TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Implementación de un sistema de seguridad basado en visión artificial para la activación del paro de emergencia en una Dobladora hidráulica CNC

PROYECTO DE INVESTIGACIÓN

PARA OBTENER EL TITULO DE

INGENIERO EN MECATRÓNICA

CON ESPECIALIDAD EN:

AUTOMATIZACIÓN Y CONTROL DE PROCESOS

P R E S E N T A

CRUZ SÁNCHEZ DIEGO IGNACIO
MATRICULA 201820819

DIRECTOR:

M. EN C. MARIO ALBERTO HERNÁNDEZ SORIANO

Coacalco de Berriozábal, Abril de 2025

Dedicatorias

Dedico este proyecto a quienes con su apoyo y motivación me inspiraron a lograr esta meta. A través de la implementación de visión artificial en dobladoras de lámina y el desarrollo de su programación, pude explorar y aplicar conocimientos en tecnología avanzada, contribuyendo a la optimización de procesos industriales. Agradezco a mis profesores, compañeros, y seres queridos por su confianza y respaldo a lo largo de este camino.

Agradecimientos

Agradezco a todas las personas y equipos que contribuyeron al éxito de este proyecto de visión artificial para dobladoras de lámina. Gracias a su apoyo y colaboración, logramos implementar la programación necesaria para optimizar el proceso y mejorar la precisión en el doblado. Este logro es un paso significativo en la aplicación de tecnologías avanzadas para la industria, y fue posible gracias a la dedicación y el compromiso de todos los involucrados. ¡Gracias por hacer de este proyecto una realidad!

Resumen

En este trabajo se presenta la propuesta de un sistema de monitoreo por medio de herramientas de visión artificial con el objetivo de activar un sistema de paro de emergencia. La metodología desarrollada emplea el lenguaje de programación Python y las librerías de OpenCV y MediaPipe para detectar si las manos de un operador están en una zona segura o insegura dentro del área de trabajo de una dobladora CNC. Los resultados muestran cómo funciona el sistema de monitoreo garantizando la seguridad del operador y mostrando la importancia de integrar nuevas tecnologías en el monitoreo industrial.

ÍNDICE

Capítulo I. Generalidades 7
1.1 Introducción 7
1.2 Justificación 8
1.3 Planteamiento del problema 8
1.4 Objetivos 9
1.5.1 Objetivo general 9
1.5.2 Objetivos específicos 9
1.5 Alcances y limitaciones 9
Visión Artificial 10
Conceptos Clave de la Visión Artificial 10
Procesamiento de Imágenes 10
Reconocimiento de Patrones 10
Aprendizaje Automático 11
Detección de Objetos 11
Aplicaciones de la Visión Artificial 12
Técnicas Comunes en la Detección de Objetos 12
Aplicaciones de la Visión Artificial en Diversos Sectores 14
21 Hands Landmark 17
Aplicaciones de la visión artificial industrial 18
Análisis de imagen 21
Lenguaje de programación 23
Python 24
Características de Python 24
Lo básico de Python 24
Tipos de datos 24
Funciones 25
Funciones, parámetros *args y **kwargs 25
Manipulación de cadenas 25
Estructuras de datos (colecciones) 25
Clases y objetos 25

Pylance 26
Serial Monitor 27
Librerías 28
Opencv 28
Mediapipe 29
Seguridad industrial en maquinaria 29
Integración de Sistemas de Visión Artificial en Seguridad Industrial 31
Dobladoras de lámina: ¿qué son y por qué son esenciales en la industria? 32
Tipos de dobladoras: manual, hidráulica y CNC 34
Proceso de fabricación y diseño de dobladoras de lámina 35
Programación en Arduino 38
Capítulo III. Desarrollo 40
Capítulo IV. Resultados 62
Capítulo V. Conclusiones 64
Referencias 65
ANEXO 1 67
ANEXO 2 68

INDICE DE FIGURAS

Figura 1 Detección de vehículos y señalamientos 14
Figura 2 Aplicaciones de la visión artificial en la industria 15
Figura 3 Detección de rostros 16
Figura 4 Red neuronal convolucionales 17
Figura 5 Deteccion de la palma de la mano 18
Figura 6 Puntos de referencia de la mano. 18
Figura 7 Visión Artificial en la Industria. 19
Figura 8 Software de análisis de imágenes 21
Figura 9 estructura de SDK 22
Figura 10 Esquema de sistemas de visión 31
Figura 11 Equipo de protección personal (EPP) 32
Figura 12 Maquina dobladora de láminas neumática. 34
Figura 13 Proceso de fabricación y diseño de dobladoras de lámina 35
Figura 14 Proceso de dobles 37
Figura 15 Lenguaje de Arduino 39
Figura 16 Visual Studio Code 40
Figura 17 Versión de Visual Studio Code 41
Figura 18 Bienvenida a Visual Studio 41
Figura 19 Área de Extensiones 42
Figura 20 Descarga de Extensiones 43
Figura 21 Librerías 43
Figura 22 Función de capturar Imagen. 44
Figura 23 Cámara de la Computadora 45
Figura 24 Código de Ejemplo para usar la cámara 46
Figura 25 Detección de mano con sus puntos 47
Figura 26 Función de seguimiento de manos 47
Figura 27 Código de muestra para detección de manos 48
Figura 28 Detección de bordes 49
Figura 29 Código de detección de bordes en objetos 50
Figura 30 Distancia de 1 metro 52
Figura 31 Distancia de 5 metros 52
Figura 32 Distancia de 10 metros 53
Figura 33 Impresión de texto Go 54
Figura 34 Impresión de texto Stop 54
Figura 35 Vista del área de la maquina dobladora con las manos puestas 55
Figura 36 Vista del área de la maquina dobladora sin manos puestas. 56
Figura 37 Software de Arduino 57
Figura 38 Programa de Arduino 58
Figura 39 Librerías de serial y time 59
Figura 40 Código con Arduino 59
Figura 41 Código con Arduino 60

Capítulo I.
Generalidades

1.1 Introducción

En el contexto de mejorar la seguridad y la capacidad de respuesta ante situaciones críticas, este trabajo se centra en la implementación de un sistema de detección de manos mediante visión artificial, utilizando las librerías OpenCV y MediaPipe. La necesidad de este proyecto surge con la intención de desarrollar un mecanismo de paro de emergencia en áreas específicas, como fábricas o laboratorios, donde la rápida identificación de gestos de manos puede ser crucial para prevenir accidentes o responder eficientemente a situaciones peligrosas.

La motivación detrás de este trabajo radica en la búsqueda de una solución que combine la precisión de la visión artificial con la eficacia de las librerías OpenCV y MediaPipe, proporcionando un enfoque práctico y de fácil implementación. La metodología adoptada se basa en la detección de características específicas de las manos, identificando gestos predefinidos que actúen como señales de emergencia, activando así un sistema de paro automático.

El método empleado implica la utilización de algoritmos de procesamiento de imágenes y técnicas de aprendizaje profundo para el reconocimiento de manos. La combinación de OpenCV y MediaPipe permite la detección precisa de gestos en tiempo real, ofreciendo una respuesta rápida y confiable en entornos críticos.

A pesar de las ventajas, este trabajo presenta limitaciones inherentes, como la dependencia de condiciones de iluminación y la necesidad de un entrenamiento continuo para adaptarse a diferentes gestos. La distribución en la estructura del trabajo aborda estas limitaciones mediante la implementación de secciones dedicadas a la discusión de desafíos y posibles mejoras.

En la estructura, se abordan temas relacionados con la introducción al problema, la justificación del enfoque elegido, la explicación detallada del método utilizado, las limitaciones identificadas y propuestas de mejoras, culminando con conclusiones que destacan la importancia y las implicaciones prácticas del sistema desarrollado.

1.2 Justificación

"La visión artificial, mediante el uso de tecnologías como OpenCV y MediaPipe, ofrece una poderosa herramienta para mejorar la seguridad y la respuesta ante emergencias en diversos entornos. En particular, la detección de manos en un área específica se vuelve crucial en situaciones críticas.

Imaginemos un entorno donde la precisión y la velocidad son esenciales, como una fábrica o un laboratorio. En este contexto, la detección de manos a través de la visión artificial puede ser implementada para identificar movimientos específicos que indiquen una situación de emergencia. Esto nos lleva a la necesidad de trabajar en este tema con el fin de desarrollar un sistema que pueda reconocer gestos específicos de las manos dentro de un área designada.

La razón principal para abordar este proyecto es mejorar la seguridad y la capacidad de respuesta en entornos críticos. Implementar un sistema que pueda identificar gestos de emergencia, como un paro de emergencia, mediante la detección de manos, permitirá una acción rápida y eficiente para prevenir o mitigar situaciones peligrosas.

El uso de OpenCV y MediaPipe se justifica por su eficacia y facilidad de implementación. Estas librerías ofrecen herramientas poderosas para el procesamiento de imágenes y la detección de características específicas, como las manos. Además, su popularidad y documentación extensa facilitan el desarrollo y mantenimiento del sistema.

El trabajar en la detección de manos con visión artificial utilizando OpenCV y MediaPipe se presenta como una iniciativa clave para mejorar la seguridad en entornos críticos, permitiendo una respuesta rápida y eficiente ante situaciones de emergencia, y contribuyendo así al bienestar y la protección de las personas involucradas en esos entornos."

1.3 Planteamiento del problema

En el campus universitario, especialmente en laboratorios y áreas de investigación, se observa un creciente flujo de actividad que involucra maquinaria especializada, sustancias químicas y equipos sensibles. La seguridad de los estudiantes, profesores y personal es de suma importancia, pero se ha identificado una brecha en la respuesta rápida ante situaciones de emergencia, como accidentes o la detección de condiciones peligrosas.

La propuesta busca implementar un sistema de detección de manos mediante visión artificial en áreas específicas de la universidad, utilizando las librerías OpenCV y MediaPipe. La necesidad surge de la importancia de contar con un mecanismo eficiente para activar un paro de emergencia en situaciones críticas. Este sistema permitirá identificar

gestos específicos de las manos que actúen como señales de alarma, asegurando una respuesta inmediata ante posibles riesgos.

La no aplicación de esta propuesta podría tener consecuencias significativas en términos de seguridad. La falta de un sistema de detección de manos en áreas específicas aumenta el riesgo de accidentes y la demora en la respuesta ante emergencias. La incapacidad para identificar gestos de advertencia podría resultar en daños a equipos costosos, exposición a sustancias peligrosas o lesiones personales. Además, la falta de un mecanismo rápido de paro de emergencia podría afectar la reputación de la institución en términos de seguridad y bienestar de la comunidad universitaria. En última instancia, la ausencia de esta propuesta podría comprometer la integridad y seguridad general en el entorno académico.

1.4 Objetivos

1.5.1 Objetivo general

Diseñar un sistema de paro de emergencia en una maquina dobladora utilizando visión artificial.

1.5.2 Objetivos específicos

- Comprender la estructura de un programa de visión artificial
- Diseñar un sistema de detección de extremidades mediante el uso de visión artificial
- Implementar la detección de extremidades cuando estén próximas a la máquina dobladora
- Probar el sistema de visión artificial.

1.5 Alcances y limitaciones

Este trabajo desarrolla un sistema práctico y aplicable en entornos industriales para monitorear la seguridad del operador en tiempo real. Utiliza tecnologías actuales como OpenCV y MediaPipe en Python, lo que realza la eficacia y precisión del sistema de monitoreo basado en visión artificial. El estudio se centra en un ambiente específico (dobladoras CNC), por lo que su aplicabilidad en otros entornos puede requerir ajustes y validación adicionales. La efectividad del sistema está sujeta a la calidad del hardware y las condiciones del entorno (iluminación, obstrucciones visuales, etc.), lo que puede limitar su desempeño.

Capítulo II.
Marco Teórico

Visión Artificial

En [1], dice que la visión artificial es un área emergente dentro de la inteligencia artificial (IA) que se dedica a dotar a las máquinas con la capacidad de interpretar y comprender el entorno visual de manera similar a como lo hacen los humanos. Su objetivo principal es permitir que las máquinas procesen, analicen y respondan a la información visual de forma autónoma, replicando el proceso de percepción visual que realizan los seres humanos. Esta tecnología integra varios campos interdisciplinarios, como la informática, la matemática, y la psicología cognitiva, para construir sistemas capaces de "ver" y entender el mundo visual.

Conceptos Clave de la Visión Artificial

Procesamiento de Imágenes

El procesamiento de imágenes es el primer paso en la visión artificial y se refiere a la manipulación y análisis de imágenes digitales para extraer información útil. Este proceso incluye:

- Preprocesamiento: Consiste en mejorar la calidad de la imagen mediante técnicas como la reducción de ruido, el ajuste de contraste y la corrección de iluminación. Estas técnicas son fundamentales para preparar la imagen para análisis posteriores.
- Segmentación: Separa una imagen en regiones o segmentos basados en características comunes como color, textura o intensidad. La segmentación ayuda a identificar áreas de interés dentro de la imagen, facilitando la extracción de información relevante.
- Extracción de Características: Identifica características clave en la imagen, como bordes, esquinas, y texturas. Estas características son utilizadas para describir y diferenciar objetos dentro de la imagen.
- Transformaciones: Incluyen técnicas como la transformación de Fourier o la transformada de Hough, que permiten analizar patrones y estructuras específicas en la imagen.

Reconocimiento de Patrones

El reconocimiento de patrones se enfoca en identificar y clasificar objetos o características dentro de una imagen. Este proceso incluye:

- Detección de Características: Identifica patrones o estructuras específicas en una imagen, como líneas, formas geométricas o texturas. Estas características son esenciales para la clasificación y reconocimiento de objetos.
- Clasificación: Asigna una etiqueta o categoría a los objetos detectados basándose en las características identificadas. Por ejemplo, clasificar una imagen de un animal como "gato" o "perro".
- Algoritmos de Reconocimiento: Utiliza técnicas como máquinas de vectores de soporte (SVM), redes neuronales convolucionales (CNN), y modelos de aprendizaje profundo para mejorar la precisión en el reconocimiento y clasificación de patrones.

Aprendizaje Automático

- El aprendizaje automático permite que los sistemas mejoren su desempeño con el tiempo a través de la experiencia y el análisis de datos. Los componentes clave incluyen:
- Modelos de Aprendizaje: Emplea algoritmos y modelos que permiten a las máquinas aprender a partir de datos de entrenamiento. Los modelos incluyen redes neuronales profundas, árboles de decisión, y modelos de regresión.
- Entrenamiento y Validación: Durante el entrenamiento, el modelo ajusta sus parámetros para mejorar la precisión en la tarea específica, como la clasificación de imágenes. La validación y prueba del modelo aseguran que el rendimiento sea consistente y generalizable a nuevos datos.
- Optimización: Mejora el desempeño del modelo mediante técnicas de optimización como el ajuste de hiperparámetros, regularización, y técnicas de aprendizaje por refuerzo.

Detección de Objetos

- La detección de objetos es una aplicación crucial dentro de la visión artificial que se centra en identificar y localizar objetos específicos en imágenes o videos. Este proceso incluye:
- Extracción de Características: Identifica características relevantes de los objetos, como bordes, texturas, y patrones únicos que ayudan en la detección y clasificación.
- Clasificación y Localización: Asigna una etiqueta a los objetos detectados y determina su ubicación precisa dentro de la imagen. Esto se realiza mediante técnicas como el uso de cajas delimitadoras y redes neuronales especializadas en detección de objetos.
- Algoritmos de Detección: Utiliza algoritmos avanzados como YOLO (You Only Look Once), Faster R-CNN, y SSD (Single Shot MultiBox Detector) que combinan la detección y clasificación en una sola etapa, permitiendo un análisis eficiente y en tiempo real.

Aplicaciones de la Visión Artificial

En [2] nos dice que la visión artificial tiene un amplio rango de aplicaciones en diversos sectores, entre los que se incluyen:

- Seguridad y Vigilancia: Identificación de comportamientos sospechosos y reconocimiento facial para mejorar la seguridad en espacios públicos y privados.
- Logística y Cadena de Suministro: Rastrear y clasificar productos en almacenes y cadenas de producción, optimizando la gestión del inventario y la eficiencia operativa.
- Medicina: Análisis de imágenes médicas como radiografías y resonancias magnéticas para diagnosticar enfermedades y planificar tratamientos.
- Agricultura: Monitoreo de cultivos y detección de plagas para mejorar la producción y gestión de recursos agrícolas.
- Automoción: Implementación en sistemas de asistencia al conductor y vehículos autónomos para mejorar la seguridad y la navegación.

La visión artificial continúa evolucionando con el desarrollo de nuevas técnicas y algoritmos, ampliando sus aplicaciones y mejorando la capacidad de las máquinas para interpretar el mundo visual de manera más precisa y eficaz.

Técnicas Comunes en la Detección de Objetos

En la visión artificial, la detección de objetos es una tarea esencial que nos permite a las máquinas identificar y localizar elementos específicos en imágenes o secuencias de video. Este proceso se basa en tres técnicas fundamentales: la extracción de características, la clasificación de objetos y la localización de objetos, cada una de las cuales juega un papel crucial en la interpretación visual de las máquinas.

La extracción de características es el primer paso en la detección de objetos. Este proceso implica identificar y destacar elementos relevantes dentro de una imagen que pueden ser utilizados para describir y diferenciar los objetos presentes. Entre los elementos clave que se extraen están los bordes, las esquinas y los patrones de textura. Los bordes son las transiciones abruptas en la intensidad de la imagen que marcan los límites de los objetos. Técnicas como el algoritmo de Canny o el de Sobel son empleadas para detectar estos bordes y resaltar las fronteras de los objetos en la imagen. Las esquinas, por otro lado, son puntos donde ocurre un cambio significativo en la dirección de los bordes, y su detección se realiza con métodos como el detector de esquinas de Harris o el de Shi-Tomasi. Estos puntos son cruciales para identificar características de interés. La textura y los patrones también son importantes, ya que proporcionan información sobre la apariencia superficial de los objetos, ayudando a distinguir entre diferentes tipos de superficies.

Una vez que se han extraído las características relevantes, el siguiente paso es la **clasificación de objetos**. Esta técnica consiste en asignar etiquetas o categorías a los objetos detectados basándose en las características extraídas. La clasificación se realiza mediante algoritmos que analizan las características para identificar el tipo de objeto en la imagen. Algoritmos como los clasificadores de máquinas de vectores de soporte (SVM) y los árboles de decisión son utilizados para esta tarea. Estos métodos requieren un conjunto de datos de entrenamiento con ejemplos etiquetados para aprender a distinguir entre diferentes clases de objetos. Las redes neuronales convolucionales (CNN), un tipo de red neuronal profunda, han transformado este proceso al permitir que las máquinas aprendan representaciones jerárquicas de las características en las imágenes. Las CNN son especialmente efectivas en manejar variaciones en la apariencia de los objetos, como cambios en la iluminación, la escala y la orientación.

La **localización de objetos** es la técnica que permite determinar la posición precisa de los objetos dentro de una imagen. Este proceso no solo identifica qué objetos están presentes, sino también dónde se encuentran. Una técnica común para la localización es el uso de cajas delimitadoras, que son rectángulos que rodean los objetos detectados. Estas cajas se definen por sus coordenadas, marcando la ubicación de los objetos en la imagen. Para una localización más detallada, se emplea la segmentación de imágenes, que divide la imagen en regiones basadas en características específicas. La segmentación semántica y la segmentación de instancias son métodos avanzados que etiquetan cada píxel de la imagen, proporcionando una descripción precisa de la forma y la posición de los objetos. Algoritmos modernos como YOLO (You Only Look Once) y Faster R-CNN combinan la detección y la localización en una sola etapa, mejorando la eficiencia y permitiendo la detección en tiempo real.

En conjunto, la extracción de características, la clasificación de objetos y la localización de objetos son técnicas interrelacionadas que permiten a las máquinas interpretar el contenido visual con gran precisión. Cada técnica contribuye a un aspecto específico del proceso de detección, facilitando aplicaciones en una amplia variedad de sectores, desde la seguridad y vigilancia hasta la logística y la medicina.



Figura 1 Detección de vehículos y señalamientos. [3]

Aplicaciones de la Visión Artificial en Diversos Sectores

La visión artificial ha demostrado ser una herramienta versátil y poderosa en una amplia gama de industrias, aportando mejoras significativas en la eficiencia, precisión y seguridad. A continuación, se exploran algunas de las aplicaciones más destacadas en distintos ámbitos:

En el ámbito de la vigilancia y la seguridad, la visión artificial juega un papel crucial en la detección de actividades sospechosas y la identificación de intrusiones. Los sistemas de cámaras equipadas con tecnologías de visión artificial pueden analizar el video en tiempo real para identificar comportamientos anómalos, como movimientos inusuales o la presencia de personas en áreas restringidas. Las capacidades avanzadas de análisis permiten la detección automática de eventos críticos, como el ingreso no autorizado o comportamientos potencialmente peligrosos, y envían alertas inmediatas a los operadores. Además, los sistemas de reconocimiento facial pueden utilizarse para autenticar identidades y gestionar el acceso a instalaciones, mejorando la seguridad general de los entornos vigilados.

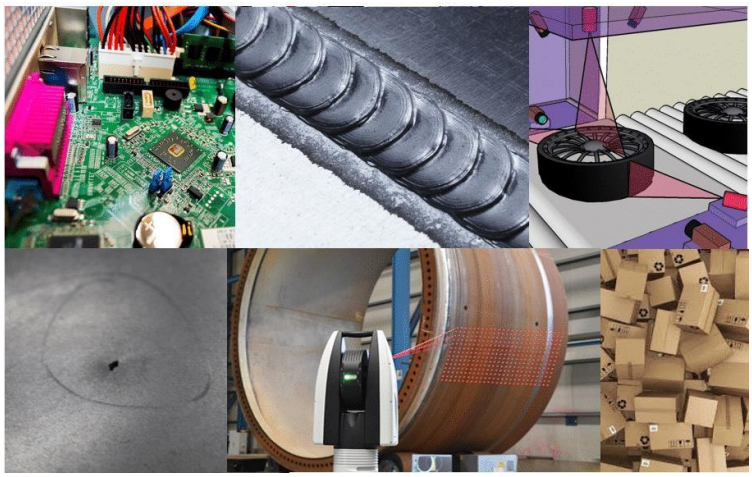

Figura 2 Aplicaciones de la visión artificial en la industria. [4]

En el ámbito logístico y de la cadena de suministro, la visión artificial facilita el rastreo y la clasificación de productos de manera eficiente. Los sistemas de visión artificial pueden automatizar el proceso de inspección de productos en las líneas de producción, identificando defectos y asegurando la calidad antes de que los productos lleguen al mercado. Además, en los almacenes, la tecnología se utiliza para el reconocimiento y la clasificación automática de mercancías, optimizando la gestión del inventario y reduciendo los errores humanos. Esto mejora la precisión en el procesamiento de pedidos y en el reabastecimiento, lo que resulta en una cadena de suministro más ágil y eficiente.

En el sector médico, la visión artificial tiene un impacto significativo en el diagnóstico y el tratamiento de enfermedades. Los sistemas de visión artificial pueden analizar imágenes médicas, como radiografías, resonancias magnéticas y tomografías computarizadas, para detectar anomalías y enfermedades con alta precisión. Por ejemplo, la tecnología se emplea en la detección temprana de cáncer mediante la identificación de patrones en imágenes de tejidos, así como en el análisis de imágenes de retina para detectar enfermedades oculares. Además, la visión artificial ayuda en la planificación quirúrgica y en el monitoreo postoperatorio, proporcionando herramientas avanzadas para los profesionales de la salud.

En la agricultura, la visión artificial se utiliza para monitorear la salud de los cultivos y gestionar el rendimiento agrícola. Los sistemas de visión artificial pueden analizar imágenes aéreas capturadas por drones para detectar plagas, enfermedades y deficiencias nutricionales en los cultivos. Esta información permite a los agricultores tomar decisiones informadas sobre el uso de pesticidas, fertilizantes y otros tratamientos, optimizando el rendimiento y reduciendo el impacto ambiental. La tecnología también se emplea en la cosecha automatizada, identificando frutas y verduras maduras para su recolección, lo que mejora la eficiencia y reduce el desperdicio.

En el sector del retail, la visión artificial mejora la experiencia del cliente y optimiza las operaciones en el punto de venta. Los sistemas de visión artificial se utilizan para el reconocimiento de productos y la gestión de inventarios en tiempo real. Además, las cámaras equipadas con visión artificial pueden analizar el comportamiento de los clientes

en las tiendas, proporcionando datos sobre patrones de compra y ayudando a personalizar las ofertas y promociones. La tecnología también se emplea en el monitoreo de las estanterías para asegurar que los productos estén correctamente almacenados y disponibles para los clientes.

La visión artificial encuentra aplicaciones en una variedad de otros sectores, incluyendo la manufactura, donde se utiliza para la inspección de calidad y el control de procesos; el transporte, para el desarrollo de vehículos autónomos y sistemas de asistencia al conductor; y el entretenimiento, donde mejora las experiencias de realidad aumentada y virtual.

La visión artificial está transformando numerosos sectores al ofrecer soluciones innovadoras que mejoran la eficiencia, la precisión y la seguridad. Su capacidad para analizar y comprender el contenido visual abre nuevas oportunidades para la automatización y el avance tecnológico en diversos campos.



Figura 3 Detección de rostros. [5]

Para llevar a cabo la detección de objetos, se recurre a algoritmos especializados que analizan patrones visuales y características con el fin de identificar y ubicar objetos dentro de una imagen o secuencia de video. Estos algoritmos pueden basarse en técnicas como las redes neuronales convolucionales (CNN) y se entrenan con conjuntos de datos específicos para mejorar su precisión en la identificación de objetos particulares.

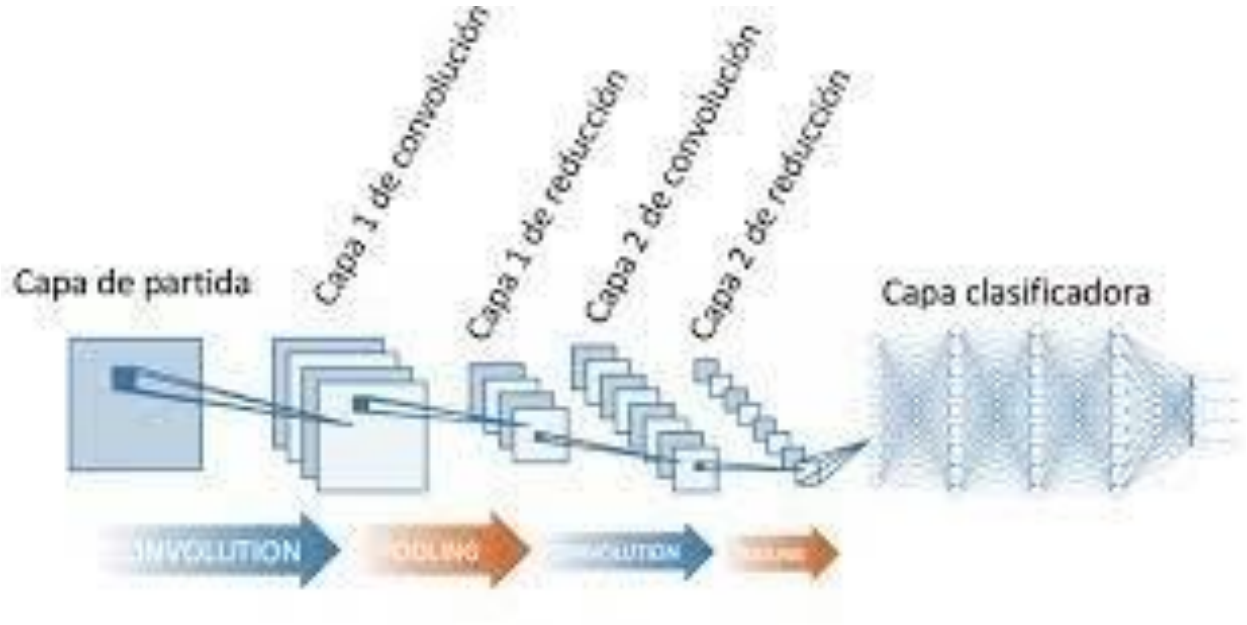

Figura 4 Red neuronal convolucionales. [6]

La detección de objetos capacita a las máquinas para reconocer y localizar objetos en entornos visuales, y su aplicación abarca una amplia gama de sectores, como la automoción, la seguridad, la logística y la medicina, entre otros. Esta tarea se lleva a cabo mediante algoritmos especializados que analizan imágenes y videos, lo que mejora la eficiencia y la automatización en diversas aplicaciones.

21 Hands Landmark.

[7] nos demuestra cómo nos permite detectar manos, identificándolas como izquierda o derecha. Pero además nos proporciona 21 hand landmarks o puntos de referencia con la ubicación de cada uno de los dedos, así como la palma de la mano.



Figura 5 Deteccion de la palma de la mano [8].

Cada punto de la mano reflejada en la imagen tiene un nombre en específico lo cual con ayuda de mediapipe se hace una detección.

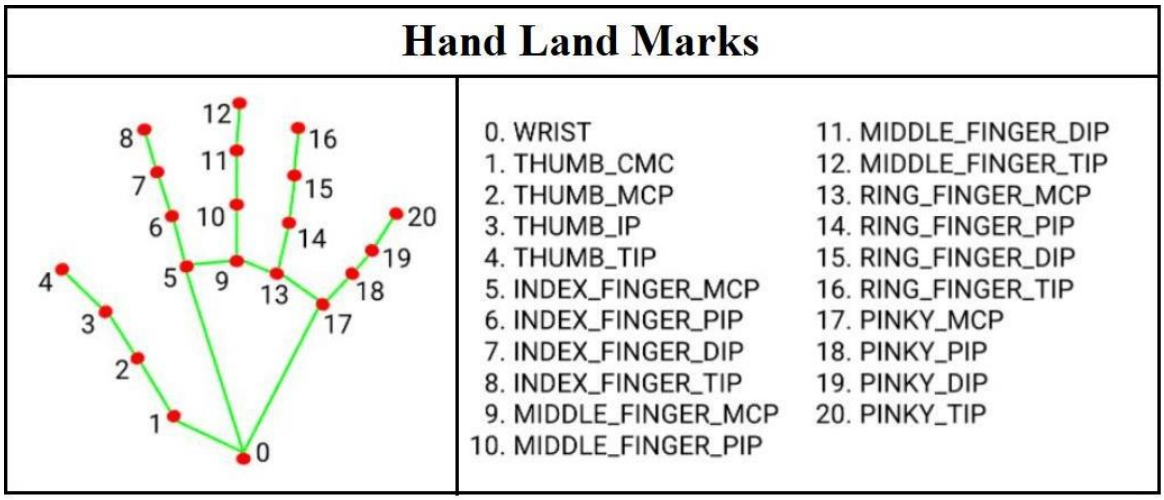

Figura 6 Puntos de referencia de la mano. [7]

Aplicaciones de la visión artificial industrial.

Las aplicaciones de los softwares de análisis de imagen son más comunes en el ámbito industrial y científico, aunque su potencial demuestra que estos usos se pueden extender a otras actividades, sobre todo gracias a las facilidades de la automatización.

En el entorno industrial, suelen ser empleadas para la captura de imágenes con el propósito de realizar mediciones con mucha exactitud y, a partir de esto, determinar si un producto es apto, si no lo es, generar la salida de la línea, o si es necesario corregir los parámetros de producción para evitar más defectos.

Por parte, en el entorno científico, el motivo por el que se opta por sistemas de captura de imagen inteligentes suele ser con el objetivo de realizar procesos de mejora y optimización. Estas metas también se logran gracias a tareas automatizadas de medición, detección de errores y ejecución de mejoras.

En ambos casos, pero sobre todo en el campo científico, estos softwares facilitan el almacenamiento y procesamiento de datos, lo que permite obtener informes gráficos y estadísticos muy al detalle.

También se han desarrollado numerosos métodos de aprendizaje automático con diferentes fines, pero todos con un mismo objetivo: permitir a las máquinas a aprender por sí solas de los errores y la información de la que disponen. Mediante algoritmos pueden identificar estructuras y patrones en los datos de los que aprender y así evitar futuros errores.

No obstante, se requiere una gran cantidad de datos para entrenar los sofisticados algoritmos de Deep Learning a fin de que lleven a cabo tareas específicas de resolución de problemas, realizando análisis comparativos de forma automatizada con respecto a imágenes suministradas.

Integrar esta tecnología con alta capacidad de análisis y toma de decisiones en ámbitos donde la investigación e innovación es clave para la adaptabilidad de las actividades a las nuevas necesidades de la sociedad, permite que la Industria 4.0 y la automatización avancen imparables.

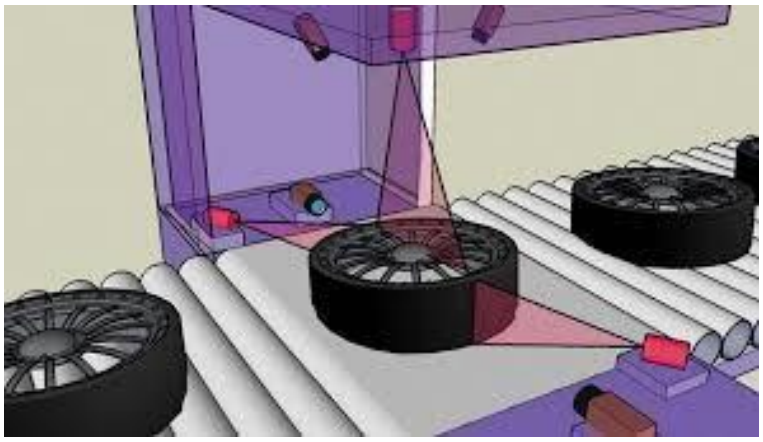

Figura 7 Visión Artificial en la Industria. [4]

Aunque ya hablemos de una tecnología muy consolidada en entornos industriales y científicos, la creación y aplicación de estos sistemas de visión avanzados no cuenta con más de 30 años de vida, y han supuesto un cambio notable en las técnicas de medición e inspección usados en plantas productivas.

Esta tecnología puede existir porque los sistemas informáticos también soportan más funciones y, en especial, han aumentado su potencia de cálculo y procesamiento. Por tanto, el ancho de banda también es otra parte importante para que estos softwares se puedan aplicar. Lo interesante es que todo este proceso de captación, analítica y procesamiento de datos se pueda producir en tiempo real y, junto a este, que la propia máquina automatizada tome decisiones sobre la línea de montaje.

Esto evidencia que uno de los valores claves de esta tecnología de análisis de la imagen es que es capaz de realizar tareas claves en el ámbito industrial en un corto espacio de tiempo. De esta manera, la cadena de producción sigue su curso y la programación de los sistemas, según parámetros establecidos, permite que se tomen decisiones rápidas y se actúe en el mismo momento, ya sea retirando un producto o corrigiendo tareas programadas.

A pesar de que la tecnología de análisis de imagen ya está bien establecida en entornos industriales y científicos, su evolución y aplicación efectiva no tienen más de 30 años. Esta tecnología ha transformado significativamente las técnicas de medición e inspección en plantas productivas y otros entornos críticos.

La capacidad para implementar sistemas avanzados de visión artificial ha sido facilitada por el notable aumento en la potencia de cálculo y procesamiento de los sistemas informáticos. Los avances en hardware, como los procesadores más rápidos y las tarjetas gráficas dedicadas, han permitido el manejo de grandes volúmenes de datos y la ejecución de algoritmos complejos en tiempo real. Además, la expansión del ancho de banda y la mejora en la infraestructura de redes han hecho posible la transmisión y el análisis de datos a alta velocidad, lo cual es crucial para aplicaciones que requieren procesamiento en tiempo real.

Uno de los aspectos más destacables de la tecnología de análisis de imagen es su capacidad para operar en tiempo real. Esto significa que la captura, análisis y procesamiento de imágenes ocurre de manera instantánea, permitiendo que las máquinas automatizadas tomen decisiones sobre la línea de montaje sin retrasos. La capacidad de ejecutar algoritmos de visión artificial en tiempo real permite que los sistemas detecten anomalías, errores o condiciones críticas al momento en que ocurren, lo que resulta en una rápida respuesta y corrección de problemas. El valor clave de la tecnología de análisis de imagen radica en su capacidad para realizar tareas críticas con alta precisión y en un corto período de tiempo. En entornos industriales, esto se traduce en una mejora en la eficiencia y la calidad de la producción. La programación de estos sistemas basada en parámetros predefinidos permite que las decisiones rápidas se tomen automáticamente, como la retirada de productos defectuosos o la corrección de procesos en la línea de montaje. Esto no solo optimiza la cadena de producción, sino que también minimiza el riesgo de errores humanos y aumenta la seguridad operativa.

Análisis de imagen.

En el contexto de sistemas de detección de manos para activar paros de emergencia, como el que estás desarrollando, el software de análisis de imagen juega un papel crucial. La capacidad de reconocer gestos específicos en tiempo real y activar mecanismos de seguridad es un ejemplo de cómo estas tecnologías pueden mejorar la seguridad en entornos críticos. La integración de herramientas como OpenCV y MediaPipe en tu sistema de visión artificial demuestra cómo los avances en análisis de imagen pueden ser aplicados para crear soluciones efectivas y automatizadas que resuelvan problemas específicos en industrias y laboratorios.

Niveles de software de visión

La razón de estos niveles es que cada uno de estos softwares requiere de diferentes grados de programación y conocimiento sobre el manejo de sistemas de visión:

- Sistemas de programación a bajo nivel basado en kits de desarrollo de software (SDK): Emplean librerías en DLL o Active X y requieren de un amplio conocimiento de hardware y software de visión y un conocimiento de programación en lenguajes estándar, como Visual C o Visual Basic.
- Sistemas de programación por menú: Son los sistemas más sencillos y no es necesario contar con competencias relacionadas con la programación.
- Aplicaciones específicas: Son programas diseñados para una solución concreta.

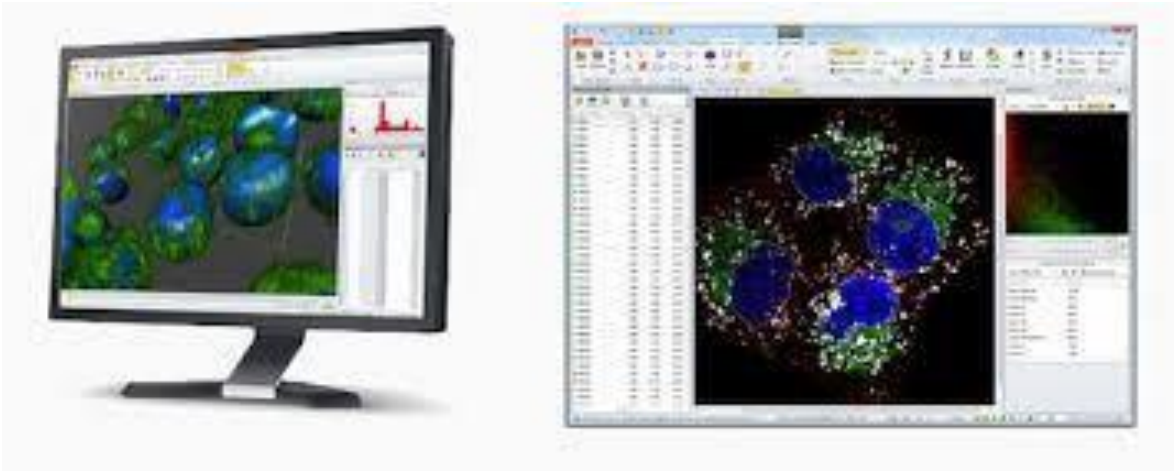

Figura 8 Software de análisis de imágenes. [9]

El software de visión artificial se ha convertido en una herramienta esencial en diversos entornos industriales y científicos, transformando significativamente las técnicas de medición e inspección. La evolución de esta tecnología ha sido posible gracias a los

avances en la potencia de cálculo y el procesamiento de datos, permitiendo la operación en tiempo real y la automatización de decisiones en líneas de producción.

En el ámbito del software de visión, existen diferentes niveles que se adaptan a las variadas necesidades y habilidades técnicas de los usuarios. Uno de los niveles más avanzados son los sistemas de programación a bajo nivel, basados en kits de desarrollo de software (SDK). Estos sistemas utilizan librerías y componentes como DLL (Dynamic Link Libraries) o controles ActiveX, y requieren un profundo conocimiento tanto del hardware de visión como de la programación en lenguajes como Visual C++ o Visual Basic. La principal ventaja de estos sistemas es su alta personalización, permitiendo un control exhaustivo sobre los parámetros y el procesamiento de imágenes, lo que es ideal para aplicaciones complejas que requieren ajustes finos y soluciones a medida.

Por otro lado, los sistemas de programación por menú están diseñados para ser más accesibles y fáciles de usar. Estos sistemas proporcionan interfaces gráficas de usuario (GUI) que permiten configurar y ajustar parámetros mediante la selección de opciones predefinidas, sin necesidad de conocimientos avanzados en programación. Son ideales para aplicaciones estándar y entornos donde la implementación rápida y efectiva es prioritaria, facilitando la operación a usuarios que no poseen un profundo entendimiento técnico.

Finalmente, las aplicaciones específicas están desarrolladas para resolver problemas concretos y se adaptan a necesidades particulares dentro de una industria o dominio específico. Aunque la complejidad de estas aplicaciones puede variar, suelen estar diseñadas para ser utilizadas por personas con distintos niveles de conocimientos técnicos, proporcionando soluciones optimizadas para casos de uso específicos. Esto permite una mayor eficiencia y eficacia en la resolución de problemas concretos, adaptando el software a las necesidades particulares de cada situación.

La diversidad en los niveles de software de visión refleja la variedad de aplicaciones y capacidades necesarias para abordar diferentes desafíos en la visión artificial. Desde soluciones altamente personalizables hasta opciones accesibles y específicas, cada nivel ofrece herramientas adecuadas para mejorar la eficiencia y precisión en entornos industriales y científicos.

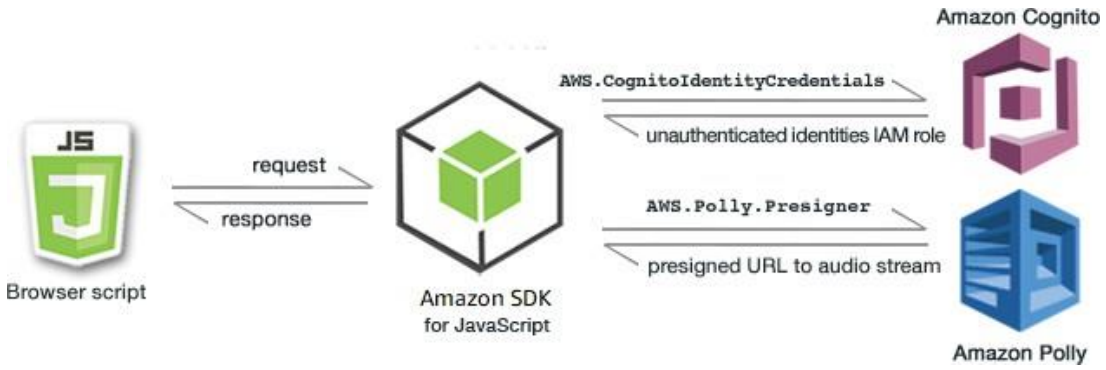

Figura 9 Estructura de SDK [10]

Lenguaje de programación.

Un lenguaje de programación es una forma de comunicarnos con una computadora, tablet o celular e indicarle qué queremos hacer.

Existen distintos tipos de lenguaje: principalmente de bajo nivel y de alto nivel. La diferencia se encuentra en lo cerca o lejos que estemos del hardware de nuestro equipo. Esta cercanía tiene que ver con el control que tengamos sobre el dispositivo, placa o controlador.

Encontramos diferentes lenguajes como C, C++, Java, PHP, Python, C#, ASP, entre otros.

Los programas habitualmente se diseñan o escriben a través de un entorno de desarrollo integrado (IDE por sus siglas en ingles). Un IDE habitualmente cuenta con un editor de texto (donde escribiremos nuestro código), el compilador y la depuradora para el lenguaje que estemos utilizando y la correspondiente plataforma en la que luego correrá. Muchas veces cuenta con un constructor de interfaz gráfica (en ingles GUI).

Un lenguaje de programación es una herramienta fundamental para comunicarnos con una computadora, tablet o celular, permitiéndonos indicarle exactamente qué queremos que haga. En el desarrollo de sistemas de visión artificial, elegir el lenguaje adecuado es crucial, ya que influye directamente en el control que se puede ejercer sobre el hardware y la eficiencia del procesamiento de imágenes.

Existen principalmente dos tipos de lenguajes de programación. Los lenguajes de bajo nivel, como C y C++, están más cerca del hardware y proporcionan un control más preciso sobre los recursos del sistema. Estos lenguajes son ideales para aplicaciones que requieren un alto rendimiento y optimización, como el procesamiento intensivo de imágenes en sistemas de visión artificial. Por ejemplo, cuando se necesita una rápida manipulación de imágenes o una integración directa con hardware específico, los lenguajes de bajo nivel permiten ajustar el código para maximizar la eficiencia.

Por otro lado, los lenguajes de alto nivel, como Python, Java y C#, están más alejados del hardware y facilitan el desarrollo mediante abstracciones que simplifican la programación. En el contexto de visión artificial, estos lenguajes son a menudo preferidos debido a su capacidad para integrar rápidamente bibliotecas y frameworks como OpenCV y MediaPipe, que simplifican el trabajo con imágenes y visión por computadora. Esto permite desarrollar y ajustar aplicaciones de visión artificial de manera más rápida, sin tener que lidiar con la complejidad del hardware directamente.

Para escribir y gestionar el código, los desarrolladores utilizan un entorno de desarrollo integrado (IDE). Un IDE proporciona herramientas esenciales como un editor de texto para escribir el código fuente, un compilador para traducir el código en un formato ejecutable, y una depuradora para identificar y corregir errores. Además, muchos IDEs incluyen un constructor de interfaz gráfica (GUI) que facilita la creación de interfaces visuales para el usuario. En sistemas de visión artificial, un IDE con estas características puede ser muy útil

para diseñar interfaces que muestren en tiempo real el análisis de imágenes y permitan una configuración intuitiva del sistema.

La elección del lenguaje de programación y el entorno de desarrollo adecuado es esencial para el éxito en el desarrollo de sistemas de visión artificial. Los lenguajes de bajo nivel ofrecen un control detallado y una optimización de rendimiento, mientras que los lenguajes de alto nivel proporcionan herramientas y librerías que simplifican el desarrollo. Un IDE bien equipado apoya en la escritura, depuración y ejecución del código, facilitando el proceso de desarrollo y la integración de funcionalidades complejas en aplicaciones de visión artificial.

Python.

Características de Python

Python un lenguaje multiparadigma, que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es interpretado, de tipado dinámico y multiplataforma.

Lo básico de Python

La sintaxis de Python es extremadamente "limpia", no se requiere de ningún carácter que indique el final de una secuencia y los bloques se definen a través de la identificación del código. Los comentarios de una sola línea se definen con el carácter almohadilla (#) y las cadenas de múltiples líneas (""") se suelen emplear para escribir comentarios multilínea.

Los operadores aritméticos +, - y / significan en Python lo mismo que en matemáticas. El asterisco (*) es el símbolo para la multiplicación, el % se usa para obtener el módulo, // para división entera y ** es el símbolo para las potencias. También disponemos de los operadores incrementar (+=) y decrementar (-=).

Existen tres operadores lógicos: and, or, y not. La semántica (significado) de estos operadores es similar a sus significados en inglés (en español “y”, “o” y “no”).

La asignación de valores, como en la mayoría de los lenguajes, se realiza con el símbolo "igual" (=). La doble igualdad (==) se usa para comprobar que dos valores son iguales, además disponemos de estos operadores de comparación: ¡distinto que (!=), mayor que (>), menor que (<), mayor o igual (>=) y menor o igual (<=).

Tipos de datos

Los tipos de datos básicos en Python son los numéricos formados por enteros, los reales y los complejos; las cadenas de texto y los booleanos. La función type nos devuelve el tipo:

Estructuras de control de flujo

Las sentencias de control de flujo son bloques de código en los que se agrupan instrucciones de manera controlada. Por un lado, tenemos las estructuras condicionales (if, if .. else, if ..elif ..else) y por otro las estructuras iterativas (for, while).

Funciones

Una función no es más que un bloque de código reutilizable encargado de realizar una determinada tarea. Para definir una función en Python debemos utilizar la palabra reservada "def" seguido del nombre de la función y los parámetros los indicamos entre parentesis. Veamos varios ejemplos:

Funciones, parámetros *args y **kwargs

En el ejemplo anterior hemos visto varias formas de definir y ejecutar a las funciones en Python, ya sea con parámetros por con valores por defecto o keywords como argumentos. Nos faltaría ver como llamarlas con colecciones como argumentos, una forma muy común de pasar valores a las funciones en Python, y que en muchas ocasiones cuesta entender. Para ello debemos tener claro que la tupla como colección de argumentos se define con *args y el diccionario con **kwargs. Veamos un ejemplo en la REPL:

Manipulación de cadenas

Las cadenas en Python pueden definirse entre comillas simples o dobles. En la REPL se muestra un ejemplo de la mayoría de las operaciones que podemos realizar ellas:

Estructuras de datos (colecciones)

Las colecciones son un tipo de datos diseñados específicamente para agrupar objetos y llevar a cabo tareas con ellos. Las estructuras de datos más utilizadas en Python son las listas, tuplas (listas inmutables) y los diccionarios, aunque personalmente también suelo usar mucho los conjuntos (sets).

No todas las estructuras de datos exponen exactamente las mismas operaciones, por ejemplo, una lista no deja buscar por clave (si bien sí por índice) y un conjunto no deja buscar ni por una cosa ni por otra.

El siguiente playground está dividido en tres ficheros, main.py desde el que se importa listas.py y diccionarios.py, para ver los resultados de uno u otro, elimina o comenta su respectivo import.

Clases y objetos

En programación orientada a objetos (POO), un objeto es una entidad que agrupa un estado y una funcionalidad relacionada. El estado se define a través de las variables denominadas atributos y la funcionalidad a través de funciones denominadas en POO cómo métodos. Por otro lado, una clase, no es más que una plantilla genérica a partir de la cuál instanciamos los objetos. Dicho de otra manera, una clase es una abstracción en la que se define el comportamiento que va a tener el objeto.

En Python, las clases se declaran mediante la palabra reservada class seguida del nombre de la clase, la clase base de la cual hereda, si no extiende de ninguna debe hacerlo de objeto; a continuación, dos puntos (:), luego el inventado y el cuerpo de la clase.

El método constructor de la clase se define con la palabra clave __init__y al igual que el resto de los métodos definidos en la clase recibe como primer parámetro self, el interprete de python requiere de este argumento para referenciarlos como métodos de instancia.

Pylance.

Pylance es un servidor de lenguaje para soportar la extensión principal de Python para Visual Studio Code. Su objetivo es ayudar a los desarrolladores a escribir código Python mejor y más rápido al proporcionar importaciones automáticas, resaltado semántico, informes integrados de advertencias (warnings) y errores, chequeo estricto de tipos de datos y firmas de métodos con información de tipo cuando se muestra la documentación emergente de Intellisense.

Pylance se basa en el servidor Microsoft Language Server Protocol (LSP) y ofrece funciones avanzadas de análisis estático y sugerencias inteligentes para el código Python. Esta extensión utiliza el motor Pyright, que es un analizador estático rápido y preciso para Python, lo que contribuye a una detección de errores más eficiente y a un soporte mejorado para la inferencia de tipos.

Entre las características clave de Pylance se incluyen:

Autocompletado avanzado: Proporciona sugerencias contextuales y precisas mientras escribes código, lo que facilita la escritura de código más rápido y sin errores.

Análisis estático: Realiza un análisis estático del código Python para identificar posibles errores y ofrecer recomendaciones, mejorando la calidad y la robustez del código.

Integración con el entorno de desarrollo: Se integra directamente con Visual Studio Code, aprovechando las funcionalidades del editor para proporcionar una experiencia de desarrollo más completa.

Soporte para tipos: Ofrece un mejor soporte para la inferencia de tipos en Python, ayudando a comprender y mantener el código de manera más efectiva.

La programación con la extensión Pylance se aplica principalmente en proyectos que utilizan el lenguaje de programación Python y que emplean el entorno de desarrollo Visual Studio Code. Su objetivo es mejorar la productividad y la calidad del código al proporcionar herramientas avanzadas de análisis y sugerencias

durante el proceso de desarrollo. Esto beneficia a los desarrolladores al permitirles escribir código más eficiente, detectar errores de manera temprana y mantener una base de código más sólida.

Las extensiones de C++ pueden incluir nuevas bibliotecas, macros, o características sintácticas que no están presentes en la versión estándar del lenguaje. Estas extensiones son a menudo específicas de un compilador o entorno de desarrollo particular, lo que significa que pueden no ser compatibles en todos los sistemas.

La programación de la extensión de C++ se utiliza en una variedad de aplicaciones y contextos, como el desarrollo de software de sistemas, aplicaciones de bajo nivel, y en entornos específicos donde se requiere un control más preciso sobre el hardware o el rendimiento.

Pylance ayuda a mantener la precisión en el desarrollo de sistemas de visión artificial al proporcionar un chequeo estricto de tipos de datos. Esto es especialmente importante en proyectos que involucran procesamiento de imágenes, donde la precisión en el manejo de datos y la correcta implementación de algoritmos son esenciales. Los errores de tipo pueden llevar a fallos en la ejecución del programa o resultados incorrectos en el análisis de imágenes. Con Pylance, puedes detectar y corregir estos errores antes de ejecutar el código, mejorando la robustez del sistema.

En términos prácticos, los programadores pueden emplear extensiones de C++ para optimizar el código, acceder a características avanzadas, o adaptarse a requisitos particulares de la plataforma.

Extensiones opcionales para un rendimiento más certero.

Serial Monitor.

La programación de la extensión Serial Monitor se refiere a la escritura de código en un entorno de desarrollo para utilizar y controlar el Serial Monitor, una herramienta comúnmente presente en entornos de desarrollo integrado (IDE) para microcontroladores y placas de desarrollo. El Serial Monitor es una interfaz que permite la comunicación serie entre un dispositivo microcontrolador y una computadora a través de un puerto serie, generalmente a través de un cable USB.

Para qué sirve:

El Serial Monitor es una herramienta esencial en el desarrollo de sistemas embebidos y programación de microcontroladores.

Permite enviar datos desde el microcontrolador a la computadora para su visualización en tiempo real.

Facilita la depuración y el análisis de datos al permitir la visualización de información, mensajes de depuración y resultados de variables durante la ejecución del programa.

Donde se aplica:

La programación de la extensión Serial Monitor se aplica comúnmente en entornos de desarrollo para microcontroladores como Arduino, Raspberry Pi y otras placas de desarrollo.

Se utiliza en proyectos que requieren comunicación entre el microcontrolador y la computadora para monitorear el estado del sistema, depurar problemas, o simplemente para visualizar datos relevantes.

La programación de la extensión Serial Monitor es esencial en el desarrollo de sistemas embebidos, ya que proporciona una herramienta de comunicación bidireccional entre el microcontrolador y la computadora, permitiendo la visualización y análisis en tiempo real de datos relevantes para facilitar el desarrollo y la depuración de proyectos.

Librerías.

Opencv

La Librería OpenCV se utiliza comúnmente en aplicaciones de visión por computadora y procesamiento de imágenes. En tercera persona, podríamos decir que OpenCV es una herramienta esencial para desarrolladores y científicos de datos que trabajan en proyectos que involucran análisis de imágenes y videos. Se emplea cuando se necesita realizar tareas como reconocimiento facial, seguimiento de objetos, detección de bordes, y otras operaciones de procesamiento de imágenes. Se utiliza en una variedad de campos, desde la investigación académica hasta el desarrollo de aplicaciones industriales.

La biblioteca OpenCV es apreciada por su eficacia y versatilidad en el procesamiento de imágenes en tiempo real, lo que la hace valiosa en aplicaciones como sistemas de vigilancia, vehículos autónomos, realidad aumentada, entre otras. Los desarrolladores confían en OpenCV debido a su amplia gama de funciones y algoritmos listos para usar, que facilitan la implementación de soluciones complejas de visión por computadora.

En cuanto a cuándo se usa, depende del contexto del proyecto. OpenCV es ideal cuando se requiere manipulación de imágenes y análisis visual, ya sea para extraer información útil, realizar seguimiento de objetos en movimiento o cualquier tarea relacionada con la visión por computadora.

El dónde sería en entornos de desarrollo de software, laboratorios de investigación, empresas de tecnología y en prácticamente cualquier lugar donde se necesite trabajar con imágenes o videos desde una perspectiva informática.

El por qué radica en la necesidad de contar con herramientas eficientes y robustas para el procesamiento de imágenes en proyectos específicos. OpenCV ofrece una plataforma sólida y ampliamente utilizada para abordar desafíos relacionados con la visión por computadora, lo que justifica su popularidad y adopción en la comunidad de desarrollo.

En este caso el uso que le daremos a la librería de Open Cv nos va a proporciona un marco de trabajo de alto nivel para el desarrollo de aplicación con visión por computador en tiempo real.

Mediapipe.

Mediapipe= La librería Mediapipe se utiliza comúnmente en aplicaciones de visión por computadora para procesar datos de medios, como imágenes y videos. En tercera persona, podría decirse que se emplea cuando una persona o un desarrollador busca implementar características avanzadas de análisis y seguimiento de medios en sus proyectos.

Por ejemplo, Mediapipe se utiliza cuando alguien necesita realizar el seguimiento de gestos o movimientos en tiempo real, como el seguimiento facial, el reconocimiento de manos, o incluso el seguimiento de objetos. Se selecciona Mediapipe como herramienta cuando se busca una solución eficiente y de alta calidad para aplicaciones que requieren el procesamiento de datos visuales.

En cuanto a cuándo se utiliza, podría decirse que Mediapipe se emplea durante la fase de desarrollo de un proyecto que requiere análisis de medios en tiempo real. Es especialmente útil en casos donde se necesitan datos precisos y rápidos para la toma de decisiones en la aplicación.

La librería Mediapipe se usa en diversas plataformas y contextos, desde aplicaciones de realidad aumentada y virtual hasta sistemas de interacción humano- máquina. Su utilización se extiende a situaciones donde se busca mejorar la experiencia del usuario a través de la detección y el análisis de elementos visuales.

En cuanto a dónde se usa, Mediapipe puede implementarse en una variedad de entornos, desde aplicaciones móviles y web hasta proyectos de escritorio. La versatilidad de la librería permite su integración en diferentes plataformas, lo que la hace adecuada para una amplia gama de aplicaciones.

En relación con el por qué se utiliza, Mediapipe ofrece un conjunto robusto de herramientas y algoritmos que simplifican la tarea de procesar y analizar datos visuales de manera eficiente. Su adopción se basa en la necesidad de contar con una solución confiable y de alto rendimiento para aplicaciones que requieren procesamiento de medios avanzado.

Seguridad industrial en maquinaria.

La seguridad industrial comprende el conjunto de regulaciones cuya pretensión es reducir los peligros en la empresa y prevenir, controlar o mitigar sus consecuencias. Como su nombre lo indica, se aplica en entornos industriales como plantas o fábricas.

También se refiere al área dentro de las empresas que aborda de manera sistemática la prevención de lesiones o accidentes laborales. Aplica especialmente en espacios industriales.

Su mayor utilidad consiste en mitigar los riesgos, previendo la ocurrencia de daños a personas o instalaciones. De esa manera, actúa como instancia preventiva más que reactiva. Cubre también el manejo e investigación de accidentes e incidentes, en caso de llegar a presentarse.

La mirada más actual del área está marcada por lo establecido en normas como la ISO 45001 y la 14001. La primera está referida a la Seguridad y Salud en el trabajo y la segunda atiende al medio ambiente.

En tal sentido, la seguridad en las instalaciones industriales persigue la preservación de la vida y la salud. Además de tener en cuenta factores como el orden y limpieza, atiende a elementos psicosociales, procurando el bienestar integral de los colaboradores.

La seguridad industrial es un campo crucial en el entorno de las plantas y fábricas, cuyo objetivo es reducir los peligros y prevenir, controlar o mitigar las consecuencias de posibles incidentes. Este conjunto de regulaciones y prácticas se enfoca en proteger a los trabajadores y las instalaciones mediante la implementación de medidas preventivas y el cumplimiento de normas específicas.

En el contexto de la seguridad industrial, se considera esencial la prevención de lesiones o accidentes laborales. Esto implica una evaluación continua de los riesgos asociados con el uso de maquinaria y la implementación de controles para minimizar esos riesgos. La seguridad en maquinaria no solo previene daños directos a las personas, sino que también protege las instalaciones y los equipos de posibles fallos o malfuncionamientos.

La seguridad industrial no se limita a la prevención reactiva de accidentes, sino que busca actuar de manera proactiva para identificar y eliminar riesgos antes de que se materialicen. Este enfoque preventivo incluye la implementación de sistemas de seguridad, procedimientos operativos seguros y la capacitación continua de los empleados. Además, en caso de que ocurran incidentes, se llevan a cabo investigaciones detalladas para comprender las causas y evitar recurrencias futuras.

En la actualidad, la seguridad industrial está guiada por normativas internacionales como la ISO 45001 y la ISO 14001. La ISO 45001 se centra en la Seguridad y Salud en el Trabajo, estableciendo requisitos para un sistema de gestión de la seguridad y salud ocupacional. Esta norma ayuda a las organizaciones a crear ambientes de trabajo seguros y saludables, reduciendo los accidentes y enfermedades laborales. Por otro lado, la ISO 14001 aborda la Gestión Ambiental, asegurando que las actividades industriales cumplan con los estándares ambientales y minimicen el impacto en el entorno.

La seguridad en las instalaciones industriales abarca múltiples aspectos, incluyendo el orden y la limpieza en el área de trabajo. La gestión de los elementos psicosociales también juega un papel importante, ya que el bienestar integral de los colaboradores está relacionado con un entorno laboral seguro y saludable. Factores como el estrés, la fatiga y el ambiente de trabajo influyen en la seguridad general y deben ser gestionados adecuadamente.

Integración de Sistemas de Visión Artificial en Seguridad Industrial

[11] Dice que la incorporación de sistemas de visión artificial en la seguridad industrial puede revolucionar la manera en que se gestionan los riesgos y se garantizan ambientes seguros. Estos sistemas, que utilizan tecnologías como OpenCV y MediaPipe, permiten la implementación de soluciones avanzadas para la detección y prevención de incidentes en tiempo real.



Figura 10 Esquema de sistemas de visión [12]

Por ejemplo, los sistemas de visión artificial pueden monitorear continuamente el entorno de trabajo y detectar comportamientos peligrosos o situaciones de riesgo, como la presencia de una persona en zonas restringidas o el mal funcionamiento de maquinaria. Al integrar sistemas de visión artificial con mecanismos de paro de emergencia, se puede activar automáticamente una parada segura de la maquinaria en caso de detectar una situación potencialmente peligrosa. Esta capacidad de respuesta rápida y automática aumenta la seguridad al minimizar el tiempo de reacción ante posibles accidentes.

La seguridad industrial en maquinaria busca crear un entorno de trabajo seguro mediante la prevención de riesgos, el cumplimiento de normas y la gestión proactiva de la seguridad. La integración de sistemas de visión artificial añade una capa adicional de protección, permitiendo una vigilancia continua y una respuesta inmediata a posibles incidentes. La combinación de prácticas tradicionales de seguridad y tecnologías avanzadas proporciona un enfoque integral para proteger tanto a las personas como a las instalaciones en entornos industriales.

Existen cuatro hallazgos que son protagonistas en los programas de Seguridad Industrial:

- Los accidentes laborales: son sucesos repentinos que sobrevienen por causa del trabajo y producen en el trabajador lesión orgánica, perturbación funcional, invalidez o muerte. También se incluye daños graves a la instalación o al medio ambiente.

- Los incidentes: estos son acontecimientos no deseados que, bajo condiciones un poco diferentes, podrían haber causado lesiones a las personas, daños a la instalación o pérdidas al proceso.
- Acto inseguro: se le llama también acto subestándar y es toda acción que realiza el trabajador de manera insegura o inadecuada, aumentando la probabilidad de que ocurra un accidente de trabajo.
- Condición insegura: medio o situación presente en el lugar de trabajo, caracterizada por la presencia de riesgos no controlados que pueden llevar a la ocurrencia de accidentes laborales.



Figura 11 Equipo de protección personal (EPP). [13]

Dobladoras de lámina: ¿qué son y por qué son esenciales en la industria?

Las dobladoras de lámina son instrumentos vitales en la metalurgia, diseñadas para modelar y adaptar láminas metálicas a distintas configuraciones. Estas herramientas, que varían desde modelos manuales hasta versiones hidráulicas o con Control Numérico Computarizado (CNC), ejercen presión sobre el metal, facilitando su adaptación a formas y ángulos determinados.

El metal laminado, cuyo grosor suele ser inferior a 6 mm, es un elemento esencial en la industria. Fabricado en finas planchas, puede originarse de materiales diversos como acero, aluminio o cobre. Estas planchas constituyen la base para múltiples aplicaciones, abarcando desde componentes vehiculares hasta estructuras en edificaciones.

Estas máquinas desempeñan un papel crucial en la elaboración de componentes que demandan exactitud y uniformidad. Gracias a ellas, se logra producir elementos con especificaciones precisas, garantizando estándares elevados de calidad. Las versiones modernas, en especial aquellas equipadas con tecnología CNC, ofrecen un nivel superior de exactitud en el proceso de modelado.

Las dobladoras de lámina operan mediante la aplicación de fuerza sobre el metal, colocándolo entre dos herramientas: un punzón superior y una matriz inferior.

La elección adecuada de estas herramientas es crucial, ya que deben ser seleccionadas en función del grosor y tamaño de las piezas a doblar, evitando deformaciones y colisiones durante el proceso. Por lo anterior, le contamos cuáles son las características de las dobladoras de láminas:

- Herramientas de precisión: las máquinas operan utilizando un punzón superior y una matriz inferior. La selección adecuada de estas herramientas es vital para evitar deformaciones y garantizar un doblado óptimo.
- Sistema de control avanzado: las versiones con tecnología CNC ofrecen programación detallada y gestión precisa del proceso, adaptándose a las especificaciones de cada proyecto.
- Monitoreo en tiempo real: gracias al sistema de retroalimentación, los operadores pueden ajustar y supervisar el proceso al instante, lo que es crucial para operaciones complejas o piezas con tolerancias estrictas.
- Seguridad integrada: las unidades contemporáneas incorporan sistemas de protección para el usuario, asegurando operaciones sin riesgos y eficientes.
- Optimización del funcionamiento: los sistemas de lubricación y refrigeración contribuyen a un rendimiento superior, minimizando el desgaste y maximizando la durabilidad de la máquina.
- Versatilidad en el doblado: la capacidad de la máquina se define por su fuerza y longitud máxima de doblado, adaptándose a distintos grosores y radios del material.
- Adaptabilidad y rigor: las dobladoras actuales se caracterizan por su flexibilidad y exactitud, permitiendo trabajar con diversas formas y aplicaciones.

Estas características y componentes garantizan que las dobladoras de lámina sean herramientas eficientes y precisas, capaces de satisfacer las demandas de la industria metalmecánica moderna.



Figura 12 Maquina dobladora de láminas neumática. [14]

Tipos de dobladoras: manual, hidráulica y CNC

Dependiendo de las especificaciones y requerimientos del trabajo, existen diferentes tipos de dobladoras de láminas:

- Dobladoras manuales: operadas mediante esfuerzo humano, estas herramientas son adecuadas para tareas menos intensivas. Su diseño sencillo y ligereza las hacen ideales para pequeños talleres y aplicaciones artesanales. Aunque no poseen la potencia de modelos más avanzados, son útiles en contextos donde la precisión extrema no es primordial.
- Dobladoras hidráulicas: utilizando sistemas hidráulicos, estas máquinas generan la fuerza necesaria para trabajar con materiales robustos. Su capacidad para manejar láminas más gruesas con precisión las convierte en una opción preferida en entornos industriales y de producción a gran escala.
- Dobladoras CNC: representando la tecnología más avanzada en el ámbito del doblado, estas unidades automatizadas garantizan una precisión milimétrica. Programables y altamente eficientes, se emplean en sectores donde la repetibilidad y exactitud son esenciales.

Proceso de fabricación y diseño de dobladoras de lámina

[15] Nos dice que los procesos de fabricación y diseño de dobladoras de lámina es una combinación de técnicas avanzadas y consideraciones de diseño que garantizan la precisión y eficiencia en la producción de piezas metálicas. La formación explosiva, por ejemplo, utiliza la potencia de una detonación para moldear el metal, siendo especialmente útil para piezas grandes. En este método, el metal y el molde se sumergen en agua, y una onda de choque resultante de la detonación forma el metal casi instantáneamente en la cavidad del molde.

Por otro lado, la formación electrohidráulica se basa en la energía de la combustión de un alambre metálico delgado. En este proceso, una descarga eléctrica a través de electrodos sumergidos en líquido vaporiza el alambre, creando una onda de choque que da forma al metal. Aunque es más adecuado para piezas más pequeñas, ofrece una precisión notable.

La integración de sistemas de visión artificial en las dobladoras de lámina puede mejorar significativamente la seguridad industrial. Estos sistemas pueden ser utilizados para monitorear continuamente el proceso de fabricación y detectar cualquier anomalía en tiempo real. Por ejemplo, los sistemas de visión artificial pueden identificar si hay un mal alineamiento de las piezas o si se está produciendo un defecto en la formación del metal. Al detectar estos problemas de manera temprana, los sistemas de visión artificial pueden alertar a los operadores o incluso activar automáticamente mecanismos de seguridad, como paros de emergencia, para prevenir accidentes.



Figura 13 Proceso de fabricación y diseño de dobladoras de lámina [15]

Además, el diseño de dobladoras de lámina y su operación deben cumplir con normas internacionales como la ISO 45001 para la seguridad y salud en el trabajo. Estas normas establecen requisitos para crear un ambiente de trabajo seguro y saludable, y la integración de tecnologías avanzadas como los sistemas de visión artificial puede ayudar a cumplir con estos requisitos. Por ejemplo, al incorporar sistemas que mejoran la precisión y reducen el riesgo de errores, se facilita el cumplimiento de normas de seguridad y se minimizan los riesgos operativos.

El proceso de fabricación y diseño de dobladoras de lámina se relaciona estrechamente con la seguridad industrial y los sistemas de visión artificial. La integración de estas tecnologías avanzadas no solo mejora la precisión y eficiencia de las máquinas, sino que también juega un papel fundamental en la mejora de la seguridad al permitir una monitorización continua, cumplir con normas de seguridad, y prevenir riesgos operativos. Estas interacciones destacan la importancia de considerar tanto la tecnología avanzada como las prácticas de seguridad en el diseño y operación de maquinaria industrial.

La formación electromagnética, otra técnica avanzada, utiliza una bobina eléctrica colocada cerca de una pieza de trabajo metálica. Una descarga eléctrica crea un campo magnético que, al interactuar con el metal, genera corrientes parásitas. Estas corrientes, a su vez, crean un campo magnético opuesto, y las fuerzas resultantes dan forma a la pieza de trabajo, siendo comúnmente utilizada para formar tubos o láminas metálicas delgadas.

Más allá de las técnicas, las consideraciones de diseño son fundamentales. Es esencial tener en cuenta factores como el grosor del material, la ubicación y el tamaño de los agujeros, y la orientación y el radio de los dobleces. Además, las propiedades del material y cómo se verán afectadas por los procesos de alta energía son cruciales para garantizar la integridad y funcionalidad de la pieza final. Estos métodos y consideraciones, en conjunto, definen el vanguardista campo de la fabricación y diseño de dobladoras de lámina.

Una de las principales ventajas de las dobladoras es su capacidad para producir piezas de alta precisión en tiempos reducidos. A diferencia de los métodos manuales, estas máquinas garantizan una precisión milimétrica, lo que se traduce en piezas de mejor calidad y en una reducción significativa de los desechos y rechazos. Además, su eficiencia operativa permite una producción en masa, lo que es esencial para proyectos de construcción a gran escala.

En cuanto a las aplicaciones, las dobladoras de lámina son fundamentales en la creación de componentes estructurales para edificios, puentes y otras infraestructuras. Su capacidad para doblar láminas de acero de diferentes grosores las hace versátiles para una variedad de proyectos. Además, no solo se limitan a la construcción: también son esenciales en la fabricación de vehículos, maquinaria y una amplia gama de productos industriales.

Por último, la adaptabilidad de estas máquinas es otra de sus grandes ventajas. Con la incorporación de tecnologías como el CNC, las dobladoras pueden ser programadas para realizar cortes y dobleces específicos, adaptándose a las necesidades cambiantes de cada

proyecto. Esta flexibilidad, combinada con su eficiencia y precisión, hace que las dobladoras de lámina sean una inversión valiosa para cualquier empresa dedicada a la construcción de acero.



Figura 14 Proceso de dobles. [15]

En cuanto a las aplicaciones, las dobladoras de lámina son fundamentales en la creación de componentes estructurales para edificios, puentes y otras infraestructuras. Su capacidad para doblar láminas de acero de diferentes grosores y tamaños las convierte en herramientas versátiles y esenciales para una amplia gama de proyectos de construcción. Estas máquinas no solo manejan acero, sino que también pueden trabajar con otros metales como el aluminio y el cobre, expandiendo su utilidad a diferentes industrias.

Además de su papel en la construcción, las dobladoras de lámina son cruciales en la fabricación de vehículos, maquinaria y una amplia gama de productos industriales. En la industria automotriz, por ejemplo, se utilizan para conformar paneles de carrocería, chasis y otras partes estructurales que requieren una alta precisión y resistencia. En la fabricación de maquinaria, permiten la creación de componentes metálicos complejos que son necesarios para el ensamblaje y operación de diversas máquinas.

La adaptabilidad de las dobladoras de lámina es otra de sus grandes ventajas. Con la incorporación de tecnologías avanzadas como el Control Numérico por Computadora (CNC), estas máquinas pueden ser programadas para realizar cortes y dobleces específicos con una precisión excepcional. Esto no solo optimiza la producción al permitir la automatización de procesos, sino que también permite a las empresas adaptarse a las necesidades cambiantes de cada proyecto. Los programas CNC facilitan la configuración

rápida y precisa de la máquina para diferentes trabajos, reduciendo el tiempo de ajuste y minimizando errores.

Además, la capacidad de personalizar los patrones de doblado y corte según las especificaciones del cliente o del diseño del proyecto agrega una dimensión de flexibilidad que es especialmente valiosa en industrias con requisitos de diseño específicos. Las dobladoras de lámina equipadas con CNC pueden producir una variedad de formas y diseños complejos que serían difíciles de lograr con métodos manuales, lo que incrementa la versatilidad y eficiencia de la producción.

Por último, la precisión y eficiencia que ofrecen las dobladoras de lámina modernas no solo garantizan la calidad de los productos finales, sino que también contribuyen a la reducción de desperdicios de material y costos operativos. Esta eficiencia en el uso de materiales y la capacidad para realizar producciones en serie hacen que las dobladoras de lámina sean una inversión valiosa para cualquier empresa dedicada a la fabricación y construcción con acero.

La versatilidad, precisión y adaptabilidad de las dobladoras de lámina, especialmente cuando se integran con tecnologías avanzadas como el CNC, las convierten en herramientas indispensables en la industria moderna. Su capacidad para manejar una amplia gama de aplicaciones y adaptarse a diferentes necesidades de proyectos subraya su importancia y valor en el ámbito de la fabricación industrial.

Programación en Arduino.

[16] Dice que el lenguaje del Arduino está basado en el mítico lenguaje C. Si ya has trabajado en C este tutorial te parecerá un paseo. Si no, te basta con saber que C es el lenguaje en el que se ha desarrollado los sistemas operativos UNIX, Linux, y cientos de sistemas, programas y aplicaciones de ordenador. El lenguaje del Arduino es una versión reducida y mucho más sencilla de manejar que el lenguaje C. El objetivo de este lenguaje es que puedas programar de una manera intuitiva concentrándote en lo que quieres hacer más que en la manera de hacerlo.

Arduino fue desarrollado originalmente en el Interactive Design Institute en Ivrea (Italia) donde los estudiantes estaban familiarizados con un lenguaje llamado Processing. Este lenguaje estaba orientado a estudiantes de arte y diseño y utilizaba un entorno de desarrollo visual e intuitivo en el cual se basó el entorno de desarrollo del Arduino y su lenguaje de programación.

Trabajar con un Arduino consiste fundamentalmente en interactuar con los diferentes puertos de entrada y salida del Arduino. A fin de evitar al programador el engorro y la complejidad de programar estos puertos (ya sean analógicos, digitales o de cualquier otro tipo) el lenguaje de Arduino usa una serie de librerías (de las que no te tienes que preocupar

ya que forman parte del lenguaje, ya las iremos viendo con detenimiento más adelante). Estas librerías te permiten programar los pins digitales como puertos de entrada o salida, leer entradas analógicas, controlar servos o encender y apagar motores de continua. La mayor parte de estas librerías de base ("core libraries") forman parte de una macro librería llamada Wiring desarrollada por Hernando Barragán.



Figura 15 Lenguaje de Arduino. [17]

Capítulo III.
Desarrollo

Herramientas de visión artificial.

Para poder hacer el uso adecuado en este trabajo se darán a conocer las herramientas necesarias para aplicarlo.

1. VS Code

Como primera parte se necesita un editor de texto donde el editor de texto a utilizar es Visual Studio Code, que se instala desde la página de Microsoft seleccionando la versión de sistema operativo, donde Visual Studio Code es un editor de texto que nos permitirá crear nuestros programas y ejecutarlos ya que cuenta con una terminal pre integrada asi como se muestran en las **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.**.

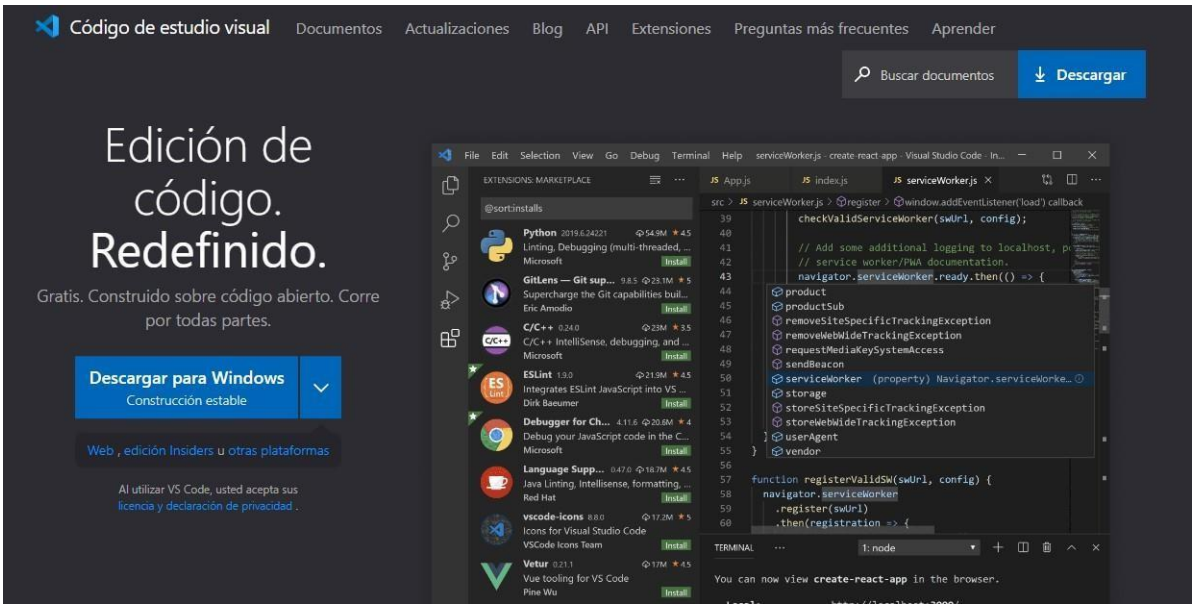

Figura 16 Visual Studio Code.

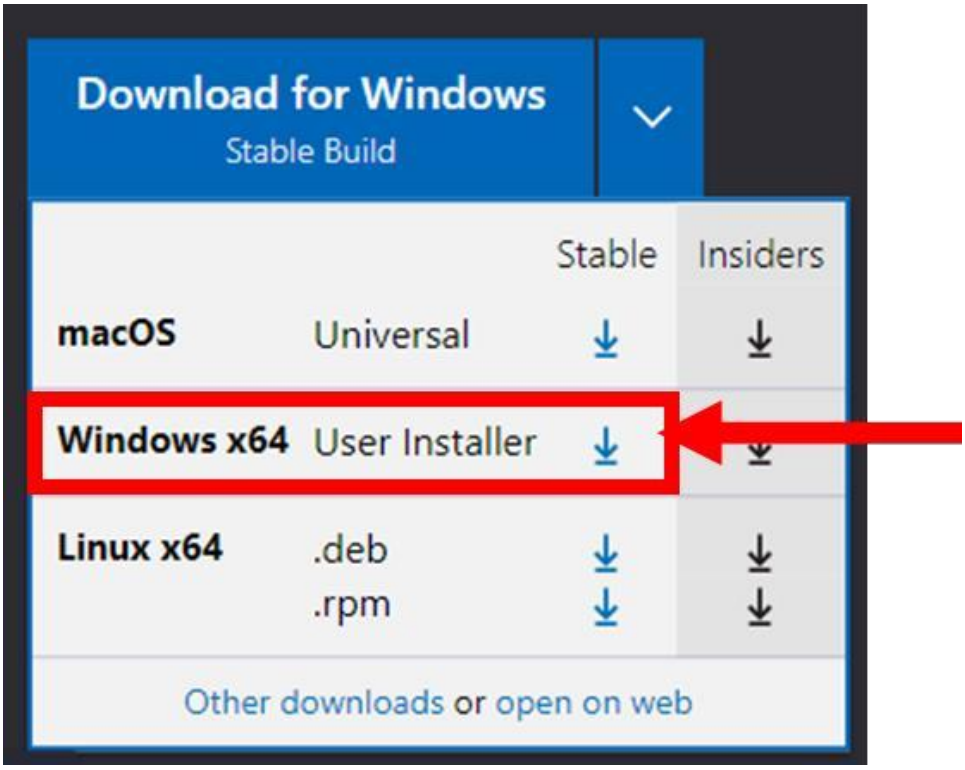

Figura 17 Versión de Visual Studio Code.

Ya instalado el programa y haberlo ejecutado como administrador se debe abrir el programa, el cuál mostrará una ventana de bienvenida como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

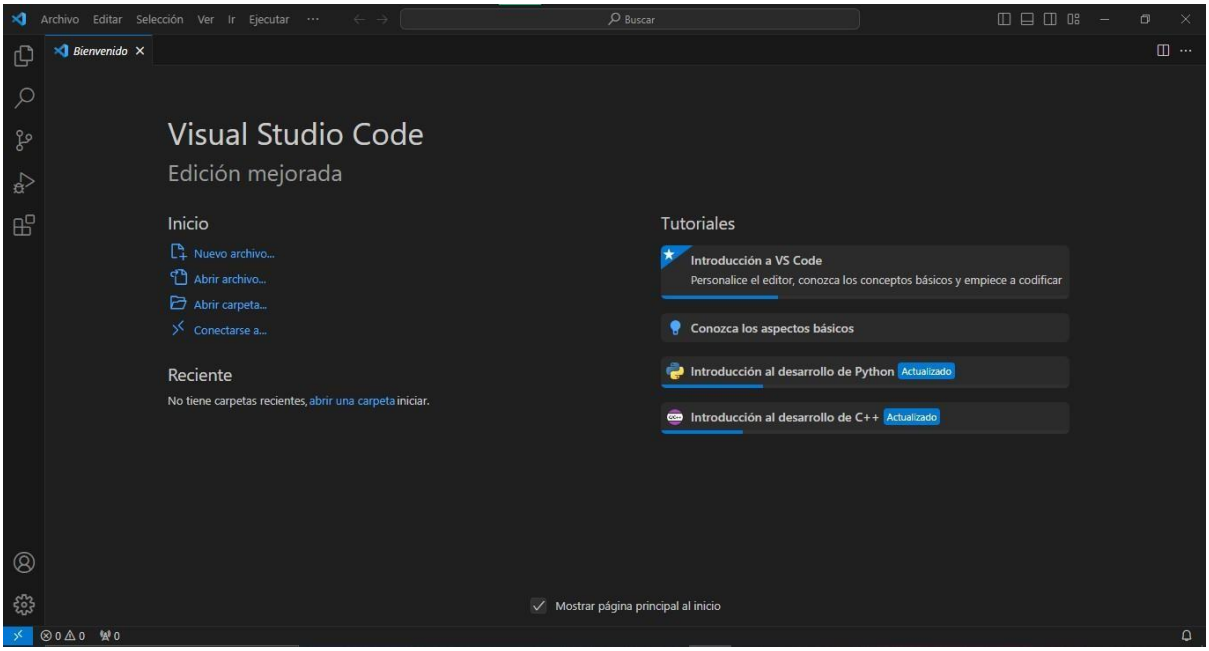

Figura 18 Bienvenida a Visual Studio.

Visual Studio solo es un editor de texto el cual que no cuenta con el compilador e intérprete de Python, por lo cual hay que instalarlo donde Visual Studio Code tiene una herramienta mediante el uso de la instalación de extensiones en la cual se puede instalar Python.

2. Extensiones (compilador e interprete python)

Posterior mente para poder hacer la descarga de las extensiones a utilizar dentro del editor de texto que es Visual Studio Code se debe ir al apartado de extensiones donde hay podremos instalar las extensiones necesarias asi como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

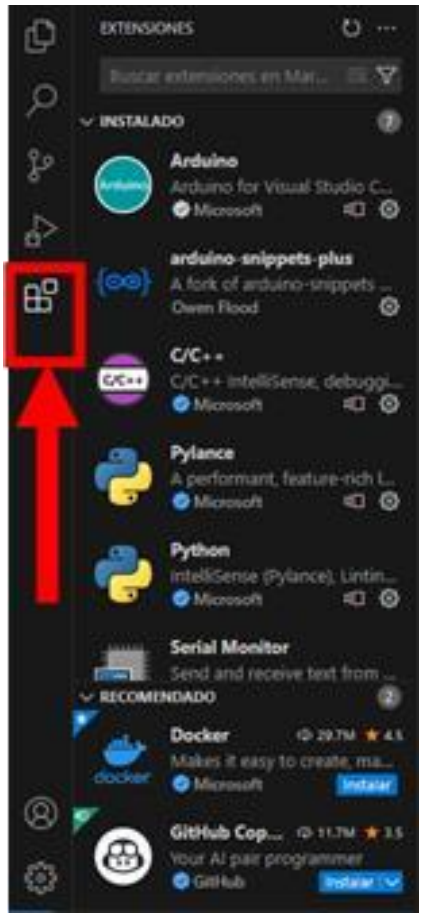

Figura 19 Área de Extensiones

Posteriormente, se escribe en el buscador las extensiones a utilizar que este caso será Python, Pylance, C/C++, Serial Monitor y Spanish Language Pack for Visual Studio Code para que Visual Studio Code, este último sirve para que se muestre en español, como se muestra en la **¡Error! No se encuentra el origen de la referencia.** se debe se instalar cada una de las librerías mencionadas.



Figura 20 Descarga de Extensiones.

3. Librerías (open cv).

Para el desarrollo del código se hace uso de 2 librerías las cuales nos ayudaran a mostrar los fotogramas de la imagen, las cuales son Opencv y Mediapipe como se muestra en la **¡Error! No se encuentra el origen de la referencia.**:



Figura 21 Librerías.

El enfoque que se le dará a estas dos librerías son las siguientes: OpenCv nos va a proporcionar funciones para la manipulación de imágenes, extracción de características, segmentación y filtrado.

Donde, además, nos incluirá implementaciones de algoritmos de aprendizaje automático que pueden entrenarse para mejorar la precisión de la detección de manos en diferentes contextos donde se demostrara en dicho código, por otro lado, Media pepe nos va a proporciona su uso en detección de objetos donde su principal uso es la visión artificial lo cual el enfoque se le dará en este trabajo será la detección de manos.

4. Captura de imagen y video desde la web cam.

Para poder hacer el uso de una cámara en este caso será la que está integrada a la computadora, como se muestra en la **¡Error! No se encuentra el origen de la referencia.**, esta función nos ayudara a llamar a la función de capturar video donde el valor "0" que se encuentra dentro del paréntesis especifica el índice del conjunto de cámaras conectadas en la computadora, para este ejemplo, el índice "0" especifica la cámara predeterminada de la computadora, donde si queremos hacer uso de una web cam externa debemos identificar previamente el índice asignado al dispositivo seleccionado.

```
def main():
    cap = cv2.VideoCapture(0)
```

Figura 22 Función de capturar Imagen.

En la **¡Error! No se encuentra el origen de la referencia.** la instancia cap es utilizada para representar un objeto de captura de video, creado mediante la clase Video Capture(0) de la librería cv2. Esta función inicializa un objeto de captura de video que se asocia con la cámara predeterminada del sistema (cámara principal).

Esta línea de código se utiliza para abrir una conexión con la cámara de video del dispositivo, permitiendo la captura de secuencias de imágenes en tiempo real. El objeto cap proporciona métodos y atributos que permiten acceder a cada fotograma de video capturado.

Esta librería es utilizada para la Captura de Video en Tiempo Real, ya que puede utilizarse para capturar y procesar secuencias de video en tiempo real desde la cámara de un ordenador.

En procesamiento de Imágenes, permite aplicar algoritmos de procesamiento de imágenes y visión por computadora en tiempo real a los fotogramas capturados.

También, se puede utilizar para implementar sistemas de reconocimiento de objetos, seguimiento de objetos en movimiento, o cualquier aplicación que involucre el análisis de video en vivo.

Por último, el uso de Visión por Computadora y Aprendizaje Automático se utiliza comúnmente en proyectos de visión por computadora y aprendizaje automático donde se requiere acceso a secuencias de video para entrenar modelos o realizar análisis visual.

Es importante destacar que esta línea de código es solo el primer paso, y generalmente se sigue con un bucle donde se capturan y procesan continuamente los fotogramas hasta que se interrumpe la ejecución del programa.

Dicha función nos permitirá iniciar el proceso de captura de video, como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

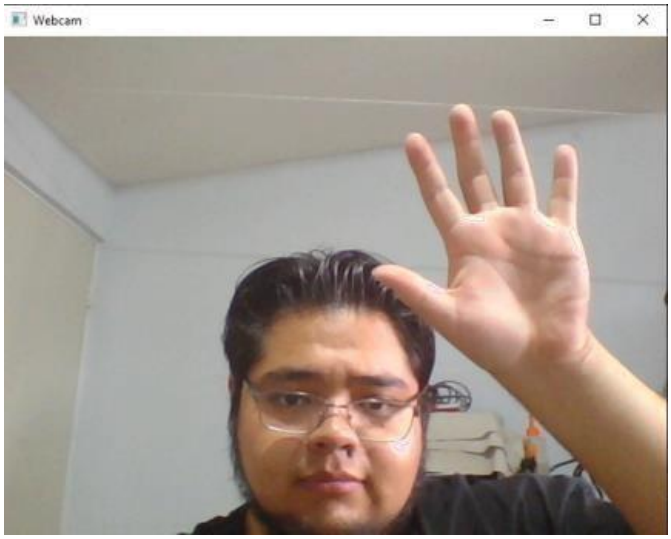

Figura 23 Cámara de la Computadora.

Como ejemplo, se muestra en la **¡Error! No se encuentra el origen de la referencia.** un código simple en donde la función que hará es mostrar el uso de la web cam de la computadora .

```python
import cv2

def iniciar_webcam():
    # Abre la conexión con la webcam
    cap = cv2.VideoCapture(0)

    while True:
        # Lee un frame de la webcam
        ret, frame = cap.read()

        # Muestra el frame en una ventana
        cv2.imshow('Webcam', frame)

        # Rompe el bucle si se presiona la tecla 'q'
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # Libera la conexión con la webcam
    cap.release()

    # Cierra todas las ventanas
    cv2.destroyAllWindows()

if __name__ == "__main__":
    iniciar_webcam()

```

Figura 24 Código de Ejemplo para usar la cámara.

1. Detección de manos mediante visión artificial

A continuación, se dará la explicación de como con ayuda de las librerías y la función de Video Capture se logra mostrar la web cam y hace la detección de las manos asi como en diferentes posiciones y distancias.

a) Librerías (Mediapipe)

Para lograr esto, se utilizan las bibliotecas OpenCV para la manipulación de imágenes y la detección de la mano, y la biblioteca Mediapipe para obtener los puntos clave de la mano y la detección de manos como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

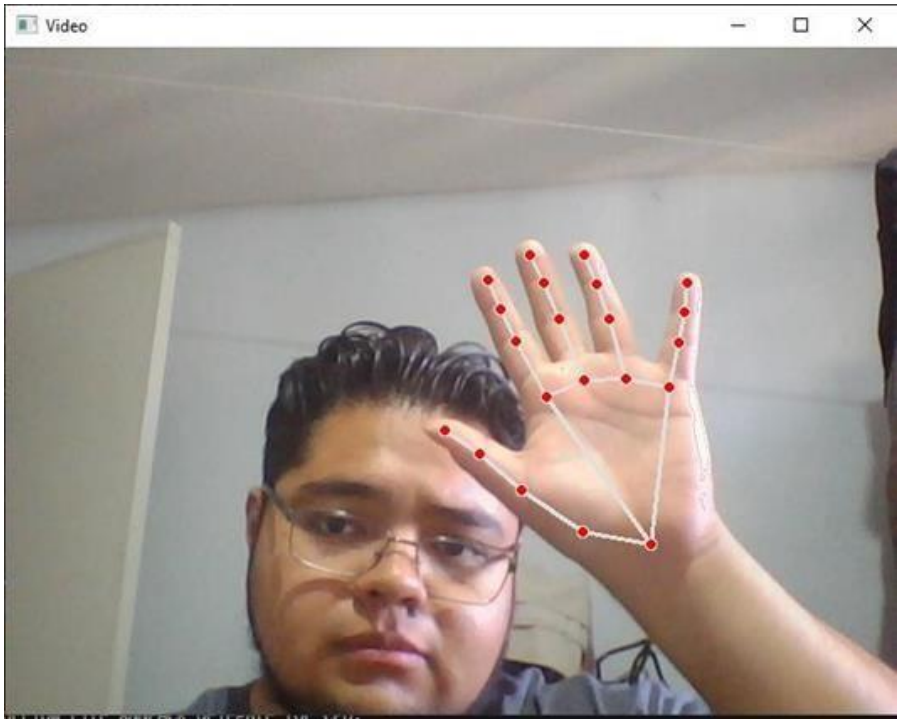

Figura 25 Detección de mano con sus puntos.

Para poder llegar a hacer la detección de las manos y objetos se hace uso de la del método hands() de la clase solutions (mp.solutions.hands), ya que la función hace el seguimiento de manos, la prime línea de código mostrado en la **¡Error! No se encuentra el origen de la referencia.** define el atributo de esa clase que es “hands” y en la segunda línea se da el nombre de la clase que en este caso es “Hands” .

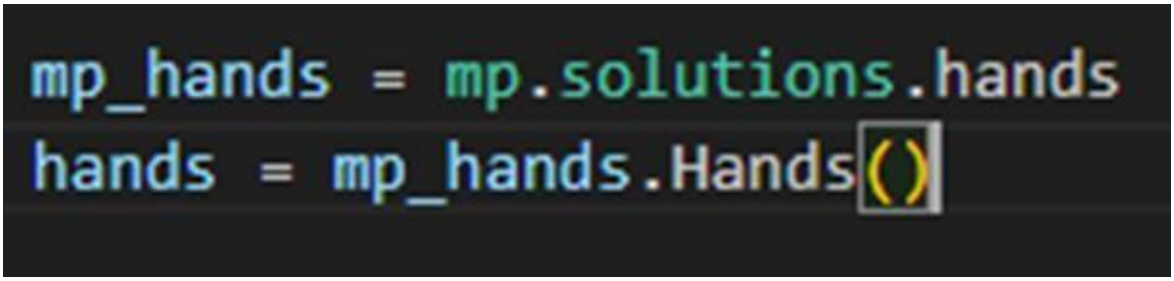

Figura 26 Función de seguimiento de manos.

Como ejemplo, se muestra en la Figura 27 el código cuya función es mostrar mediante el uso de la web cam la detección de las manos y puntos biométricos de la mano.

```python
import cv2
import mediapipe as mp
# Inicializar mediapipe
mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
# Inicializar la cámara
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        print("No se pudo abrir la cámara.")
        break
    # Convertir el frame a BGR para mediapipe
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    # Procesar el frame con mediapipe
    results = hands.process(rgb_frame)
    # Verificar si se detectaron manos
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            # Dibujar los puntos de la mano
            for point in hand_landmarks.landmark:
                height, width, _ = frame.shape
                cx, cy = int(point.x * width), int(point.y * height)
                cv2.circle(frame, (cx, cy), 5, (255, 0, 0), -1)
    # Mostrar el frame
    cv2.imshow("Hand Tracking", frame)
    # Salir con la tecla 'q'
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# Liberar la cámara y cerrar las ventanas
cap.release()
cv2.destroyAllWindows()
```

Figura 27 Código de muestra para detección de manos.

Detección de cercanía entre manos y objetos

Para poder hacer la detección de objetos se hace uso del método de Canny, ya que este método permite detectar los bordes de los objetos, en la **¡Error! No se encuentra el origen de la referencia.** se muestra un ejemplo de detección de bordes, en este caso no se tiene un código que detecte los bordes de un objeto especifico, por lo tanto, mostrará todo borde que se sea detectado y lo encerrará dentro de un recuadro en color verde.

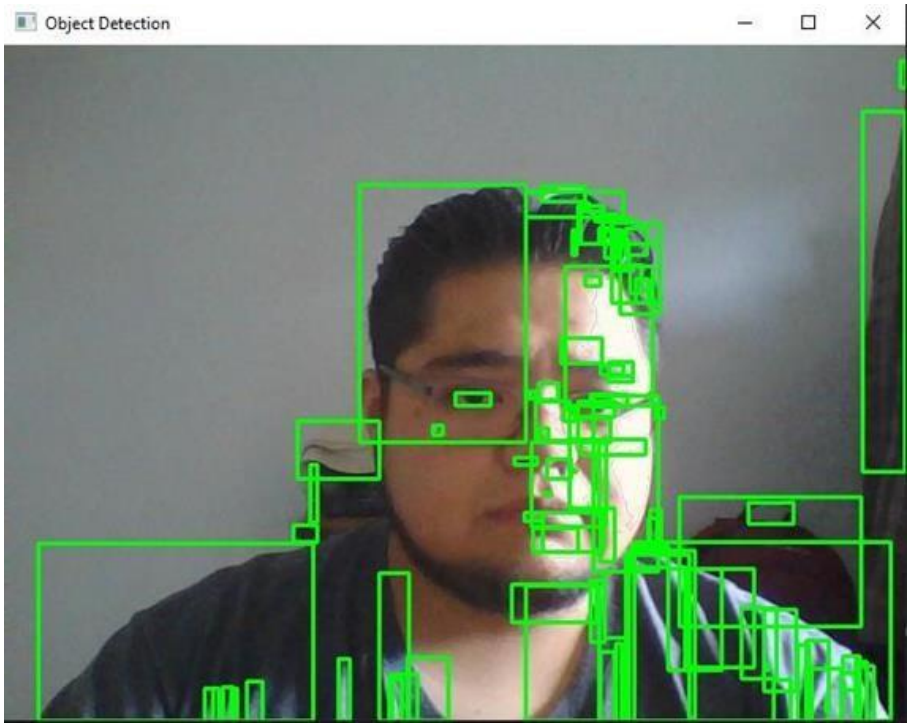

Figura 28 Detección de bordes.

Como referencia, la Figura 17 muestra el código y la explicación de su funcionamiento.

```python
import cv2
# Inicializar la cámara
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        print("No se pudo abrir la cámara.")
        break
    # Convertir el frame a escala de grises
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Aplicar un desenfoque para reducir el ruido
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    # Detectar los bordes con Canny
    edges = cv2.Canny(blurred, 50, 150)
    # Encontrar contornos en los bordes
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    # Dibujar rectángulos alrededor de los objetos
    for contour in contours:
        x, y, w, h = cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    # Mostrar el frame con los rectángulos
    cv2.imshow("Object Detection", frame)
    # Salir con la tecla 'q'
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# Liberar la cámara y cerrar las ventanas
cap.release()
cv2.destroyAllWindows()
```

Figura 29 Código de detección de bordes en objetos.

1. Inicialización de la cámara:
    - Se inicializa la cámara mediante la función cv2.VideoCapture(0), donde 0 indica el índice de la cámara web predeterminada.
2. Bucle principal:
    - Se inicia un bucle while que se ejecutará mientras la cámara estéabierta (cap.isOpened()).
3. Captura de frames:
    - En cada iteración del bucle, se captura un frame de la cámaramediante cap.read().
4. Conversión a escala de grises:
    - El frame capturado se convierte a escala de grises utilizando

cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY).

5. Desenfoque gaussiano:
    - Se aplica un desenfoque gaussiano al frame en escala de grises para reducir el ruido utilizando cv2.GaussianBlur(gray, (5, 5), 0).
6. Detección de bordes con Canny:
    - Se utiliza el algoritmo Canny para detectar los bordes en el framedesenfocado mediante cv2.Canny(blurred, 50, 150).
7. Encontrar contornos:
    - Se encuentran los contornos en los bordes detectados mediantecv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE).
8. Dibujar rectángulos alrededor de los objetos:
    - Se itera sobre los contornos encontrados y se obtiene el rectángulo delimitador utilizando cv2.boundingRect(contour). Luego, se dibujaun rectángulo verde alrededor de cada objeto en el frame original.
9. Mostrar el frame con rectángulos:
    - Se muestra el frame con los rectángulos dibujados mediante cv2.imshow("Object Detection", frame).
10. Salir del programa:
    - El bucle se ejecuta continuamente hasta que se presiona la tecla 'q', momento en el cual se libera la cámara y se cierran las ventanas utilizando cap.release() y cv2.destroyAllWindows().

Pruebas de detección de manos en diferentes entornos

Distancia de captura:

- Distancia a 1 metro



Figura 30 Distancia de 1 metro.

- Distancia a 5 metros



Figura 31 Distancia de 5 metros.

- Distancia a 10 metros.



Figura 32 Distancia de 10 metros.

Detección y alerta de área de trabajo insegura

Para lograr hacer que el código pueda detectar un paro de emergencia y en un área en especifica se debe hacer que pueda imprimir los textos de go y stop que serán los que dispare cuando detecte la mano como se puede ver en la **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.**

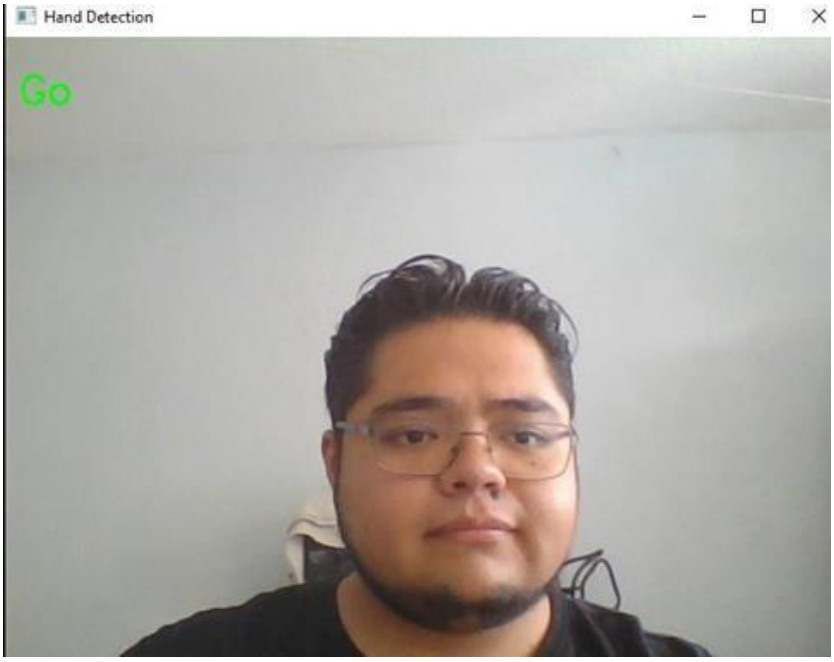

Figura 33 Impresión de texto Go.

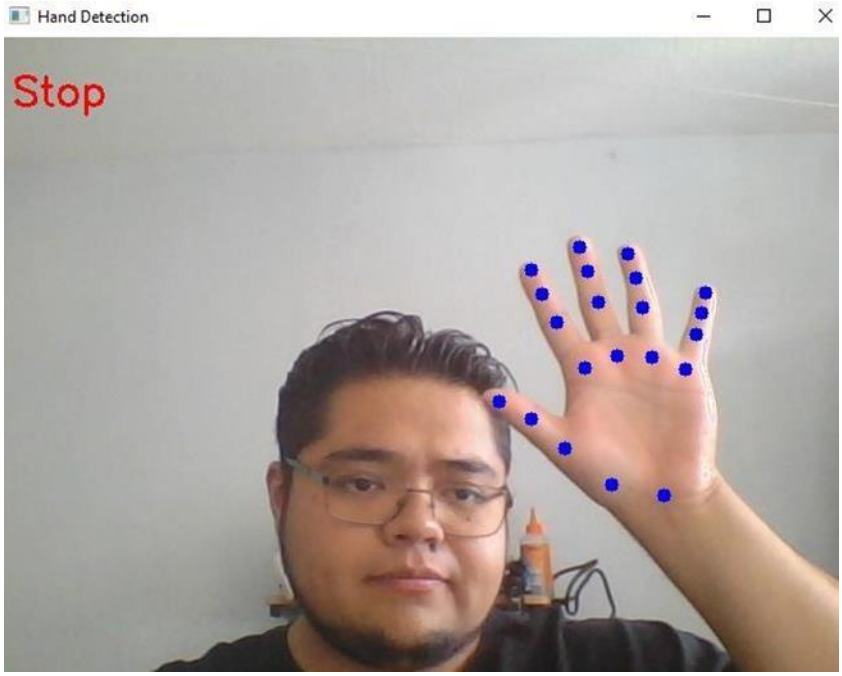

Figura 34 Impresión de texto Stop.

Detección de objetos

Como se mostró anterior mente en la detección de objetos se observa que hay mucho marguen de error ya que en un entorno el método de cani detectara los bordes de los objetos en el área de visión de la cámara.

Para poder hacer que se tenga un enfoque más certero al encerrar objetos se usa la función object_classifier = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml') donde la funcionalidad que tendrá esta línea es:

Crear un objeto llamado object_classifier mediante la utilización de la clase cv2.CascadeClassifier de la biblioteca OpenCV en Python. Este objeto se emplea para cargar un clasificador en cascada destinado a la detección de rostros. La ruta al modelo preentrenado se establece concatenando la ubicación de los archivos de datos de OpenCV (cv2.data.haarcascades) con el nombre del archivo XML específico para la detección de caras frontales, es decir, 'haarcascade_frontalface_default.xml'. En resumen, la línea de código configura un clasificador en cascada para la detección de caras frontales, que puede utilizarse posteriormente para identificar rostros en imágenes o secuencias de video.

Detección de máquina dobladora

para poder hacer la detección de la hoja que en este caso es de una maquina dobladora de lámina se debe dibujar un área limitada que será un rectángulo en el centro de la cámara donde este rectángulo tiene a tener medidas donde se puede ajustar ya que la maquina presenta una longitud y una altura y es el área que dibujar.

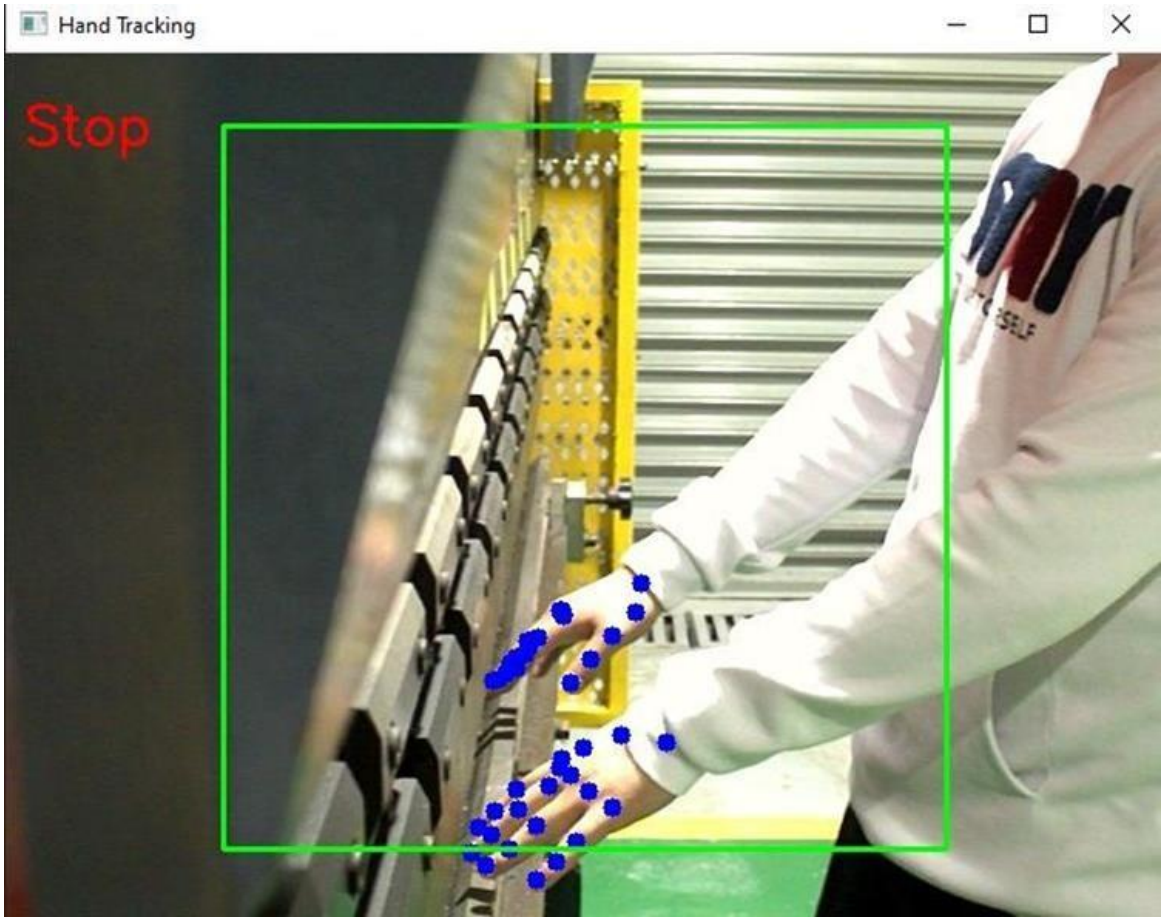

Figura 35 Vista del área de la maquina dobladora con las manos puestas.



Figura 36 Vista del área de la maquina dobladora sin manos puestas.

El área que se ocupará poder hacer el paro de emergencia será el área interna del rectángulo ya que esa será la detección de la hoja de la máquina, en este caso el área que se dibuja en los fotogramas llegará tener un marguen de error ya que no se puede hacer una medida certera como se muestra en la **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.**.

Conexión con Arduino.

Como parte de la programación se debe de realzar la conexión con el Arduino ya que esto dará pauta de que el programa funciona al 100% como uno de los pasos que se debe de tomar en cuenta es la descarga del programa Arduino ya que con esto se debe de hacer la carga del programa con el cual dará el funcionamiento.

1er paso: Descarga del software Arduino.

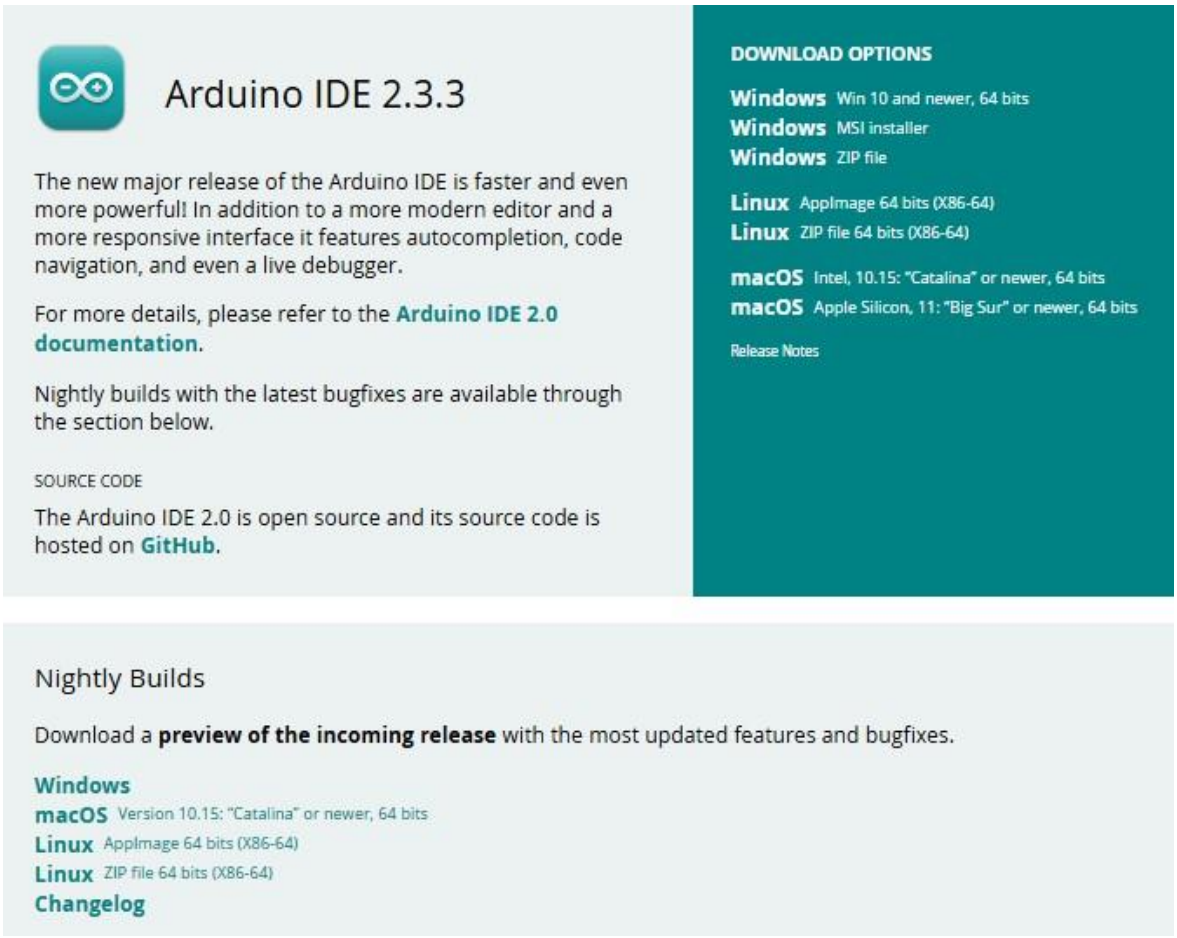

Figura 37 Software de Arduino

2do paso: carga del programa en el Arduino.

```
int ledPin = 13;  // Pin del LED, ajusta según tu conexión

void setup() {
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    if (Serial.available() > 0) {
        char command = Serial.read();
        if (command == '1') {
            digitalWrite(ledPin, HIGH);  // Enciende el LED
        } else if (command == '0') {
            digitalWrite(ledPin, LOW);   // Apaga el LED
        }
    }
}
```

Figura 38 Programa de Arduino.

El programa que se muestra en la Figura 38 nos muestra el código que se declaró en el software de Arduino para que pueda ser cargado en el Arduino y esta sea una referencia de las acciones que va a realizar el Arduino ya con el programa cargado y dirigido a él.

3er paso: programación de arduino en visual studio code.

Para poder hacer la conexión de Arduino se necesitan 2 nuevas librerías las cuales van a hacer de suma importancia en la comunicación con el arduino.

1ra librería: Serial

La librería serial en Python, la cual se importa con import serial, permite la comunicación a través de puertos seriales los cuales en este caso serían los puertos de nuestro dispositivo USB de la computadora. Esto es especialmente útil en aplicaciones donde se necesitan comunicarse con dispositivos externos, como microcontroladores, sensores, o módulos que requieren la transmisión y recepción de datos en serie. En Visual Studio Code, usar import serial permite programar scripts que interactúan con estos dispositivos mostrada en la Figura 39

2da librería: Time

La librería time en Python, que se importa con import time, nos proporciona varias funciones útiles para manipular y controlar el tiempo en el programa. Al usar import time en Visual Studio Code o en cualquier otro entorno, puedes realizar tareas relacionadas con

la medición, el manejo de tiempos de espera y el formateo de fechas y horas asi como se muestra en la Figura 39.

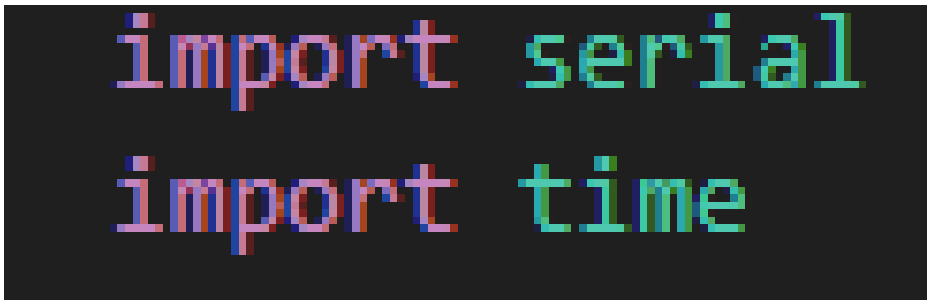

Figura 39 Librerías de serial y time.

Por consiguiente, en la Figura 40 y la Figura 41 se muestra el código completo de la conexión con arduino.

```python
import cv2
import mediapipe as mp
import serial
import time

def main():
    # Configuracion de la conexión con el Arduino y el puerto COM
    arduino = serial.Serial('COM4', 9600)
    time.sleep(2)  # Espera a que se establezca la conexión

    cap = cv2.VideoCapture(0)

    mp_hands = mp.solutions.hands
    hands = mp_hands.Hands()

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            continue

        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        result = hands.process(frame_rgb)

        if result.multi_hand_landmarks:
            for landmarks in result.multi_hand_landmarks:
                # Dibujar los puntos de la mano
                for point in landmarks.landmark:
                    height, width, _ = frame.shape
                    cx, cy = int(point.x * width), int(point.y * height)
                    cv2.circle(frame, (cx, cy), 5, (255, 0, 0), -1)
```

Figura 40 Código con Arduino

```
                # Dibujar un rectángulo en el centro
                rect_size = 400
                cv2.rectangle(frame, (width//2 - rect_size//2, height//2 - rect_size//2),
                              (width//2 + rect_size//2, height//2 + rect_size//2), (0, 255, 0), 2)

                # Verificar si la mano está dentro del rectángulo
                if width//2 - rect_size//2 < cx < width//2 + rect_size//2 and height//2 - rect_size//2 < cy < height//2 + rect_size//2:
                    print("Stop")
                    arduino.write(b'1')  # Enviar comando para encender el LED
                    cv2.putText(frame, 'Stop', (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
                else:
                    print("Go")
                    arduino.write(b'0')  # Enviar comando para apagar el LED
                    cv2.putText(frame, 'Go', (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
            else:
                # Si no se detecta una mano, el estado es "Go"
                arduino.write(b'0')
                cv2.putText(frame, 'Go', (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

            cv2.imshow('Hand Tracking', frame)

            if cv2.waitKey(1) & 0xFF == 27:
                break
        # Cerrar conexión y liberar recursos
        arduino.close()
        cap.release()
        cv2.destroyAllWindows()

if __name__ == "__main__":
    main()
```

Figura 41 Código con Arduino.

Dando como esto el resultado la conexión completa del Arduino con visual studio code y su funcionamiento completo con el programa donde en la Figura 42 marca go y en la Figura *43* marca stop.

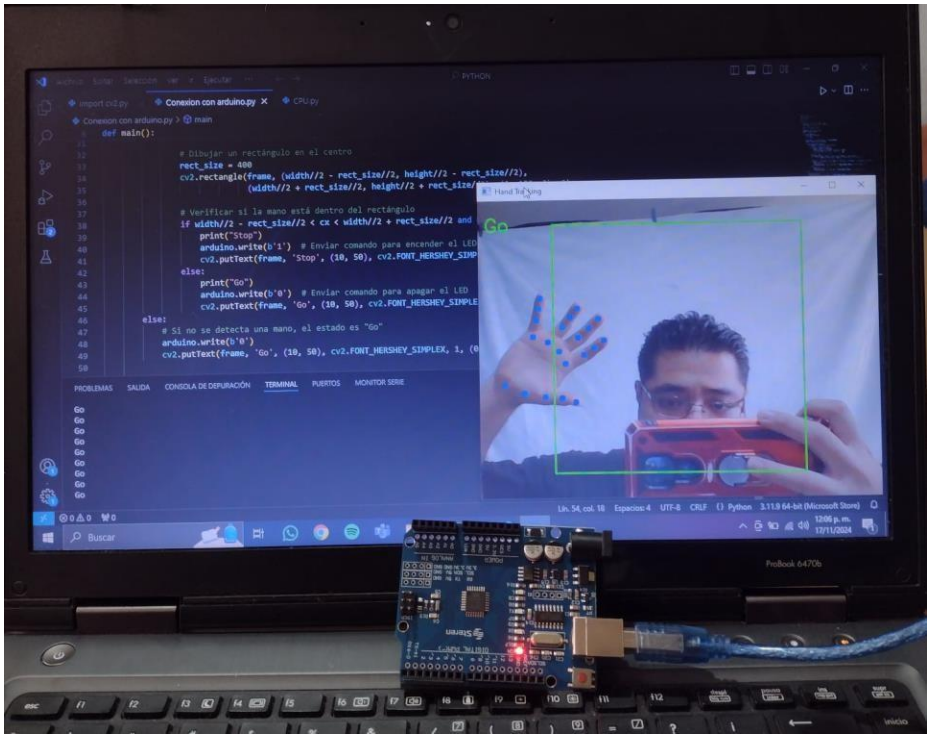

Figura 42 Conexión con Arduino marcando go.

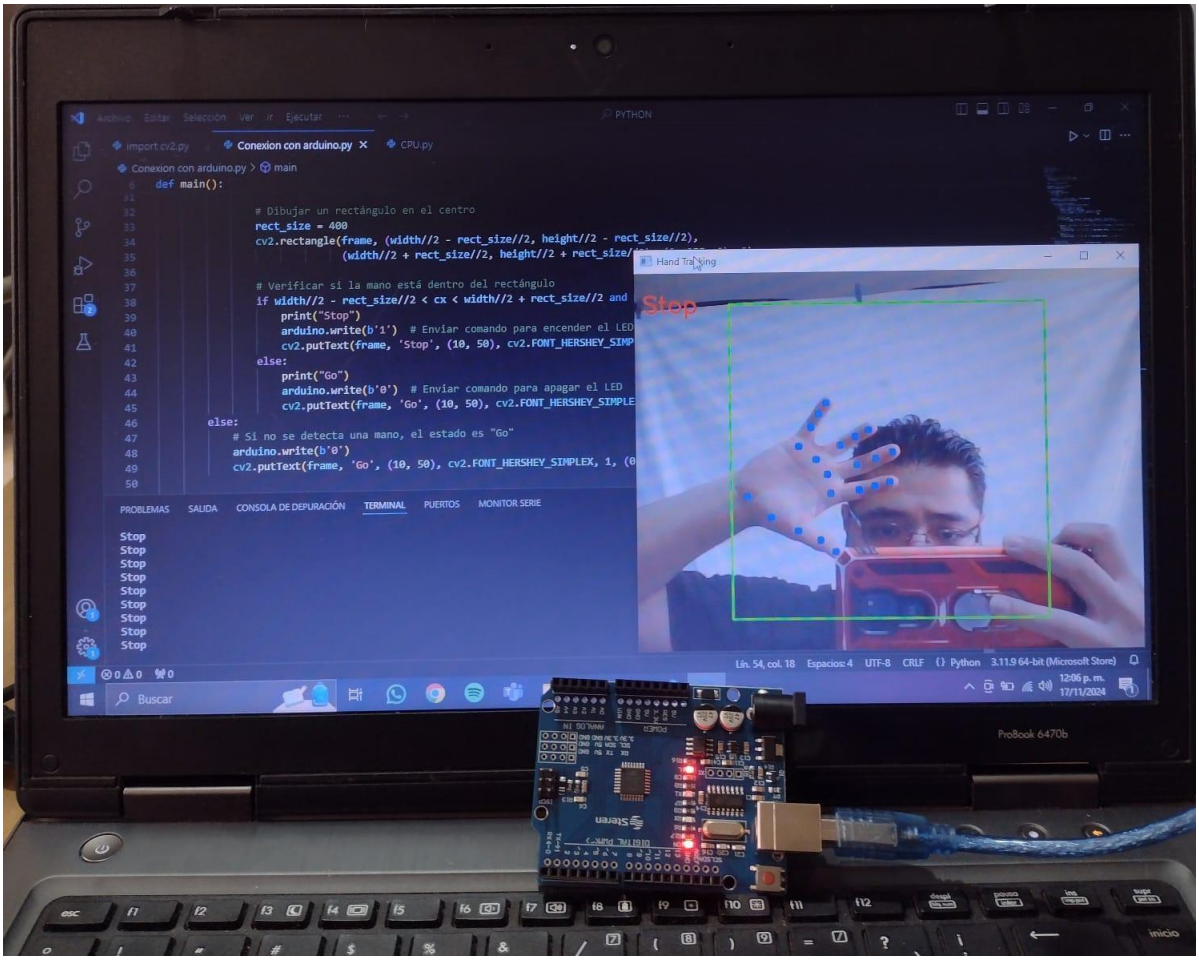

Figura 43 Conexión con Arduino marcando stop.

Capítulo IV.
Resultados

Los procedimientos explicados en la sección de desarrollo fueron aplicados en la propuesta de un sistema de seguridad de paro de emergencia en una Dobladora CNC del tipo dobladora hidráulica de cortina CNC WC67K-400. La propuesta consiste en monitorear mediante visión artificial la operación de la dobladora CNC y detectar la cercanía de las manos al accionamiento de la cortina, si se detecta que las manos están demasiado cerca de la cortina, se corta la señal del pedal hacia la dobladora CNC y de esta manera impedir que la cortina siga bajando resguardando la integridad del operador.

Para la propuesta se utilizó una cámara webcam externa a la computadora y se fijó en la dobladora CNC de tal manera que la captura de video estuviera enfocada en el área de trabajo correspondiente a la operación de la cortina. En la captura de video el área de operación insegura se encerró en un rectángulo de color verde, de tal forma que cuando el algoritmo de detección de manos las detecte dentro del área de trabajo insegura se dispara una alerta de STOP mediante un letrero de color rojo como se muestra en la Figura 15, si la o las manos se encuentran fuera del área insegura, el letrero mostrado es la mediante la palabra GO como se muestra en la Figura 14, que corresponde a un funcionamiento seguro.

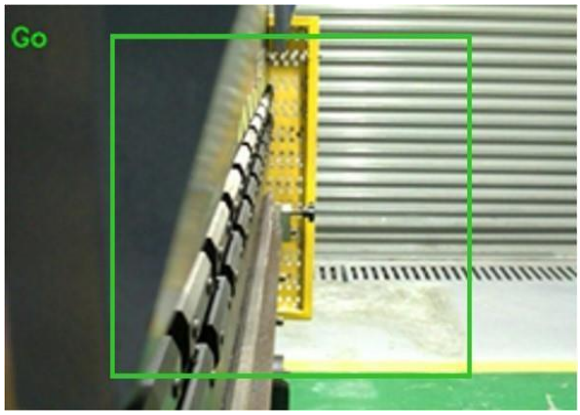

Figura 44 Funcionamiento seguro de la dobladora CNC



Figura 45 Paro de emergencia de la dobladora CNC

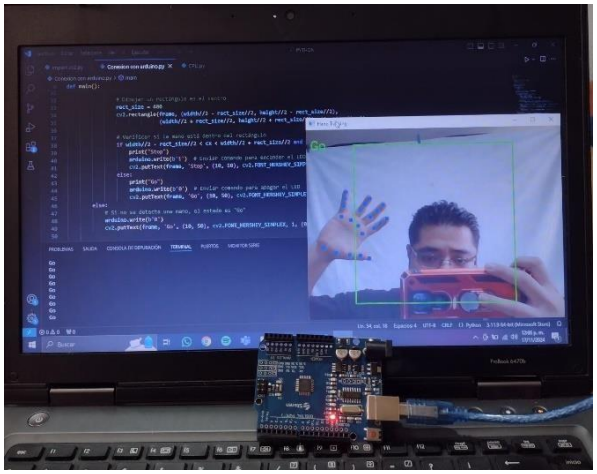

Figura 46 Conexión Arduino marcando “go”

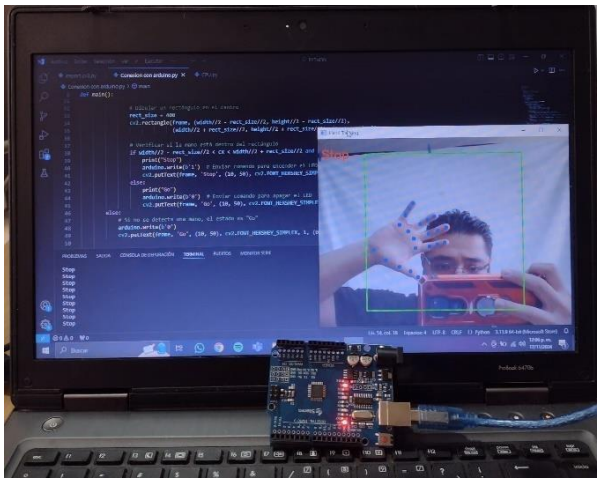

Figura 47 Conexión Arduino marcando “stop” y activando paro de emergencia

Capítulo V.
Conclusiones

En resumen, el estudio abordó de manera exitosa la implementación de un sistema de detección de manos mediante el uso de visión artificial, haciendo uso de las bibliotecas OpenCV y MediaPipe. La motivación principal se centró en elevar los estándares de seguridad y capacidad de respuesta en entornos críticos, resaltando la importancia de identificar gestos manuales como medio para activar un protocolo de paro de emergencia. A pesar de los logros alcanzados, se reconocieron limitaciones, como la sensibilidad a condiciones lumínicas.

La metodología empleada demostró su eficacia al fusionar algoritmos de procesamiento de imágenes con técnicas de aprendizaje profundo, logrando un reconocimiento preciso de los gestos. La estructura del trabajo permitió abordar estas limitaciones, proporcionando secciones específicas para la discusión de desafíos, la propuesta de mejoras y la destacada relevancia práctica del sistema desarrollado.

Finalmente, el sistema propuesto emerge como una contribución significativa para mejorar la seguridad en entornos críticos, presentando una solución práctica y efectiva para la detección de manos mediante el reconocimiento de patrones clave dentro de un área delimitada que se considera no segura, disparando un mecanismo de paro de emergencia. El corte de señal en el pedal fue realizado mediante un relevador, activado por un transistor y controlado por un Arduino nano que recibe una señal desde el puerto serial a través de la computadora y procesada por medio de Python. El prototipo aún sigue en desarrollo, se pretende como trabajo futuro mejorar el algoritmo de detección, así como, reemplazar la computadora por una Raspberry Pi.

Referencias

[1] IBM, «IBM,» 04 07 2024. [En línea]. Available: https://www.ibm.com/mx-es/topics/computer-vision#:~:text=La%20visi%C3%B3n%20artificial%20es%20un,se%20identifican%20defectos%20o%20problemas..

[2] Iberdrola, «IBERDROLA,» 22 07 2024. [En línea]. Available: https://www.iberdrola.com/innovacion/vision-artificial.

[3] E. Blog, «Educalix Blog,» 08 12 2024. [En línea]. Available: https://blog.educalix.com/aprende-a-crear-una-aplicacion-web-de-deteccion-de-objetos-con-tensorflow-opencv-y-flask/#google_vignette.

[4] Atria, «Atria,» 02 12 2024. [En línea]. Available: https://atriainnovation.com/blog/aplicaciones-de-la-vision-artificial-en-la-industria/.

[5] F. C. S.L., «Freepik Company S.L.,» 03 12 2024. [En línea]. Available: https://www.freepik.es/imagen-ia-premium/primer-plano-lente-camara-entrenada-peatones-analizando-su_38373762.htm.

[6] D. Calvo, «Diego Calvo,» 28 11 2024. [En línea]. Available: https://www.diegocalvo.es/red-neuronal-convolucional/#google_vignette.

[7] G. Solano, «Omes,» 08 09 2024. [En línea]. Available: https://omes-va.com/mediapipe-hands-python/.

[8] G. Solano, «OMES,» 12 09 2024. [En línea]. Available: https://omes-va.com/mediapipe-hands-python/.

[9] M. Devices, «Moleculas de Vices,» 22 07 2024. [En línea]. Available: https://es.moleculardevices.com/products/cellular-imaging-systems/high-content-analysis/in-carta-image-analysis-software.

[10] A. W. Services, «AWS,» 08 12 2024. [En línea]. Available: https://aws.amazon.com/es/what-is/sdk/.

[11] Prysmex, «Prysmex,» 23 08 2024. [En línea]. Available: https://www.prysmex.com/blog/seguridad-industrial-que-es-y-para-que-sirve#:~:text=La%20seguridad%20industrial%20comprende%20el,industriales%20como%20plantas%20o%20f%C3%A1bricas..

[12] A. 21, «Centro de formacion para la industria,» 02 12 2024. [En línea]. Available: https://www.cursosaula21.com/que-es-la-vision-artificial/.

[13] SEGUSA, «SEGUSA,» 02 12 2024. [En línea]. Available: https://www.segusa.com.mx/equipo-de-proteccion-personal-para-industrias-especificas/.

[14] Tsinfa, «Tsinfa,» 19 01 2025. [En línea]. Available: https://www.tsinfa.com/es/hydraulic-sheet-bending-machine-heavy-duty-china-manufacturer/.

[15] Metalmecanica, «Metalmecanica,» 31 07 2024. [En línea]. Available: https://www.metalmecanica.com/es/noticias/dobladoras-de-lamina-que-son-y-por-que-son-esenciales-en-la-industria.

[16] Prometec, 22 09 2024. [En línea]. Available: https://www.prometec.net/intro-programacion/.

[17] Prometec, «Prometec programacion,» 17 11 2024. [En línea]. Available: https://www.prometec.net/intro-programacion/.

[18] Cilsa Corporation , «Cilsa,» 11 05 2024. [En línea]. Available: https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-lenguaje-de-programacıon/.

[19] Stack, «Stack,» 02 06 2024. [En línea]. Available: https://es.stackoverflow.com/tags/pylance/info#:~:text=Pylance%20es%20un%20servidor%20de,Python%20para%20Visual%20Studio%20Code.

[20] G. Solano, «OMES,» 24 11 2024. [En línea]. Available: https://omes-va.com/basicvideo/.

[21] T. Service, «TRBL Service,» 24 11 2024. [En línea]. Available: https://trbl-services.eu/blog-opencv-y-deep-learning-vision-por-computador-en-redes-neuronales/.

[22] A. W. Services, «AWS,» 08 12 2024. [En línea]. Available: https://aws.amazon.com/es/what-is/sdk/.

ANEXO 1

Datos técnicos - Dobladora Hidráulica De Cortina CNC WC67K-400/6000.



Características

1. No. de golpes por minuto: Mayor o igual a 2.5.
2. El tope trasero es construído con husillos de bolas sobre guías lineales de alta calidad.
3. Control numérico de 2 ejes con opción de agregar más ejes de acuerdo a sus necesidades.
4. Control ESTUN (estándar).
5. Control DELEM (opcional).

Ficha técnica

1. Motor 50 Hp
2. Profundidad De Garganta400 Mm
3. Peso 41,100 Kg
4. Presión De Cierre 400 T
5. Longitud De La Mesa 6,000 Mm
6. Ajuste De La Carrera 220 Mm
7. Dimensiones (Largo X Ancho X Alto) 6,150 X 3,250 X 3,900 Mm
8. Distancia Entre Bastidores 5,000 Mm
9. Carrera De La Cortina 320 Mm
10. Altura Máxima De La Cortina600 Mm

ANEXO 2

Divulgación científica

1. Mario Alberto Hernández Soriano, Diego Ignacio Cruz Sánchez, Diego Rodríguez Espinosa, Ian Omar Madrigal Rueda, “Sistema de monitoreo por visión artificial para la activación de un sistema de paro de emergencia en una dobladora CNC”, Ponencia, “7º Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología (CEIAAIT)”, Facultad de Estudios Superiores de Cuautitlán, UNAM, 6 de noviembre de 2024.

2. Mario Alberto Hernández Soriano, Diego Ignacio Cruz Sánchez, Diego Rodríguez Espinosa, Ian Omar Madrigal Rueda, “Sistema de monitoreo por visión artificial para la activación de un sistema de paro de emergencia en una dobladora CNC”, Publicación artículo en las memorias del “7º Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología (CEIAAIT)”, Facultad de Estudios Superiores de Cuautitlán, UNAM, 6 de noviembre de 2024.

Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Cuautitlán
Departamento de Ingeniería

Otorgan la presente

Constancia

A: Mario Alberto Hernández Soriano, Diego Ignacio Cruz Sánchez, Diego Rodríguez Espinosa, Ian Omar Madrigal Rueda

Por su participación como ponente de la conferencia
Sistema de monitoreo por visión artificial para la activación de un sistema de paro de emergencia en una dobladora CNC
En el **7º Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología (CEIAAIT)** realizado el 6 y 7 de noviembre de 2024.

“POR MI RAZA HABLARÁ EL ESPÍRITU”
Cuautitlán Izcalli, Estado de México, noviembre de 2024.

Folio: 202400562

Dr. David Quintanar Guerrero
Director

#SíCuentaLa DOCENCIA

Sistema de monitoreo por visión artificial para la activación de un sistema de paro de emergencia en una dobladora CNC

Mario Alberto Hernández Soriano
Departamento de Ingeniería Mecatrónica
Tecnológico de Estudios Superiores de Coacalco
Av.16 de septiembre No. 54, Coacalco de Berriozábal, Estado de México
mario_mct@tesco.edu.mx

Diego Ignacio Cruz Sánchez
Departamento de Ingeniería Mecatrónica
Tecnológico de Estudios Superiores de Coacalco
Av.16 de septiembre No. 54, Coacalco de Berriozábal, Estado de México
diego_ignacio.mtc@tesco.edu.mx

Diego Rodríguez Espinosa
Departamento de Ingeniería Mecatrónica
Tecnológico de Estudios Superiores de Coacalco
Av.16 de septiembre No. 54, Coacalco de Berriozábal, Estado de México
202120811@tesco.edu.mx

Ian Omar Madrigal Rueda
Departamento de Ingeniería Mecatrónica
Tecnológico de Estudios Superiores de Coacalco
Av.16 de septiembre No. 54, Coacalco de Berriozábal, Estado de México
ian_omar.mtc@tesco.edu.mx

***Resumen*—En este trabajo se presenta la propuesta de un sistema de monitoreo por medio de herramientas de visión artificial con el objetivo de activar un sistema de paro de emergencia. La metodología desarrollada emplea el lenguaje de programación Python y las librerías de OpenCV y MediaPipe para detectar si las manos de un operador están en una zona segura o insegura dentro del área de trabajo de una dobladora CNC. Los resultados muestran cómo funciona el sistema de monitoreo garantizando la seguridad del operador y mostrando la importancia de integrar nuevas tecnologías en el monitoreo industrial.**

***Keywords*—Python, OpenCV, MediaPipe, Dobladora CNC**

I. Introducción

En el ámbito de la industria, la seguridad de los trabajadores es una prioridad que no puede ser comprometida, los accidentes laborales no solo conllevan riesgos físicos y emocionales para los empleados, sino que también implican costos significativos para las empresas, tanto en términos económicos como de reputación. A medida que la automatización y la maquinaria avanzada se integran en los procesos de producción, la necesidad de sistemas de seguridad eficientes se vuelve aún más crucial. La implementación de tecnologías innovadoras, como la visión artificial, está revolucionando la forma en que se aborda la seguridad industrial, proporcionando herramientas precisas y confiables para prevenir accidentes y mejorar la vigilancia en tiempo real.

La visión artificial, es una rama de la inteligencia artificial que tiene sus raíces en la década de 1960, cuando los primeros sistemas comenzaron a ser desarrollados para interpretar imágenes y reconocer patrones. Estos primeros intentos se limitaban a aplicaciones simples y requerían una gran cantidad de procesamiento manual. Sin embargo, con el avance de la informática y la disponibilidad de hardware más potente, la visión artificial ha evolucionado significativamente. Hoy en día, se emplean algoritmos avanzados y técnicas de aprendizaje automático que permiten a las máquinas "ver" y "entender" su entorno de manera autónoma y eficiente.

Las aplicaciones de la visión artificial en la industria son numerosas y diversas. En el control de calidad, por ejemplo, permite la inspección minuciosa de productos en líneas de ensamblaje, detectando defectos que podrían pasar desapercibidos para el ojo humano. En áreas como la automatización, facilita la guía de robots en tareas complejas, como el montaje de componentes electrónicos o la clasificación de materiales. Además, en la gestión de almacenes, la visión artificial se utiliza para identificar y rastrear inventarios, optimizando el flujo de materiales y reduciendo errores en el proceso.

En el contexto de la seguridad industrial, la visión artificial se ha convertido en una herramienta indispensable. Los sistemas de monitoreo equipados con cámaras y sensores pueden vigilar continuamente áreas de trabajo, detectando movimientos inusuales y potencialmente peligrosos. Esto es especialmente útil en entornos donde el riesgo de accidentes es alto, como en la manipulación de maquinaria pesada o en procesos con sustancias peligrosas. Al integrarse con sistemas de alarma y mecanismos de paro automático, la visión artificial no solo previene accidentes, sino que también puede intervenir activamente para proteger a los trabajadores.

II. Marco teórico

A. Conceptos básicos de la visión artificial

La inteligencia artificial también conocida como IA de acuerdo con el diccionario de la real academia española menciona que, la IA “es la diciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o razonamiento lógico” [1], [2], [3].

En [4], [5], [6] y [7] podemos encontrar las ramas más importantes de la inteligencia artificial, como: Machine Learning, Deep Learning, Computer Vision, Neural Networks, Natural Language Processing, Natural Language