

Algoritmo de Control de Calidad en Piezas de Manufactura

M. Q. Jean Luis, *Ingeniería Mecatrónica (Universidad Veracruzana, Bv. Adolfo Ruíz Cortines 455, Costa Verde, 94294 Veracruz, Ver., México, zS18018322@estudiantes.uv.mx)*

Resumen— La visión artificial es una tecnología de la industria 4.0 que emula la visión humana la cual consiste en identificar, interpretar, procesar y analizar las imágenes y videos tomadas por cámaras. Esto con el objetivo de generar información que pueda ser útil para que las máquinas comprendan las imágenes a través de procesos digitales que permitan tomar medidas de control en los sistemas que las incorporen.

En este reporte de proyecto se mostrará cómo podemos manejar los datos de manufactura de un sistema de producción mediante la implementación de inteligencia artificial para el reconocimiento de piezas para un control de calidad automatizado para detectar alguna desviación de calidad y así poder dar mantenimiento preventivo.

I. INTRODUCCIÓN

Las plantas de manufactura moderna emplean enormes y complejas máquinas y equipamiento de producción, empaquetado y de testeo, entre muchas otras más. Aunque son diseñadas para tener unos altos estándares de calidad y estabilidad, ninguna maquina es inmune al desajuste y la variación, a través del tiempo componentes de las maquinas comienzan a comportarse ligeramente diferente a como fueron instaladas y ajustadas en un inicio, y estos cambios a su vez provocan variaciones en la producción a la larga lo que se ve reflejado en la calidad de los productos y sobre todo en un incremento en los costos de producción.

Encontrar estas desviaciones es importante por las siguientes razones:

- Identificar la máquina dentro de la cadena de producción que requiera reparaciones o mantenimiento
- Corregir o calibrar los desvíos por software o por hardware
- Vincular la calidad de producción con cada máquina
- Investigar problemas de rendimiento y fallas de fiabilidad que hayan causado el desajuste o incrementado las desviaciones



Figura 1. Diagrama lógico de algoritmo

II. ESTADO DEL ARTE

Una de las principales problemáticas en el estudio de los datos en inteligencia artificial es la clasificación de las características ya sean imágenes de bases de datos de las cuales se puedan cuantificar y obtener descriptores que sean lo más exactos posibles, para ello hay que saber a qué grupo de conjuntos pertenecen. Para todo esto se necesitan técnicas de ciencia de datos como lo son las Redes Neuronales y Machine Learning (ML).

Una de las primeras proposiciones de las neuronas artificiales fue en año de 1943 por Warren McCulloch, un neuropsicólogo, y Walter Pitts, en colaboración en la Universidad de Chicago. El concepto de Redes Neuronales fue propuesto por primera vez por Alan Turing en el año de 1948, en su artículo "Intelligent Machinery" con la creación de la prueba de Turing, una máquina compuesta de numerosas compuertas lógicas tipo NAND interconectadas, en esencia, una precursora de lo que sería hoy en día una red neuronal moderna.

Las redes neuronales han avanzado en gran medida en los últimos años para lo que es el *Procesamiento del Lenguaje Natural (NPL por sus siglas en inglés)*. Lo que se busca con ellas es simular el lenguaje natural de una neurona y así entrenar modelos de algoritmos que aprendan a resolver determinadas tareas. Para estos modelos de la programación fueran posibles se necesitaron numerosos avances de hardware en cuanto a GPUs y TPUs se refiere, a tal medida que hoy en día es posible incluso el uso de supercomputadoras de manera virtual para ciencia de los datos (lo que se le conoce como *Cloud Computing*) para facilitar las bondades por potencia de hardware para el desarrollo de modelos más complejos además de una basta recopilación de *datasets* de uso libre.

* Reporte de Proyecto de la Facultad de Ingeniería Eléctrica y Electrónica (FIEE), Universidad Veracruzana – Tópicos Avanzados de Inteligencia Artificial

III. DISEÑO E IMPLEMENTACIÓN

Para el modelo implementado se trabajó con el lenguaje de programación de Python en la versión 3, lo que permitió el uso de diversas herramientas para la identificación de las imágenes que se querían procesar. Las principales librerías utilizadas fueron OpenCV, Matplotlib, Numpy, Pandas. Lo que facilita la implementación de algoritmos que incorporen Visión por Computadora, Ciencia de Datos, Redes Neuronales y Machine Learning. Otra herramienta muy útil a la hora de programar redes neuronales es Google Colab, un entorno de programación basado en Jupiter Notebook, que permite la programación en Python de tu algoritmo con *cloud computing*.

Primero se comenzó recopilando unos pequeños datasets sobre las piezas de manufactura que se desean analizar. En mi caso opte por trabajar con un conjunto de keycaps que tenía a mi alcance en mi oficina, lo que me permitió clasificarlas primero manualmente por categorías para su posterior procesamiento. Cabe aclarar que las tomas para las muestras tienen que ser lo más uniformes posibles y capturarse bajo las mismas condiciones para el correcto funcionamiento del algoritmo. A lo que se refiere, tener las mismas condiciones de luz, mismo plano, categorizar correctamente el mismo tipo de piezas entre otras más.

Posteriormente para la parte de la maquinaria y los robots utilizados opté por simularlos meramente bajo algoritmos computacionales, es decir, el sistema de producción de manufactura en el cual trabajé es un entorno simulado y controlado para evitar cualquier tipo de falla. La descripción del sistema es la siguiente: se cuenta con una red de producción de una máquina que produce keycaps, las cuales pasan a través de una banda transportadora en donde se cuenta con una cámara que cuenta con visión artificial y un brazo robótico que las clasifica de acuerdo con sus características. Si el modelo de la keycap no concuerda con el dataset entonces es retirada por el brazo.

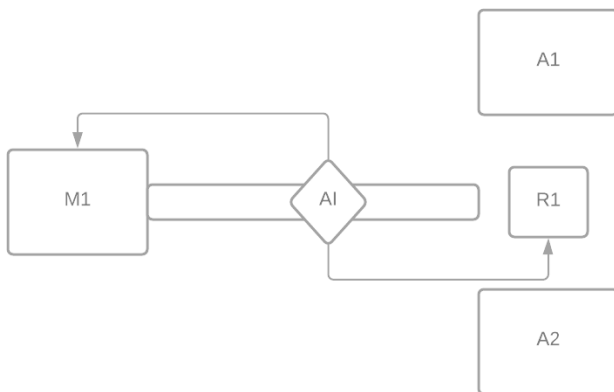


Ilustración 1. Posicionamiento de planta

Para el procesamiento de la imagen se comienza adaptando las muestras a escala para dar uniformidad en el tamaño. Se aplica un filtro gaussiano para evitar cualquier tipo de *ruido impulsional* que se tenga y se preparan las imágenes para su posterior uso separándolas por ahora en escala de grises, RGB y *HSV*, necesarias para la identificación de bordes, colores y

esquinas y demás características. Haciendo una pequeña recapitulación, las computadoras por sí solas no son capaces de identificar las imágenes como una persona lo haría por lo que es necesario adaptarlas a una entrada de una matriz conformada por unos y ceros, que esta conformada por cada píxel dentro de la imagen, leídos además en otra escala de colores, esto con el fin de que sea más fácil interpretar la información de la imagen.

Procedemos a identificar los bordes de la pieza con *Canny* y aplicar un *Convex Hull* a los puntos reconocidos, para encapsular así en una *máscara* la parte que se requiere de la imagen (en este caso la pieza a identificar omitiendo el fondo).

Continuamos con una de las técnicas más usadas en visión por computadora, la generación de *histogramas* para el tratado de imágenes mediante la ecualización de esta, además de ser más útil para interpretar las intensidades y características de las imágenes por medio de gráficos que interpretan intensidades en las lecturas de las imágenes. Una vez calculados los histogramas procedemos a su ecualización.

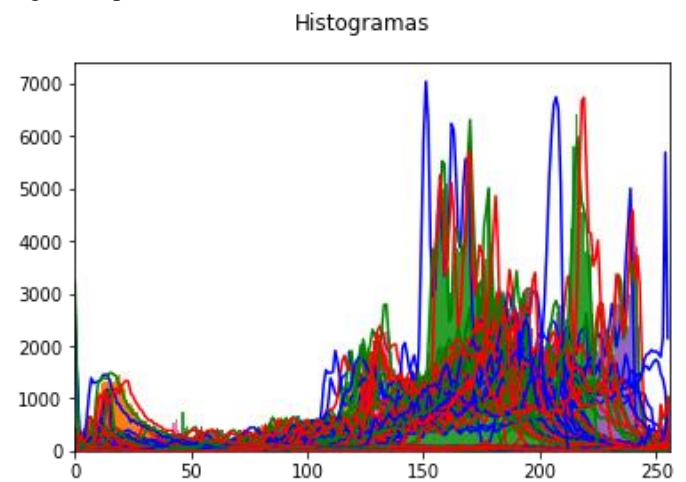


Figura 2 Histogramas en escalas de RGB de las piezas analizadas (sin ecualizar)

Por último, encontramos las esquinas con la función de Harris y determinamos los puntos de las esquinas para poder clasificarlas mejor mediante datos estadísticos que se utilizarán a la hora de implementar la red neuronal. En este caso usamos la herramienta *scatter* que permite hacer un gráfico de los puntos característicos de las piezas que analizamos en nuestros pequeños datasets de keycaps.

Ahora solamente que la clasificación de las piezas ya esta hecha podemos vincular cada pieza con un id único que nos permitirá rastrear la pieza referida a la maquina que la produjo y así en base al número de piezas totales sacar un relación de las piezas buenas contra las malas, y conforme a esas estadísticas relacionarlas con la desviación de cada máquina, es decir, monitorear individualmente la producción de cada máquina al estar vinculadas las piezas provenientes de las mismas para así poder sacar un margen de repetibilidad y que tanto estas se desvían de producir exactamente las mismas piezas y cuando la desviación sobrepasa el umbral entonces tomar acción en el proceso de manufactura, ya sea calibrando nuevamente las máquinas o dando mantenimiento físico, esto con el objetivo de reducir costos, tanto en pérdidas por merma,

así como al tener que parar la producción por una mala gestión en los tiempos de mantenimiento de cada máquina.

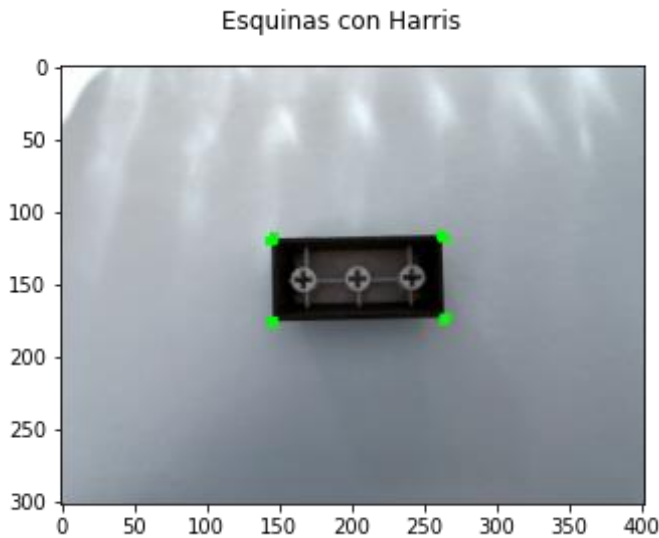


Figura 3 Esquinas identificadas con Harris y la máscara ya hecha

IV. RESULTADOS

En este proceso se hace la representación con una simulación de los procesos de producción de manufactura, pero sienta las bases para ser adaptado a un modelo real y escalable que incorpore todos los procedimientos aquí descritos, por consiguiente la obtención de las características de los datasets deberán de estar bien definidas para cada caso de uso dependiendo de que se requiera analizar, en este ejemplo opté por keycaps, pero bien se podría analizar algún otro tipo de producto simple del cual pueda ser posible la fácil extracción de sus características.

Los primeros resultados que podemos sacar de la identificación de las piezas son la distribución de las características de las piezas clasificadas en dos, principalmente las piezas buenas y las piezas malas mostradas en la figura [4].

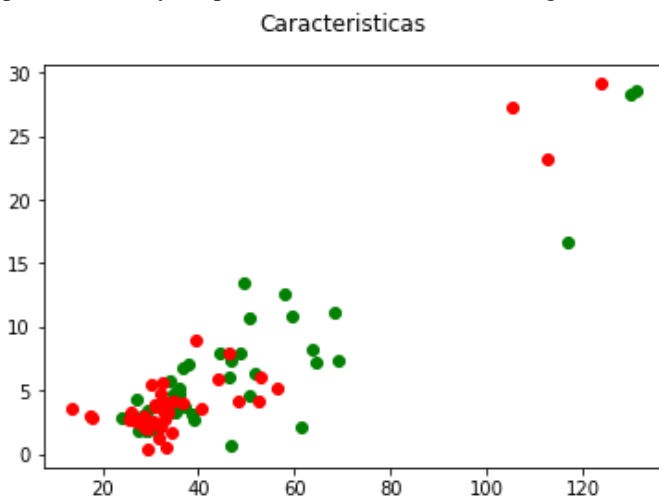


Figura 4 Obtención de características para separación de las piezas buenas contra las malas

Posteriormente proseguimos a meter estas características dadas en una red neuronal, que será capaz de identificar por sí sola cuales son las piezas buenas y cuáles son las piezas malas, dadas ciertas entradas y salidas, obviamente dependiendo de las características dadas que requiera el usuario, si lo que la persona busca es identificación por textura, tamaño, tipo, etc. Ya dependerá de los parámetros que se introduzcan en la red neuronal.

Finalmente generamos un reporte de todas las máquinas involucradas en el proceso de manufactura y las cuales cuya desviación haya sido mayor para así poder sacar una relación y poder medir su desempeño.

Como era de esperarse los resultados en las pruebas de entrenamiento fueron bastante imprecisos, obteniendo solamente hasta el 60% de precisión en el reconocimiento de las piezas, esto debido al dataset tan pequeño con el que se cuenta (contando que ha sido recopilado exclusivamente para este proyecto), además de la clasificación de las características que simplemente no son suficientes para que el modelo pudiera aprender por sí solo.

Esto se mitiga con los numerosos datasets de materiales que se pueden encontrar de la industria, y aun con un poco de trabajo y clasificando bien las etiquetas de estos podríamos llegar a mejorar bastante los números del modelo de redes neuronales, esperando así que el algoritmo aprenda a identificar si una pieza es la solicitada o no.

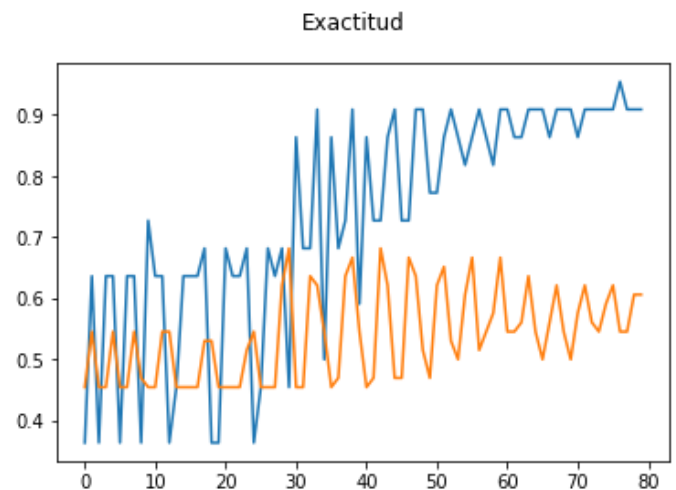


Figura 5 Exactitud del aprendizaje del algoritmo

Error

VI. GLOSARIO

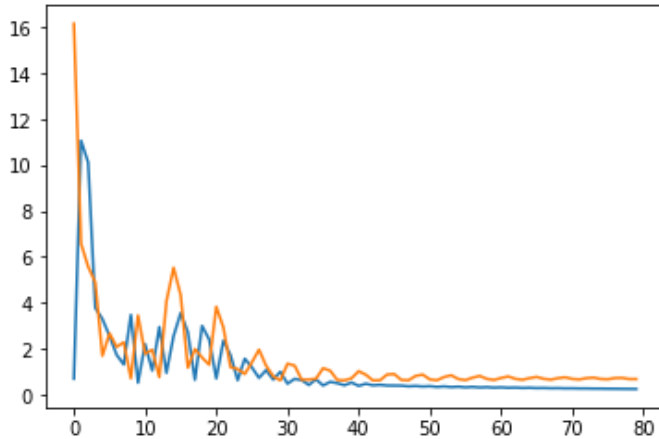


Figura 6 Error del aprendizaje del algoritmo

V. CONCLUSIONES Y TRABAJOS FUTUROS

A través de este trabajo vimos las posibles clasificaciones de los datos para su análisis y la separación por características de las piezas de manufactura lo que permite relaciones entre las piezas en buen estado y las piezas en mal estado (o simplemente la separación de sus características) gracias a un pequeño dataset de imágenes que representan cada conjunto por lo que permite que el algoritmo comience a aprender a clasificar con los datos analizados. Por lo que podemos decir que es posible entonces mediante datos estadísticos realizar la separación y la vinculación de las piezas de manufactura a cada máquina que las produce para así ir visualizando la repetibilidad y precisión de la producción en base a los datos que se analizan del dataset.

Sin embargo, hay ciertas limitaciones como por ejemplo, el tamaño del dataset principalmente, puesto que se realizan las tomas bajo ciertas condiciones controladas para permitir la uniformidad de las muestras, además de que los datasets son imágenes capturadas del espacio de trabajo y no se está capturando video en tiempo real por lo que se espera poder hacer esa implementación y mejorar la robustez del algoritmo mediante la toma de más características de las piezas para poder separarlas y clasificarlas de una mejor manera.

Canny: es un algoritmo popular de detección de bordes para técnicas de visión por artificial desarrollado por John F. Canny.

Cloud Computing: es una tecnología que permite el uso remoto de servicios en la nube como lo son softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local.

Convex Hull:

Dataset: es una colección de conjuntos de información o datos que se relacionan compuesto por elementos separados pero que una computadora puede manipular como una unidad.

Histograma: es un gráfico de la representación de las distribuciones de frecuencia, en el que se proyectan las intensidades de color de la imagen dentro de unas coordenadas por medio de rectángulos (para escala de grises) o líneas (en caso de proyección de intensidades de RGB).

HSV: (Hue, Saturation, Value por sus siglas en inglés) es un modelo de color en términos de sus componentes tinte, matiz o tonalidad (hue, en inglés), saturación (saturation) y brillo o valor (value).

Mascara: es un conjunto de datos extraídos selectivamente de otro conjunto de datos para una manipulación más fácil.

Procesamiento del lenguaje natural: es el campo de conocimiento de la Inteligencia Artificial que se ocupa de investigar la manera de comunicar las máquinas con las personas mediante el uso de lenguas naturales, como la escritura y la interpretación de textos entendibles para el humano.

Ruido impulsional: es un tipo de distorsión o alteración de la imagen causada por imperfecciones en la señal que captura el sensor.

TensorFlow: es una de las principales bibliotecas de código abierto para el aprendizaje automático utilizado en las redes neuronales y demás técnicas de inteligencia artificial.

TPU: unidad de procesamiento tensorial (TPU por sus siglas en inglés), es un circuito integrado mejorado y potenciado para el uso de aprendizaje automático desarrollado por Google.

VII. REFERENCIAS

- [1] P. N. Stuart J. Russell, Artificial Intelligence : A Modern Approach, Pearson, 2021.
- [2] A. K. Gary Bradski, Learning OpenCV, California: O'Reilly, 2008.
- [3] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.
- [4] R. E. W. Rafael C. Gonzalez, Digital Image Processing, New Jersey: Prentice Hall.
- [5] R. G. Duque, Python Para Todos.