son las partes sensibles del molde y donde se tiene evidencia de mayor ruptura tal como se observa en la imagen 4.2.

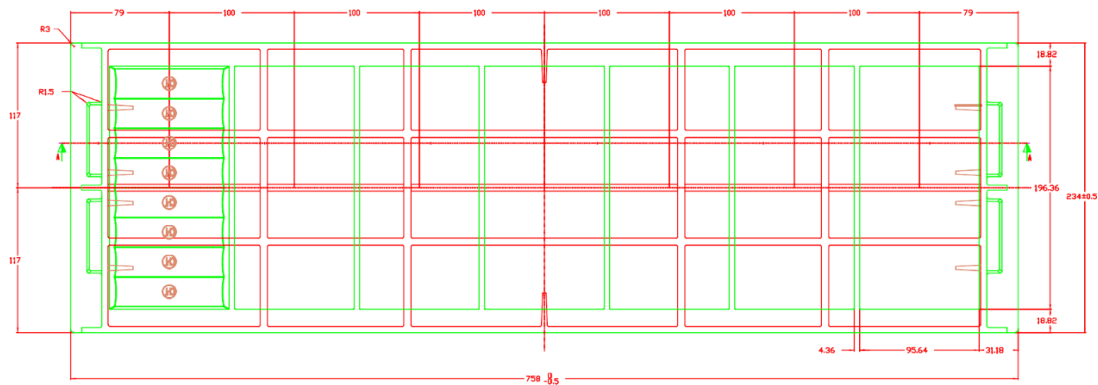


Figura 4.1 – Vista superior- Fuente del autor

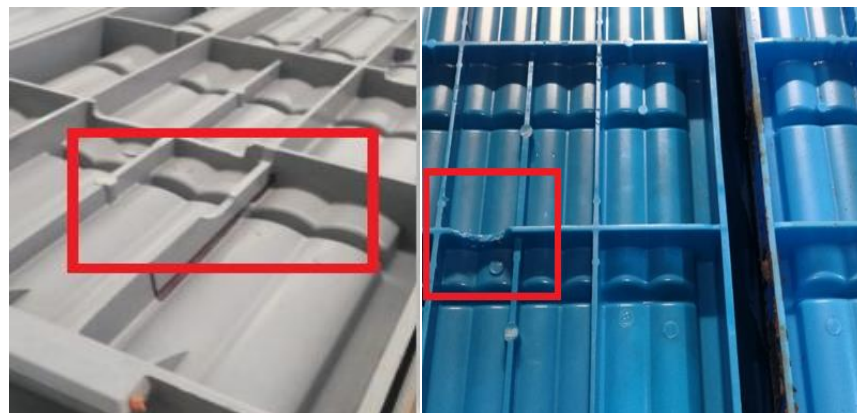


Figura 4.2 – Betas de ruptura- Fuente del autor

En figura 4.3 observamos la cara lateral del molde, vemos las cavidades de inyección el cual es el punto de menor riesgo de ruptura puesto que el de mayor resistencia debido que es donde se le da forma al chocolate, pero es un punto crítico ya que es allí donde puede caer una pieza de molde y pueda solidificarse dentro del producto como se muestra en la imagen 4.4.

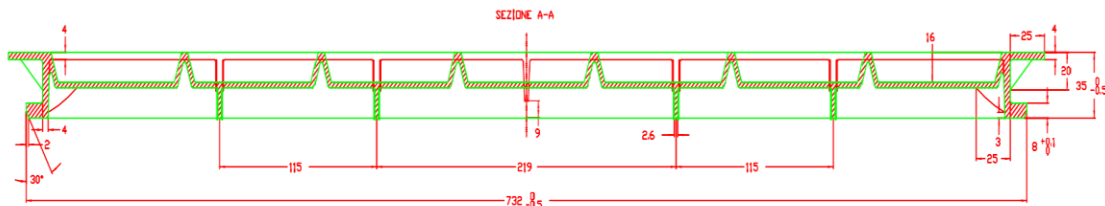


Figura 4.3 – Vista posterior- Fuente del autor



Figura 4.4 Producto contaminado con pieza de molde- Fuente del autor

En figura 4.5 vemos la cara lateral derecha del molde, esta representa el punto de sujeción del molde en la cadena de traslado durante el recorrido por la línea, esta pestaña está identificada como punto crítico, puesto que es la de mayor ruptura al someterse al sobre esfuerzo el sistema de desmolde tal como se observa en la figura 4.6.

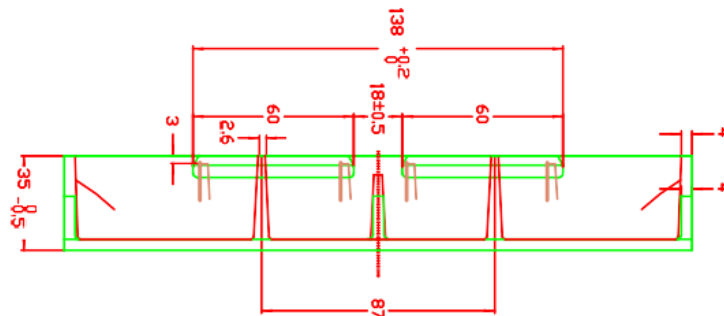


Figura 4.5 – Vista lateral derecha- Fuente del autor



Figura 4.6 Ruptura de molde en área de sujeción- Fuente del autor

4.2 TOMA DE IMÁGENES

Una vez conocidas las variables anteriores de cuáles son los puntos de mayor riesgo e incidencia de ruptura en el molde, se adecuó el sistema para la toma de imágenes en donde se utilizó la cámara teleobjetivo de 18 megapíxeles 4K del Smartphone Samsung Galaxy note 20 Ultra, la cual permitió una excelente resolución bajo el entorno de adquisición utilizado. Puesto que se han analizado imágenes de moldes de diferentes colores, se decidió separarlas en diferentes lotes según la construcción del molde tal como se muestra en la tabla 2.

Tal y como se ha mencionado anteriormente, el algoritmo desarrollado se ha probado con 2 tipos de imágenes completamente diferentes. Los 2 tipos de imágenes se clasificaron en imágenes como imágenes normales e imágenes con anomalías para las siguientes referencias de moldes a utilizar como se muestra en la tabla 3.

Tipo de molde	Cantidad	Clasificación	Porcentaje dentro de la base de datos
Moldes Azules	347	Normal	58%
Moldes Grises	48	Normal	8%
Moldes blancos	98	Normal	16%
Moldes Mixtos	104	Normal	17%
Moldes Azules	263	Anomalia	58%
Moldes Grises	36	Anomalia	8%
Moldes blancos	74	Anomalia	16%
Moldes Mixtos	79	Anomalia	17%

Tabla 2. Clasificación de imágenes dentro de la base de datos

Tipo de Imágenes	Resolución de las imágenes	total imágenes
Con Anomalías	96 ppp	453
Normales	96 ppp	517

Tabla 3. Resolución de imágenes en base de datos

La cámara empleada está ubicada la zona de retorno de moldes justo antes de entrar a la dosificadora puesto que es el único sitio en donde se puede ubicar la cámara exitosamente, siendo así garantizada que los moldes a inyectar estén validados por el sistema.

Para la adecuación del sistema de captura de imágenes propuesto en la imagen 4.8. se ubicó un trípode a una altura de 17 cm de la base de la cadena de retorno de moldes a la parte externa de la cabina, ubicando la cámara en un ángulo de 12° para que no tomara una foto de frente si no en un ángulo que permitiese mirar de manera lateral el molde y así capturar mejor las betas que lo componen, con este ángulo de inclinación podemos capturar el 65% del molde, para lo que se propone más adelante la implementación de dos cámaras que permitan el censo del 100% del molde como se muestra en la imagen 4.7.

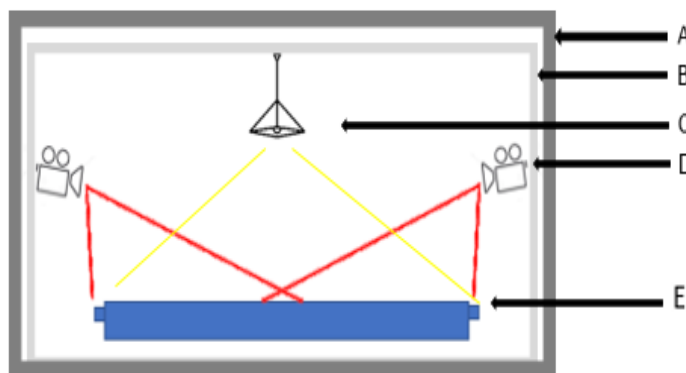


Figura 4.7 Figura de representación del sistema de captura de imágenes. Fuente del autor

- A: Cabina
- B: Soporte para la fijación de cámaras e iluminación
- C: Iluminación
- D: Cámara
- E: Molde



Figura 4.8. Posición de montaje de cámaras e laminación en línea de producción. Fuente del autor

Para la captura de imágenes se utilizó la aplicación IP CAM y por medio de la Matlab se implementó el siguiente código para la captura de imágenes, este permite capturar la imagen del molde en 3 segundos.

10 moldes por minuto = 60 segundos

1 captura es igual a 3 segundos

60 segundos de inyección/3 segundos de captura = 20 moldes.

Se tomó la velocidad mínima de 300 RPM o 10 moldes por minuto hasta las 750 RPM o 25 moldes se logró capturar la imagen del molde en 3 segundos, es decir el algoritmo de captura está en la capacidad de leer 20 moldes por minuto, siendo así, cuando la línea de producción está trabajando a máxima velocidad de 750 RPM el sistema cumple con la validación del 80% de los moldes que inyecta la línea en un minuto, cabe resaltar que la operación normal de la línea es de entre 700 RPM y 730 RPM es decir entre 20 y 24 moldes por minuto, lo que amplía el porcentaje de validación de entre el 80% y 99% de los moldes que son inyectados en la línea de producción.

```
cam = ipcam ('http://192.168.39.113:8080/videoFrame')
preview(cam)
cam = webcam();
% Captura una imagen para calcular el tamaño
videoFrame = snapshot(cam);
frameSize = size(videoFrame);
% Creamos un objeto para mostrar la imagen en un video
videoPlayer = vision.VideoPlayer('Position',[100 100 [frameSize(2),frameSize(1)]+30]);
for i = 1:100
    videoFrame = snapshot(cam); % Toma las fotos
    % step(videoPlayer,videoFrame); % Muestra el video
    mkdir('fotos'); % Crea la carpeta
    imwrite(videoFrame,['fotos\'','captura.jpg']); % Se guarda en la carpeta fotos con la extensión .jpg
end
```

Código de captura de imágenes



Figura 4.9 Imágenes capturadas mediante el sistema propuesto. Fuente del autor

Durante la fase de adquisición de imágenes para la base de datos se debieron descartar las imágenes que no cumplieran con las condiciones de luz, posición o de resolución, así mismo se realizaron diferentes tipos de filtrado tratando de buscar una mejor imagen que permitiese una mejor precisión del algoritmo a la hora de analizar la imagen, como se muestra en la imagen 4.9.1. Aunque existen varios métodos de filtrados de imagen, para este caso no es viable ya que al algoritmo de la red neuronal google net, realiza una comparación de la imagen de entrada vs las imágenes de la base de datos que le permitan diferenciar las características, si cambiamos las imágenes en otros escenarios, la probabilidad de acierto disminuye.



Figura 4.9.1 Imágenes descartadas durante la creación de la base de datos. Fuente del autor

4.3 PROCESAMIENTO DEL ALGORITMO

Para la implementación del algoritmo lo primero que se realizó fue la construcción de una base de datos llamada Dataset que contienen dos carpetas, una para imágenes normales y otra con imágenes con anomalías como se muestra en la parte izquierda de la imagen 4.9.2.

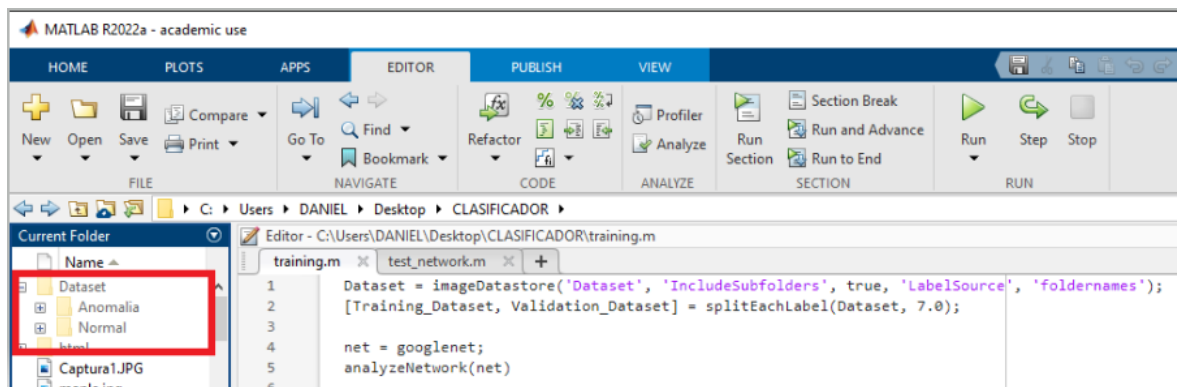


Figura 4.9.2 Base de datos para las imágenes. Fuente del autor

Ahora en Matlab se procedió a instalar el toolbox que contiene google net.

Una vez instalado se crea una variable que llame el conjunto de imágenes guardadas, el primer argumento de esta función es la ruta de las imágenes guardadas que queremos cargar y que incluya las subcarpetas que necesitamos para etiquetar la categoría de datos, los siguiente que se realizo fue dividir el conjunto de datos en conjunto de datos para entrenamiento y validación, la etiqueta divide el los datos en proporción 70/30 es decir que el 70% de las imágenes se usaran para entrenamiento y el 30% para validación.

```
Dataset = imageDatastore('Dataset', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
[Training_Dataset, Validation_Dataset] = splitEachLabel(Dataset, 7.0);
```

Luego cargamos la red de google y la guardamos en un variable net que vamos a entrenar en google net, entonces necesitamos saber la arquitectura de esta red y para ello utilizamos la variable analyzeNetwork(net).

```
net = googlenet;
analyzeNetwork(net)
```

Ejecutando esta parte del código obtenemos:

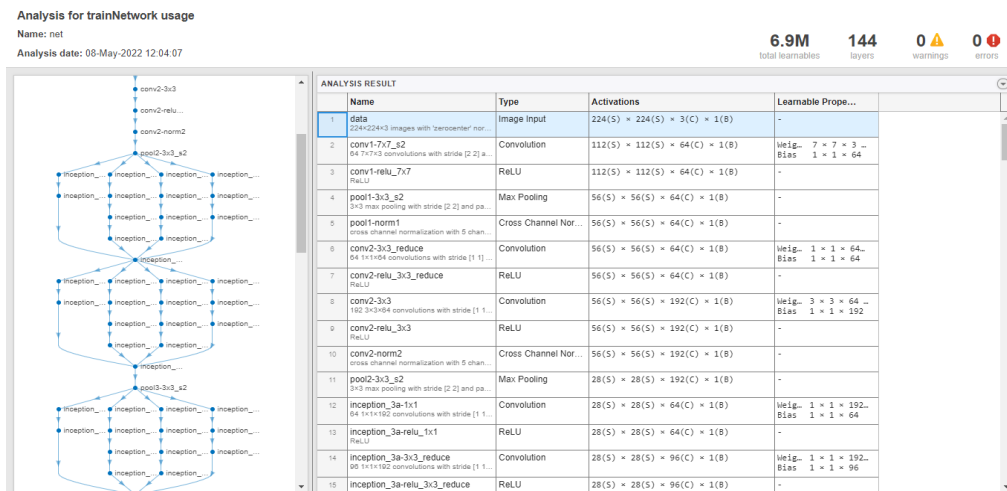


Figura 4.9.3 Arquitectura de la red neuronal durante el primer procesamiento. Fuente del autor

En la imagen observamos la arquitectura de google y podemos ver en la parte superior que tiene 144 capas que son los datos que procesa la red neuronal, por lo que es una red bastante robusta, así mismo observamos que en la primera capa que esta es la capa de entrada y que este el tamaño de las imágenes debe ser 224 x 224 pixeles de imágenes de tres canales.

Como las imágenes que tomamos pueden ser de diferentes tamaños lo que nos dificultaría la entrada de la primera capa de la red neuronal, modificamos el tamaño de los datos a la entrada de la red para que nos acepte diferentes tamaños de entrada, creamos una variable de entrenamiento redimensionada y usamos una variable de almacenamiento de datos para la imagen aumentada, haciendo lo mismo para los datos de validación.

```
Input_Layer_Size = net.Layers(1).InputSize(1:2);  
Resized_Training_Image = augmentedImageDatastore(Input_Layer_Size, Training_Dataset);  
Resized_Validation_Image = augmentedImageDatastore(Input_Layer_Size, Validation_Dataset);
```

Para clasificar las imágenes de nuestro conjunto de datos, la arquitectura de la red neuronal google net nos dice que la capa de aprendizaje de características es la 142, como modificamos la capa de entrada que queremos que esta red aprenda de nuevo esta característica por lo que realizamos una modificación a esta capa, google net está diseñado para que pueda clasificar hasta 1000 tipos de imágenes de diferentes cosas, y esta clasificación la realiza en la capa 144 por lo que también debemos modificar esta capa también.

```
Feature_Learner = net.Layers(142);  
Output_Classifier = net.Layers(144);
```

Ahora necesitamos averiguar la cantidad de clases que tenemos en nuestro conjunto de datos de almacenamiento, para ello en primer lugar la red debe obtener las etiquetas del conjunto de datos de entrenamiento por lo que usamos categorías para diferenciarlos, finalmente usamos la función numérica para encontrar la cantidad de categorías y las asignamos a la siguiente variable.

```
Number_of_Classes = numel(categories(Training_Dataset.Labels));
```

Para modificar la capa de aprendizaje de características creamos una nueva variable llamada función de aprendizaje y usamos la capa conectada para modificar la función de aprendizaje,

el primer argumento es el número de clases, el segundo argumento es el nombre y el valor que vamos a llamar que en este caso lo nombramos como función de molde, el siguiente argumento es ponderar el valor de la tasa de aprendizaje que establecimos en 10, el último factor que llamamos es el porcentaje de aprendizaje y sesgo que también pusimos en 10.

```
New_Feature_Learner = fullyConnectedLayer(Number_of_Classes, ...
    'Name', 'Molde Feature Learner', ...
    'WeightLearnRateFactor', 10, ...
    'BiasLearnRateFactor', 10);
```

Una vez definida la capa de aprendizaje sin características modificamos la capa de clasificación, tomamos una nueva variable que llamamos capa sin clasificar y usamos la función de clasificador de capa que llamamos clasificador de molde.

```
New_Classifier_Layer = classificationLayer('Name', 'Molde Classifier');
```

Para modificar la arquitectura de la red es necesario obtener el grafico de capa completo para lo que creamos la siguiente variable.

```
Layer_Graph = layerGraph(net);

New_Layer_Graph = replaceLayer(Layer_Graph, Feature_Learner.Name, New_Feature_Learner);
New_Layer_Graph = replaceLayer(New_Layer_Graph, Output_Classifier.Name, New_Classifier_Layer);

analyzeNetwork(New_Layer_Graph)
```

Ejecutando este comando obtenemos una red modificada, con una nueva capa de aprendizaje con nuevas funciones llamadas función de aprendizaje de molde y una nueva capa llamada clasificadora de molde, como lo podemos ver en la figura 4.9.4

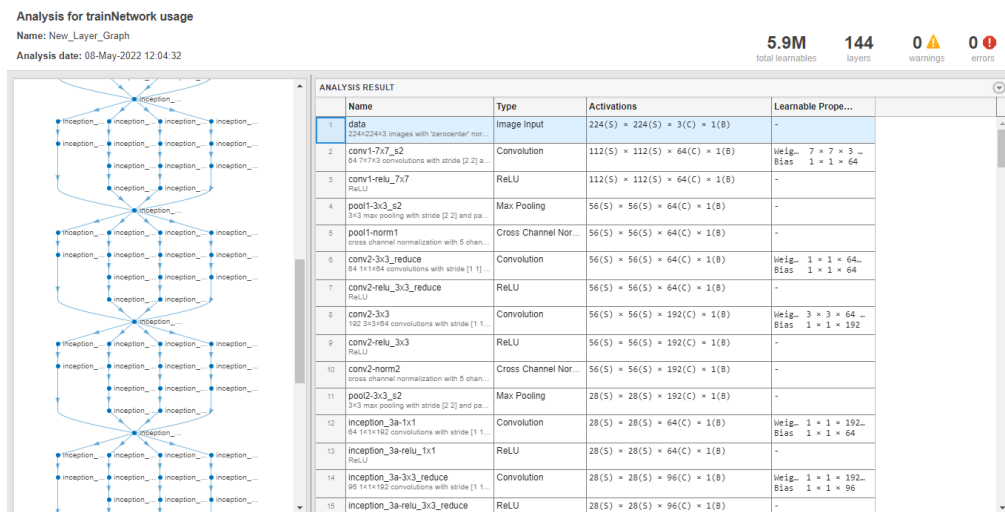


Figura 4.9.4 Arquitectura de la red neuronal durante el segundo procesamiento. Fuente del autor

Ahora para entrenar la red usamos un mini lote de tamaño 5, también calculamos la frecuencia de validación tomando el valor mínimo del número de archivos que tenemos en la imagen de entrenamiento dividido por el tamaño de los mini lotes anterior mente creados, para entrenar la red tenemos que especificar varias opciones de entrenamiento.

Tomamos una variable de entrenamiento y usamos la opción de entrenamiento para especificar las opciones, el primer argumento es el argumento de aprendizaje de regla que para este caso usamos 'sgdm' que significa impulso de gradiente estocástico, seguido de la variable mini lote que ya hemos definido, utilizamos una variable de época de 6 y una tasa de aprendizaje es 3×10^{-4} , utilizamos la variable 'Shuffle', 'every-epoch', para barajar los datos de entrenamiento en cada época, los datos de validación se almacenan en la variable de imagen de validación redimensionada y creamos una variable de frecuencia de validación, como queremos observar el proceso de entrenamiento utilizamos el siguiente comando de opciones de entrenamiento 'Plots', 'training-progress' y por ultimo clasificamos nuevamente la variable net para iniciar el proceso de entrenamiento.

```
Size_of_Minibatch = 5;
Validation_Frequency = floor(numel(Resized_Training_Image.Files)/Size_of_Minibatch);
Training_Options = trainingOptions('sgdm', ...
    'MiniBatchSize', Size_of_Minibatch, ...
    'MaxEpochs', 6, ...
    'InitialLearnRate', 3e-4, ...
    'Shuffle', 'every-epoch', ...
    'ValidationData', Resized_Validation_Image, ...
    'ValidationFrequency', Validation_Frequency, ...
    'Verbose', false, ...
    'Plots', 'training-progress');

net = trainNetwork(Resized_Training_Image, New_Layer_Graph, Training_Options);
```

Una vez ejecutado todo el código la red neuronal iniciara un proceso de entrenamiento que demora entre unos 15 y 28 minutos, variable que cambia según el tipo de procesador el equipo donde se está ejecutando y de la cantidad de imágenes comprendidas en la base de datos.

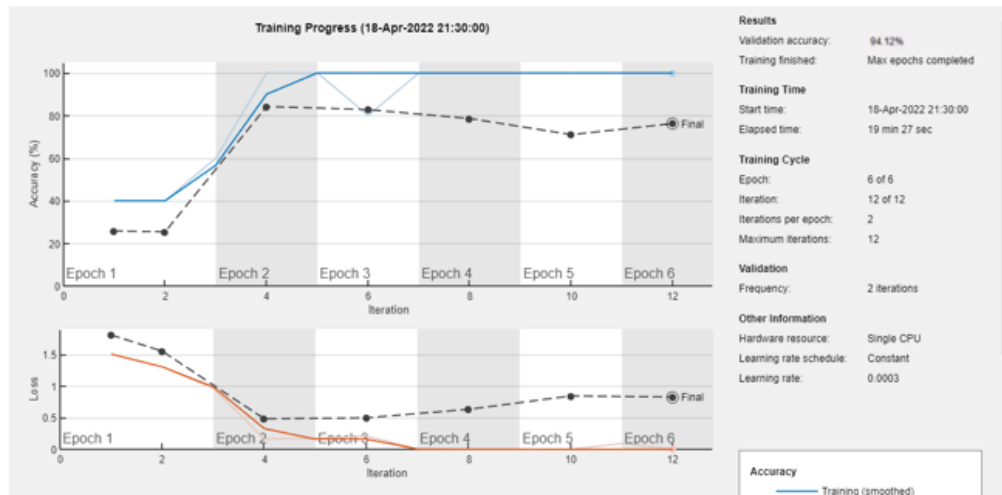


Figura 4.9.5 Arquitectura de entrenamiento de la red neuronal. Fuente del autor

Completado el proceso de entrenamiento tal como se observa en la figura 4.9.5 la precisión de la red es del 94,12% lo que nos da un excelente margen de precisión a la hora de ejecutar el algoritmo. Teniendo la parte del entrenamiento lista creamos una nueva función que llamamos red de prueba.

Iniciamos llamando la función net y a la imagen como entrada, luego creamos una variable para cargar la imagen y otra para cambiar el tamaño de la imagen de 224 x 224 pixeles, para clasificar la imagen usamos la variable clasificadora y usamos etiquetas para nombrar la imagen y colocarle el porcentaje de probabilidad.

```
function test_network(net, ~ )
I = imread("captura.jpg");
T = imresize(I, [224, 224]);
[Label, Probability] = classify(net, T);
imshow(T);
title(string(Label) + " = " + num2str(max(Probability)*100) + "%");
```


5. PROTOCOLO DE PRUEBAS

Una vez finalizado el algoritmo y ubicada la cámara en la posición deseada se procedió a insertar en la línea 15 moldes con anomalías para comprobar el funcionamiento del algoritmo y comprobar que los resultados sean confiables, al ser un objetivo de este trabajo de grado de discriminar los moldes en moldes normales y con anomalías, las estadísticas a evaluar determinaron la eficiencia del algoritmo para cada una de estas, de igual manera se evaluó la eficiencia total del algoritmo, es decir la cantidad de moldes buenos que pasan por malos (falso rechazo), como la cantidad de moldes malos que pasan por buenos (falsa aceptación). El protocolo de pruebas consistió en:

- Obtener las imágenes en un ambiente controlado.
- Modificar el algoritmo para que este de forma cíclica evalué la base de datos previamente revisada y muestre en pantalla una señal de alerta
- Evaluar el estado del molde detectado
- Evaluar la eficiencia del algoritmo.

Puesta en marcha el algoritmo para la detención de anomalías en algoritmo permite la detención de los defectos planteados para los moldes tal como se muestra en la figura 5.

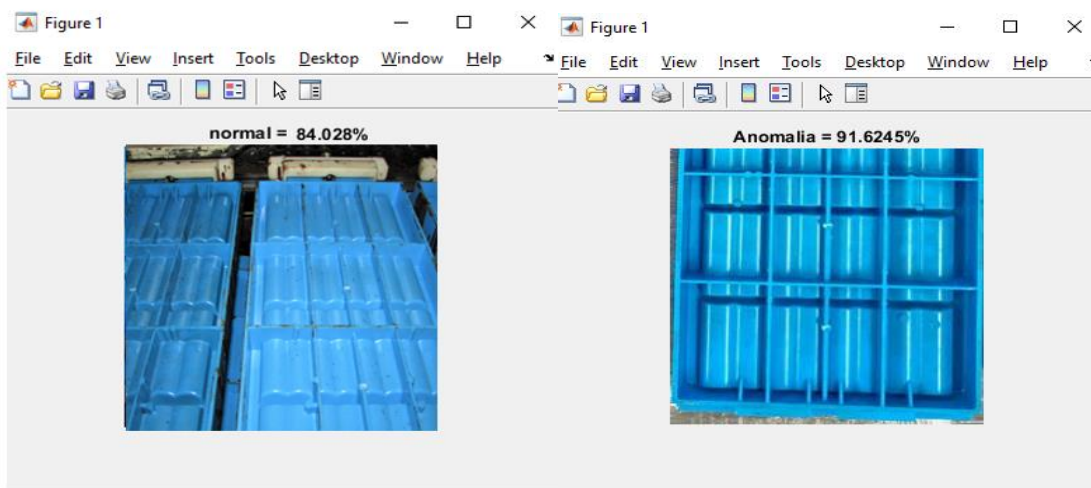


Figura 5. Imágenes de resultados. Fuente del autor

Una vez realizaron en total 20 ensayos a diferentes tiempos según lo permitido por la empresa para realizar estas validaciones, en consecuente a ello se obtuvieron 17 aciertos, 2 falso rechazos que correspondían a dos moldes buenos pero que estaban sucios y 1 falsa aceptación tal como se encuentran tabulados en la tabla 3, grafico 2 y figura 5.1.

Numero de prueba	Clasificacion	Porcentaje de clasificacion	Eficiencia
1	Anomalia	99,9905%	Real
2	Anomalia	99,9105%	Real
3	Anomalia	99,8702%	Real
4	Anomalia	99,8869%	Real
5	Normal	87,068%	Real
6	Normal	84,0280%	Real
7	Anomalia	99,8795%	Real
8	Anomalia	99,7998%	Falso rechazo
9	Normal	54,8793%	Real
10	Normal	60,1240%	Real
11	Normal	89,6743%	Real
12	Normal	90,9034%	Real
13	Anomalia	99,9067%	Falso rechazo
14	Normal	99,0784%	Real
15	Anomalia	99,8877%	Real
16	Normal	99,6723%	Real
17	Normal	89,9053%	Real
18	Anomalia	99,9011%	Real
19	Anomalia	99,7865%	Falsa aceptacion
20	Normal	99,7834%	Real

Tabla 3. Resultados de validación con la puesta en marcha del algoritmo. Fuente del autor

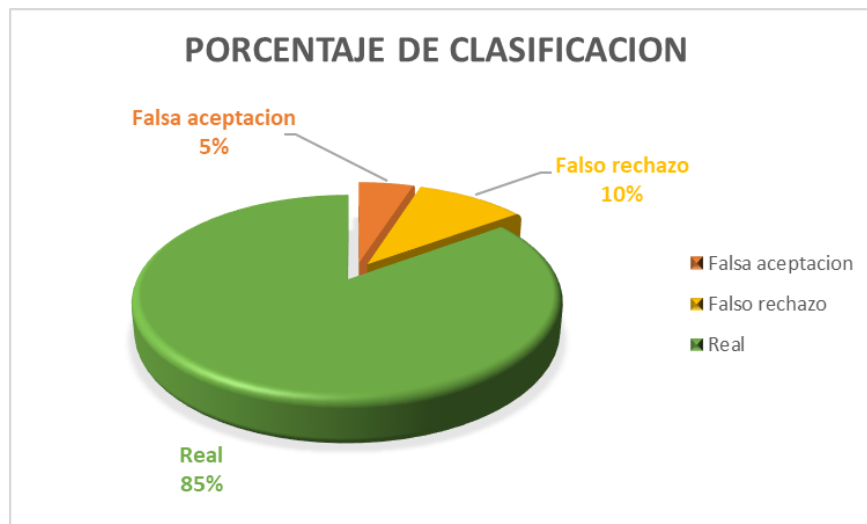


Grafico 2. Porcentaje de validación con la puesta en marcha del algoritmo. Fuente del autor

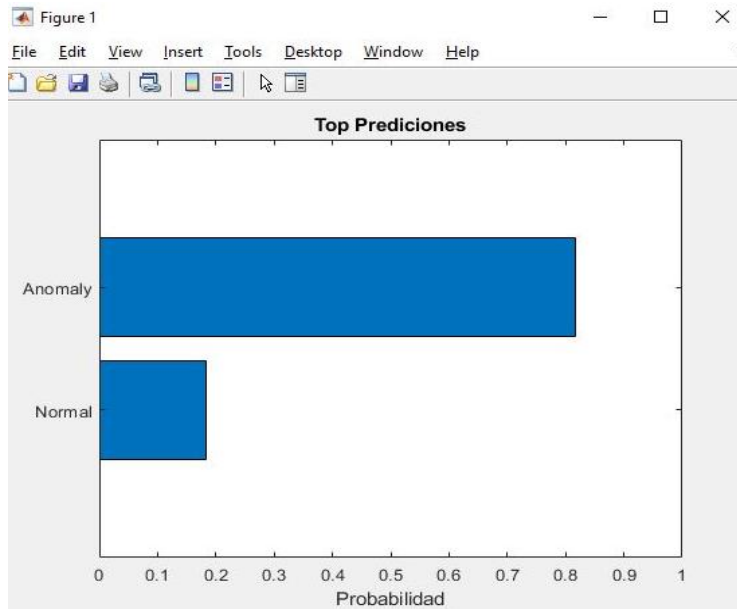


Figura 5.1. Top de predicciones de clasificación. Fuente del autor

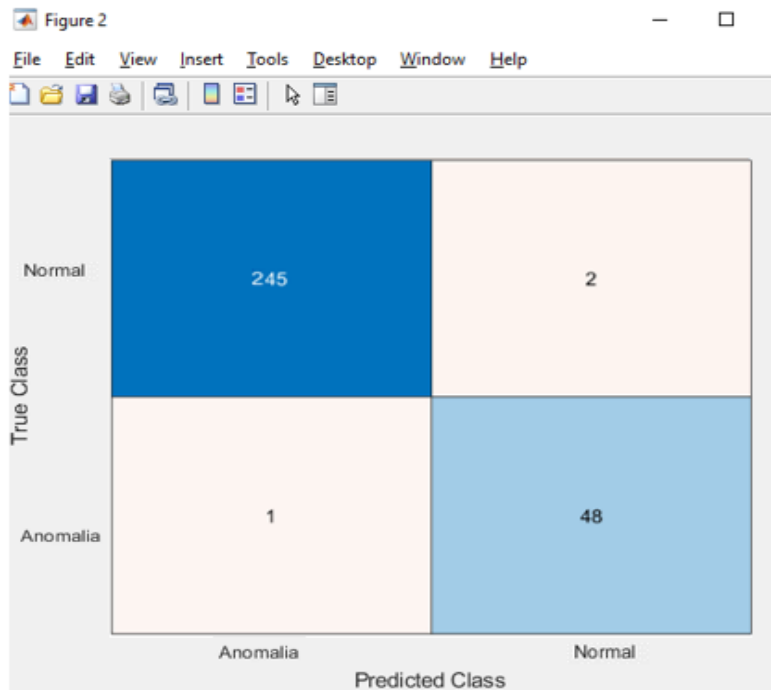


Figura 5.2. Matriz de confusión. Fuente del autor

Matriz de confusión donde se aprecia la precisión y exactitud de la red.

Como mejora se inserta un algoritmo de amplificación de imagen como se observa en la figura 5.3, la cual permite al operador de la línea validar el estado del molde que se acaba de ser detectado y con esto comprobar si efectivamente se debe iniciar el protocolo de inocuidad por molde roto.

```
my_large_image_display(T)
function my_large_image_display(I)
hfig = figure('Toolbar','none', ...
             'Menubar','none', ...
             'Name','My Large Image Display Tool', ...
             'NumberTitle','off', ...
             'IntegerHandle','off');

himage = imshow(I);
hpanel = imscrollpanel(hfig,himage);
set(hpanel,'Units','normalized','Position',[0 .1 1 .7]);
hMagBox = immagbox(hfig,himage);
pos = get(hMagBox,'Position');
set(hMagBox,'Position',[0 0 pos(3) pos(4)]);
hovervw = imoverview(himage);
end
```

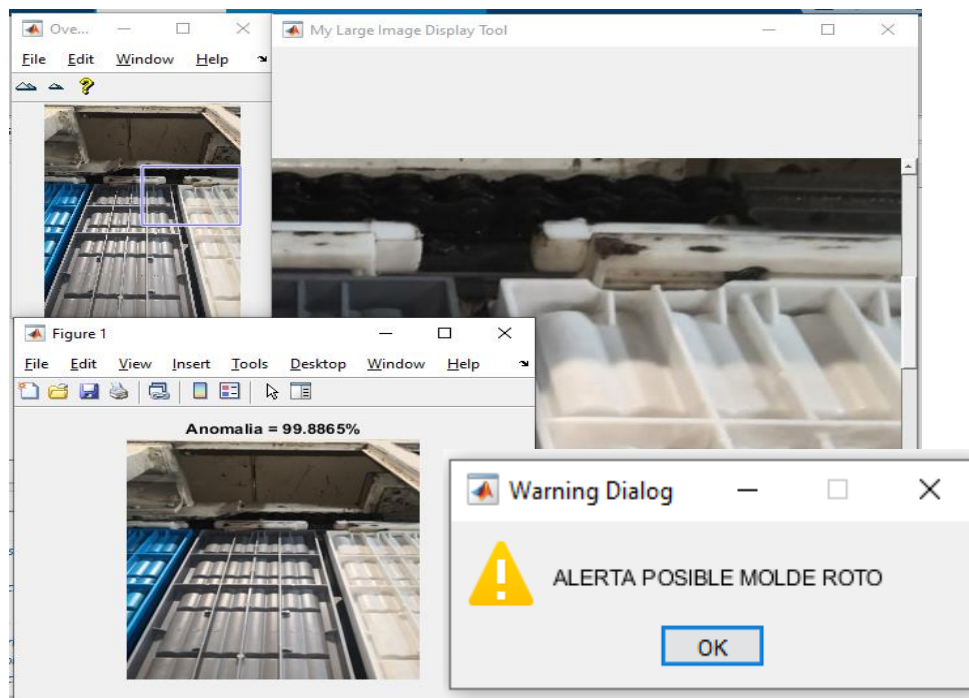


Figura 5.3. Visualización de la anomalía mediante el uso del algoritmo de aumento. Fuente del autor

Una vez el sistema genera la alerta de posible molde roto el operador por medio de la función de lupa o aumento verificara desplazándose por toda la imagen y analizar tramo a tramo y comprobar si efectivamente se trata de un molde detectado con anomalía, o caso contrario corresponde a un falso rechazo, activando así el protocolo de parada de producción por posible molde roto tal como se muestra en el gráfico 3, el cual se basa en el protocolo de búsqueda de piezas faltantes que tiene casa luker, pero en el que se le incorpora las actividades a realizar con el algoritmo de verificación.

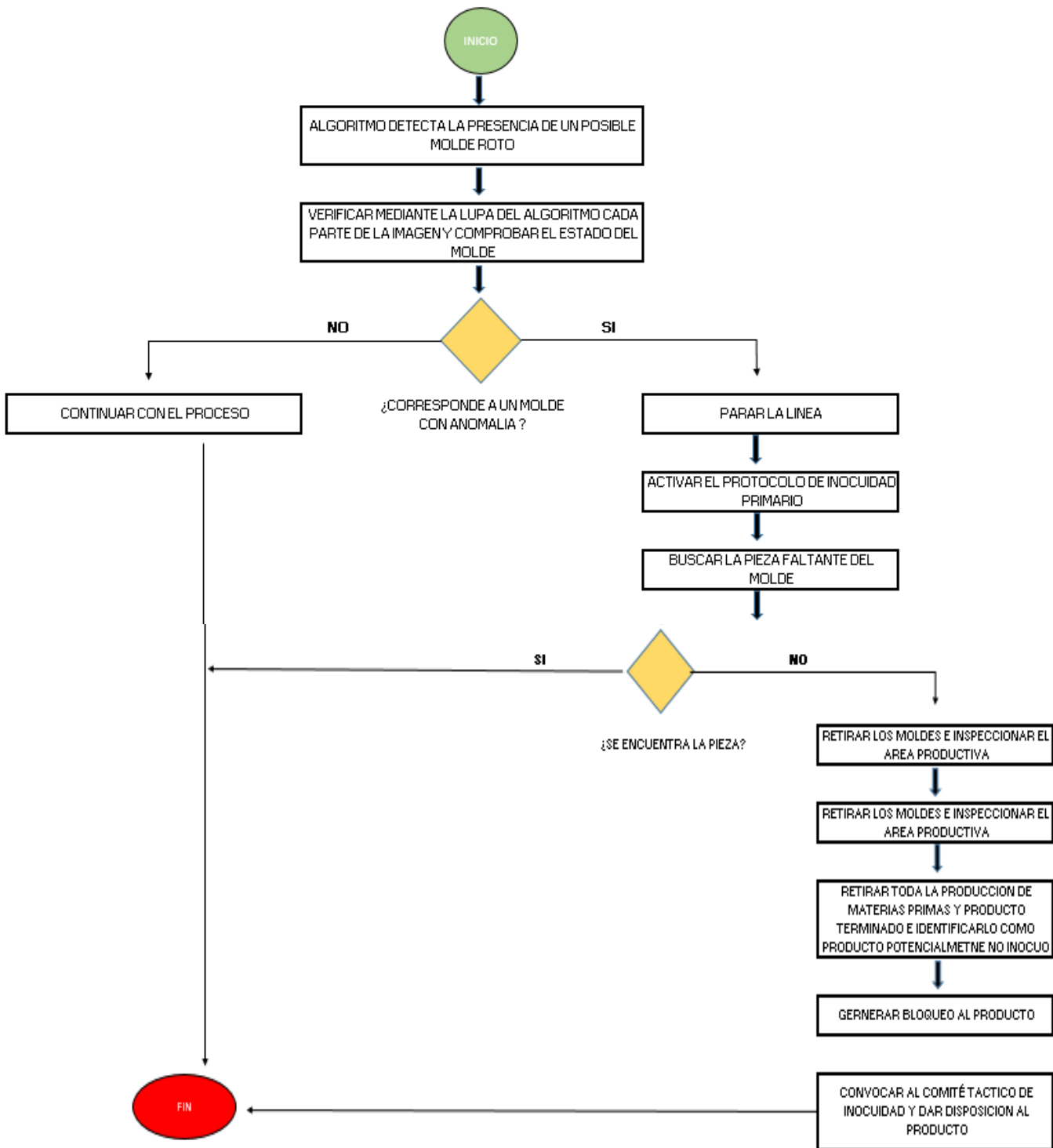


Grafico 3. Protocolo para parda de produccion cuando se identifica una alerta del algoritmo. Fuente del autor.

Durante el procesamiento del algoritmo aparte de reconocer las imágenes ya mencionadas en las estadísticas anteriores, el sistema reconoció un molde que aún no se había partido en su totalidad como se muestra en la figura 5,4, lo que quiere decir que el sistema también está en la capacidad de alertar sobre una posible ruptura de molde evitando así una futura parada de la línea para activar el protocolo de búsqueda de pieza faltante descrito en el diagrama 3.

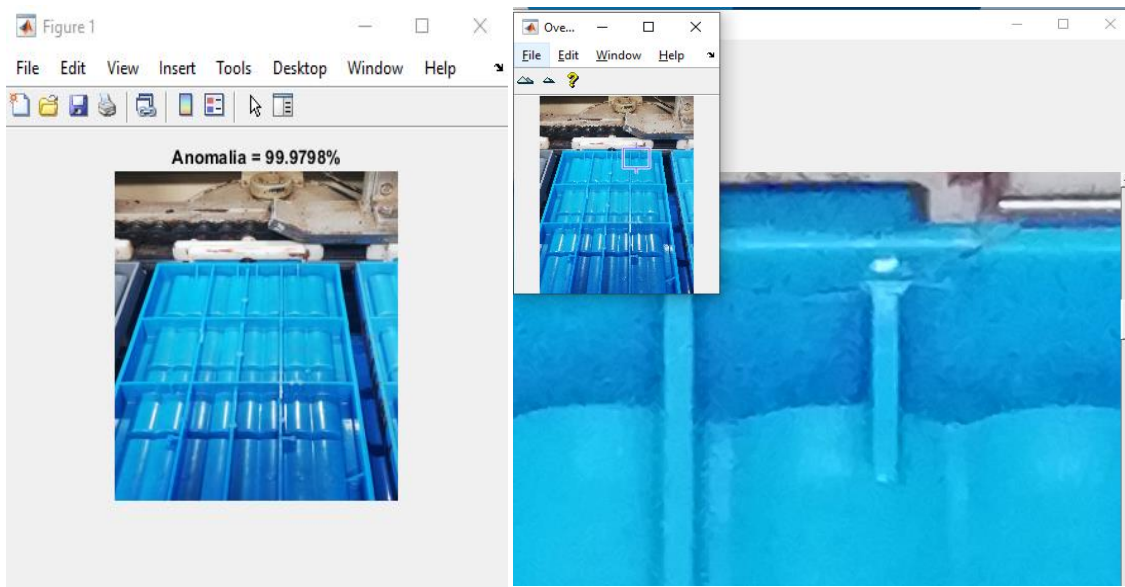


Figura 5.4. Visualización de la anomalía mediante la lupa. Fuente del autor

6. RESULTADOS Y CONCLUSIONES

1. Como primera conclusión se analizaron las variables de velocidad de recorrido del molde, área del molde, textura del molde, color del molde que intervienen en el proceso de fabricación de chocolate se tomó la velocidad mínima de 300 RPM o 10 moldes por minuto hasta las 750 RPM o 25 moldes se logró capturar la imagen del molde en 3 segundos, es decir el algoritmo de captura está en la capacidad de leer 20 moldes por minuto, siendo así, cuando la línea de producción está trabajando a máxima velocidad de 750 RPM el sistema cumple con la validación del 80% de los

moldes que inyecta la línea en un minuto, cabe resaltar que la operación normal de la línea es de entre 700 RPM y 730 RPM es decir entre 20 y 24 moldes por minuto, lo que amplía el porcentaje de validación de entre el 80% y 99% de los moldes que son inyectados en la línea de producción.

Como el algoritmo de captura va enlazado con el test_network de validación, la velocidad de procesamiento es la misma que la de captura.

En cuanto al color del molde y textura del mismo, no fue una variable que afectara en la detección de las anomalías puesto que estas dependen de la validación de la base de datos y de la resolución de las imágenes que para este caso la más óptima fue de 96PPP.

2. Se logró diseñar una base de datos de imágenes con anomalías generadas por eventos de inocuidad con moldes rotos, junto con imágenes de moldes sin anomalías para contrastarlas mediante el procesamiento de imágenes en Matlab, a través de un algoritmo que permita procesar la captura imágenes para compararla por medio de contrastes detectado así las anomalías presentadas en los moldes.
3. Durante la funcionalidad del sistema se comprobó que es efectivo ya que cumple con un porcentaje de aceptación del 85% lo que permite cumplir con los objetivos planteados, aunque es un porcentaje aceptable aún queda un 15% de margen de error, margen que se puede reducir volviendo mucho más robusta la base de datos, a mayor cantidad de imágenes el sistema será mucho más preciso, otra variable a mejorar es la toma de imágenes con una cámara superior a los 36 megapíxeles ya que esto permitirá una mejor validación de las imágenes.
4. Cumple con las validaciones de velocidad y validación de los moldes por minuto, disminuye el tiempo de revisión de mano de obra física para la validación de los moldes y disminuye el tiempo de parada relacionado con eventos de inocuidad de molde roto de alrededor de un 90%, puesto que en este momento la empresa solo dispone de una persona cada 4 horas para realizar la validación.

5. Se diseñó un protocolo de detención de la maquina al momento de detectar un molde roto, para que una vez generada la alerta de la posible anomalía se pueda retirar el molde de la linea y se pueda continuar la produccion sin ningún riesgo.

Para futuros proyectos e implementación

- Se recomienda la ampliación de la base de datos, con imágenes de mayor calidad y resolución, realizar una base mucho más robusta que permita disminuir el porcentaje de error en la validación,
- Una de las limitantes puede ser el sistema de procesamiento ya que se utilizó el equipo disponible del autor el cual no es lo suficiente mente acto para el procesamiento de datos, el uso de un equipo con mayor memoria RAM y mejor procesador permitirá disminuir los tiempos de ejecución y entrenamiento de la red neuronal.
- Aunque la iluminación utilizada por el autor fue buena, se propone mejorar la iluminación en la zona de retorno de moldes, sitio que fue utilizado para la captura de imágenes, puesto que las partes metálicas de la linea de produccion generan reflejos del flash de la cámara generando disminución de la calidad de la imagen,
- Migrar el código de procesamiento a otro lenguaje comercial, el proyecto fue generado en Matlab, el cual es un software bastante robusto y que se debe pagar una licencia para su funcionamiento, la cual no es recomendable adquirir solo para este procesamiento, se recomienda migrar el código a otro entorno como Python o R.

7. BIBLIOGRAFIA

- [1]. Sustitución de cultivos: llega la hora del cacao, (2022).
<https://www.portalparalapaz.gov.co/publicaciones/939/sustitucion-de-cultivos-llega-la-hora-del-cacao/>
- [2] Chocolatería, una industria que crece sosteniblemente en el país (2021).
Andi. <http://www.andi.com.co/Home/Noticia/15568-chocolateria-una-industria-que-crece>
- [3]. Casaluker, (2013). Manual de operación super cavemil, Daruma,
- [4] Cabrellon, molds, Polycarbonate molds and BPAfree.
<https://www.cabrellon.it/en/molds/materials/>
- [5]. Casaluker, (2013). Manual de operación super cavemil, Daruma,
- [6]. P. Carela. (2019) trabajo de fin de grado grado en ingeniería para el tratamiento digital de imágenes en matlab para la detección de fallas. Universidad politécnica de Cataluña, España
- [7] https://la.mathworks.com/help/matlab/creating_plots/image-types.html
- [8] J. Soto, 2020, Convolución Matricial Aplicado al Procesamiento de Imágenes, Instituto Tecnológico de Costa Rica, Costa Rica
- [9]. Processing. 2022, Procesamiento de imágenes,
http://www.mywonderland.es/curso_js/processing/pro_procesar.html
- [10]. Bootcamp AI, 2019, Intro a las redes neuronales convolucionales,
<https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8>
- [11]. Matlab, 2022, Redes neuronales convolucionales,
<https://la.mathworks.com/discovery/convolutional-neural-network-matlab.html#:~:text=Flujo%20de%20trabajo%20de%20deep,y%20datos%20de%20series%20temporales.>

[12]. Bootcamp AI, 2019, Intro a las redes neuronales convolucionales, <https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8>

[13]. Matlab, 2022, Redes neuronales convolucionales, <https://la.mathworks.com/discovery/convolutional-neural-network-matlab.html#:~:text=Flujo%20de%20trabajo%20de%20deep,y%20datos%20de%20series%20temporales.>

[14]. Matlab, 2022, GoogLeNet convolutional neural network, <https://la.mathworks.com/help/deeplearning/ref/googlenet.html>

[15]. Train and test data, 2022, A blog on Data Science & Machine Learning for beginners, <https://dataandbeyond.wordpress.com/2017/08/24/split-of-train-and-test-data/>

[16]. P. Carela. (2019) trabajo de fin de grado grado en ingeniería para el tratamiento digital de imágenes en matlab para la detección de fallas. Universidad politécnica de Cataluña, España

```
Dataset = imageDatastore('Dataset', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
[Training_Dataset, Validation_Dataset] = splitEachLabel(Dataset, 7.0);

net = googlenet;
analyzeNetwork(net)

Input_Layer_Size = net.Layers(1).InputSize(1:2);
Resized_Training_Image = augmentedImageDatastore(Input_Layer_Size, Training_Dataset);
Resized_Validation_Image = augmentedImageDatastore(Input_Layer_Size, Validation_Dataset);

Feature_Learner = net.Layers(142);
Output_Classifier = net.Layers(144);

Number_of_Classes = numel(categories(Training_Dataset.Labels));

New_Feature_Learner = fullyConnectedLayer(Number_of_Classes, ...
    'Name', 'Molde Feature Learner', ...
    'WeightLearnRateFactor', 10, ...
    'BiasLearnRateFactor', 10);
New_Classifier_Layer = classificationLayer('Name', 'Molde Classifier');

Layer_Graph = layerGraph(net);

New_Layer_Graph = replaceLayer(Layer_Graph, Feature_Learner.Name, New_Feature_Learner);
New_Layer_Graph = replaceLayer(New_Layer_Graph, Output_Classifier.Name, New_Classifier_Layer);

analyzeNetwork(New_Layer_Graph)

Size_of_Minibatch = 5;
Validation_Frequency = floor(numel(Resized_Training_Image.Files)/Size_of_Minibatch);
Training_Options = trainingOptions('sgdm', ...
    'MiniBatchSize', Size_of_Minibatch, ...
    'MaxEpochs', 6, ...
    'InitialLearnRate', 3e-4, ...
    'Shuffle', 'every-epoch', ...
    'ValidationData', Resized_Validation_Image, ...
    'ValidationFrequency', Validation_Frequency, ...
    'Verbose', false, ...
    'Plots', 'training-progress');

net = trainNetwork(Resized_Training_Image, New_Layer_Graph, Training_Options);
```

```

function test_network(net, ~ )
I = imread("muesta.jpg");
T = imresize(I, [224, 224]);
[Label, Probability] = classify(net, T);
imshow(T);
title(string(Label) + " = " + num2str(max(Probability)*100) + "%");

my_large_image_display(T)
function my_large_image_display(I)
hfig = figure('ToolBar','none', ...
             'MenuBar','none', ...
             'Name','My Large Image Display Tool', ...
             'NumberTitle','off', ...
             'IntegerHandle','off');

himage = imshow(I);
hpanel = imscrollpanel(hfig,himage);
set(hpanel,'Units','normalized','Position',[0 .1 1 .7]);
hMagBox = immagbox(hfig,himage);
pos = get(hMagBox,'Position');
set(hMagBox,'Position',[0 0 pos(3) pos(4)]);
hovervw = imoverview(himage);
end

classNames = net.Layers(end).ClassNames;
numClasses = numel(classNames);
disp(classNames(randperm(numClasses,2))) |

    'Normal';
    'Anomalia';

[label,scores] = classify(net,T);
label;

[~,idx] = sort(scores,'descend');

idx = idx(2:-1:1);
classNamesTop = net.Layers(end).ClassNames(idx);
scoresTop = scores(idx);

figure
barh(scoresTop)
xlim([0 1])
title('Top Predicciones')
xlabel('Probabilidad')
yticklabels(classNamesTop)

C = confusionchart(numClasses,Label);

end

```