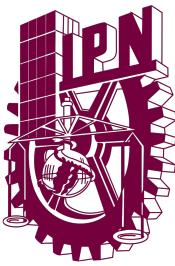


Bitácora de Cambios

Fecha	Revisión	Descripción del cambio
03/04/2025		Corrección de estilo en el capítulo 2.
04/04/2025		Se añade la sección de Gaze and Depth.
10/04/2025		Se añade la sección 5.2 y 5.2.1 con sus respectivas imágenes

Tabla 2: Cronograma de escritura de tesis por semanas

Semana	Capítulo/Sección	Actividad de Escritura
1	Introducción	Redacción del capítulo de introducción: planteamiento del problema, justificación, objetivos y alcance.
2	Marco Teórico (Objetivo 1)	Escritura del marco teórico sobre distracción visual y somnolencia en conductores.
3	Marco Teórico (Objetivo 2)	Escritura del marco teórico sobre visión artificial para monitoreo del entorno vehicular.
4	Marco Teórico (Objetivo 3)	Redacción sobre sistemas de alerta sonora y seguridad activa en vehículos.
5	Metodología	Desarrollo del capítulo de metodología: descripción general del sistema, tecnologías utilizadas y estructura general.
6	Implementación	Redacción del proceso de implementación para cada módulo: seguimiento de mirada, entorno y sistema de alerta.
7	Resultados	Escritura del capítulo de resultados: pruebas realizadas, condiciones del experimento y datos recolectados.
8	Discusión	Redacción del análisis de resultados, comparación con trabajos previos y discusión crítica.
9	Conclusiones y Revisión Final	Escritura de conclusiones, propuestas de trabajo futuro y revisión integral del documento completo.



Instituto Politécnico Nacional

**Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada
Unidad-Querétaro**

Sistema de Monitoreo Automovilístico Basado en Aprendizaje Automático para la Detección de Distracciones del Conductor y Obstáculos Externos

T E S I S

Maestría en Tecnología Avanzada

PRESENTA:

Claudia Beatriz Resendiz Jurado

Estados Unidos Mexicanos
Santiago de Querétaro, Qro.
2024



Índice general

0.1	Abstract	1
1	Planteamiento del problema	2
2	Estado del arte	3
3	Solución propuesta	5
4	Marco Teórico	6
4.1	Uso de LiDAR en la detección de objetos en carretera	6
4.2	LiDAR A1M8	6
4.2.1	Características del LiDAR A1M8	6
4.3	Conceptos básicos de profundidad estéreo	7
4.3.1	Calibración de Cámara para Estimación de Profundidad	7
4.3.2	Cálculo de Profundidad (Cámaras Estéreo)	8
5	Experimentos y Resultados	11
5.0.1	Pruebas en Python LiDAR A1M8	11
5.0.2	Pruebas en RoboStudio 2.0.0	11
5.0.3	Pruebas con cámara de profundidad	12
5.1	Modelo de Seguimiento de Mirada y Estimación de Profundidad	16
5.1.1	Seguimiento de la Mirada con GazeL2CS	16
5.1.2	Estimación de Profundidad con Depth Pro	16
5.1.3	Integración de GazeL2CS y Depth Pro	16
5.2	Visualización de Resultados en Pantalla	17
5.2.1	Dibujo de la Flecha de Mirada	18
6	Conclusiones	27

0.1. Abstract

Capítulo 1

Planteamiento del problema

En México y el mundo los accidentes automovilísticos ocurren todos los días , y esto es causante de lesiones graves y perdidas humanas. Muchas veces estas situaciones son causadas por la distracción del conductor como puede ser enviar mensajes de texto ,recibir llamadas, ver el la pantalla del GPs o alguna distracción del copiloto que afecte al conductor incluso los pensamientos o estrés del conductor lo distraen y se puede llegar a la consecuencia de producir un accidente. Estas situaciones reducen la atención y la capacidad de reacción del conductor. La Organización Mundial de la Salud (OMS) asegura que el 20 % de accidentes de tráfico son debido a estas distracciones (WHO, 2015).

El ambiente representa una parte crítica también ya que si se tiene escasa iluminación o mal clima existe mayor dificultad para identificar una persona delante del vehículo o algún obstáculo que represente un peligro en la vía. Un informe de la Comisión Europea indica que las condiciones meteorológicas adversas pueden aumentar el riesgo de accidentes en un 30 % (European Commission, 2020).

Capítulo 2

Estado del arte

Se realizo la búsqueda de artículos relacionados con el tema en ACM Digital library, se encontró un proyecto de investigación sobre detección de Peatones Usando Machine Learning donde el proyecto busca desarrollar un sistema de monitoreo de visión para detectar peatones y obstáculos mientras un vehículo se mueve en reversa. Aunque este artículo hable sobre la detección de reversa y nosotros planteamos que la detección es al frente estamos hablando de las mismas características de detección. Se utiliza una combinación de arquitecturas de aprendizaje profundo: Inception V3 para el análisis de imágenes y CNN 1D para procesar señales de sensores de proximidad. Se adquirieron 75,440 imágenes y datos de distancia utilizando una cámara y cuatro sensores ultrasónicos instalados en vehículos. Los datos se recogieron en diferentes entornos, y las imágenes se clasificarán en dos categorías: con peatones y sin peatones. Para el procesamiento las imágenes y los datos de distancia se limpian y se dividieron en conjuntos de entrenamiento (80 %) y prueba (20 %). Después de entrenar y validar el modelo, se realizaron predicciones introduciendo datos de entrada (imágenes y distancias) para determinar la presencia de peatones. En los resultados se obtuvo un resultado exitoso en la clasificación de imágenes con y sin peatones.[1]

En la recopilación de artículos relacionados se encuentra la detección de objetos mediante el uso de radares, se tomó este artículo ya que se tendría una opción de usar sensor Lidar en lugar de cámaras. Este artículo habla sobre la predicción de colisiones mediante el uso de radares de onda continua modulada en frecuencia (FMCW) y detección inercial, dirigido a aplicaciones de seguridad en vehículos autónomos y cascos inteligentes en deportes de contacto. La clave del sistema radica en convertir las mediciones de radar en matrices de alcance-Doppler (RDM) y entrenar una red neuronal convolucional profunda para clasificar situaciones como colisión inminente o sin colisión.

El contexto de la investigación destaca la importancia de predecir colisiones antes de que ocurran. El enfoque propuesto utiliza aprendizaje profundo para adaptarse a condiciones cambiantes y mejorar la precisión de las predicciones. Se enfatiza la necesidad de un sistema de detección que opere independientemente de otras redes y que utilice información cinemática relativa. El radar FMCW se elige por su capacidad de detectar objetos sin requerir sensores adicionales. A diferencia de métodos tradicionales que enfrentan desafíos debido al ruido y desorden en las mediciones, el enfoque basado en aprendizaje profundo permite un modelo adaptable que puede actualizarse en tiempo real para mantenerse efectivo en entornos dinámicos. El estudio demuestra la eficacia del sistema mediante experimentos en el mundo real, alcanzando una puntuación F1 de 0,91, superando a métodos tradicionales en precisión y adaptabilidad. En resumen, este trabajo presenta una solución prometedora para mejorar la seguridad mediante la predicción anticipada de colisiones en diversas aplicaciones.[2]

En el trabajo presentado en [2] los autores enfatizan la importancia de detectar al peatón. El texto destaca la importancia de la conducción autónoma en la intersección de la robótica y el aprendizaje profundo, enfatizando que el sistema de percepción es crucial para una comprensión integral del entorno y la toma de decisiones de conducción. Se menciona que los peatones son un elemento significativo en los conjuntos de datos relacionados con la conducción autónoma, lo que ha llevado a un enfoque creciente en tareas centradas en el ser humano, como la detección, reidentificación y predicción de la trayectoria de peatones. La detección de peatones se considera una tarea fundamental en aplicaciones del mundo real, ya que busca identificar instancias de peatones y predecir sus ubicaciones en imágenes o videos, lo que requiere alta precisión y eficiencia. Se señala que la detección de video puede aprovechar el contexto temporal para mejorar la velocidad y precisión de la detección, abordando desafíos como el desenfoque de movimiento y la occlusión. En la última década, la detección de objetos, incluida la detección de peatones, ha avanzado notablemente gracias a técnicas de aprendizaje profundo, logrando altos rendimientos en conjuntos de datos reconocidos como ImageNet, Pascal VOC y MS COCO.[3]

Agregue table.

Paper	Autor	Fecha	Método Aplicado	Limitaciones
Deep Learning-based eye tracking system to detect distracted driving	Son Xin, Sho Zhang,Wan Rong Xu	20 Mayo 2024	<ul style="list-style-type: none"> • Seguimiento ocular • YoloV5 para detección de objetos 	No considera cambio de estado del semáforo
Traffic Object Detection and Recognition Based on the Attentional Visual Field of Drivers	Mohsen Shirpour, Nima Khairdoost , Michael Bauer	31 Diciembre 2022	<ul style="list-style-type: none"> • Utilizan modelos multiescala basados en HOG-SVM y Faster R-CNN 	Limitado a cuatro clases de objetos de tráfico
Detection of distracted driving through the analysis of real-time driver, and roadway volatilities	Seik M Usman, Asad J.	16 Abril 2024	Análisis en la mirada del conductor , velocidad del vehículo , distancia del vehículo a la línea central del carril y el vehículo siguiente.	No considera somnolencia en el conductor.

Figura 2.1: Comparación de Antecedentes.

Capítulo 3

Solución propuesta

Para resolver el problema se empleará una cámara de profundidad para la detección temprana de posible colisión con un peatón; para eso, se obtiene la distancia al peatón respecto al vehículo para dar tiempo de reacción.

Dentro de la característica para la detección de peatones:

1. Contornos o siluetas de personas



Figura 3.1: Obtención de Contornos de la imagen.

Capítulo 4

Marco Teórico

4.1. Uso de LiDAR en la detección de objetos en carretera

El LiDAR (Light Detection and Ranging) es una tecnología que utiliza láseres pulsados para medir distancias precisas hacia objetos en el entorno. En el contexto de la conducción autónoma y los sistemas avanzados de asistencia al conductor (ADAS), el LiDAR se usa para crear mapas tridimensionales del entorno, permitiendo la detección de vehículos, peatones, ciclistas y otros obstáculos en la carretera. La alta precisión y la capacidad de operar en diversas condiciones de iluminación lo convierten en una herramienta clave para la navegación y la seguridad en carretera. Los sistemas LiDAR generalmente se componen de un emisor láser, un detector y un sistema de procesamiento que interpreta los datos en tiempo real.

El LiDAR se destaca por su capacidad para medir distancias con alta precisión y generar nubes de puntos en 3D del entorno. En aplicaciones viales, el LiDAR puede detectar objetos a largas distancias y a velocidades relativas elevadas, siendo una tecnología fundamental en la percepción para la conducción autónoma [4].

4.2. LiDAR A1M8

El LiDAR A1M8 es un sensor de tecnología LiDAR diseñado para aplicaciones de detección y mapeo en entornos complejos. Este dispositivo es conocido por su capacidad de proporcionar datos de alta resolución en 3D, lo que es esencial para la detección precisa de objetos y la creación de modelos detallados del entorno. El A1M8 se utiliza comúnmente en aplicaciones como vehículos autónomos, mapeo topográfico, inspección de infraestructuras y monitoreo ambiental.

4.2.1. Características del LiDAR A1M8

Algunas características clave del LiDAR A1M8 incluyen:

- **Alta frecuencia de muestreo:** Permite capturar datos en tiempo real y con alta precisión, lo que mejora la detección de objetos en movimiento, como vehículos y peatones.
- **Rango de detección extendido:** Capaz de operar a distancias significativas, lo que es crucial para la seguridad en carretera y la navegación autónoma.
- **Resistencia a condiciones ambientales adversas:** Diseñado para funcionar en una variedad de condiciones climáticas, lo que lo hace adecuado para su uso en exteriores.

- **Integración con sistemas de procesamiento de datos:** Facilita la interpretación de datos y la creación de mapas 3D en tiempo real.

4.3. Conceptos básicos de profundidad estéreo

La visión de profundidad estéreo funciona calculando la disparidad entre dos imágenes tomadas desde puntos ligeramente diferentes. La visión estereoscópica funciona de forma muy parecida a nuestros ojos. Nuestros cerebros (subconscientemente) estiman la profundidad de los objetos y escenas en función de la diferencia entre lo que ve nuestro ojo izquierdo y lo que ve nuestro ojo derecho. En las cámaras OAK-D, es exactamente lo mismo; tenemos cámaras izquierda y derecha (del par de cámaras estéreo) y el OAK hace coincidencias de disparidades en el dispositivo para estimar la profundidad de los objetos y las escenas. La disparidad se refiere a la distancia entre dos puntos correspondientes en la imagen izquierda y derecha de un par estéreo. [4].

4.3.1. Calibración de Cámara para Estimación de Profundidad

La **calibración de cámara** es el proceso de determinar los parámetros intrínsecos, extrínsecos y de distorsión de una cámara. Este proceso permite:

- Mapear puntos del mundo 3D a puntos en una imagen 2D.
- Corregir distorsiones en las imágenes.
- Calcular la profundidad a partir de pares estéreo.

Parámetros de Calibración

La calibración incluye los siguientes parámetros:

Parámetros Intrínsecos Relacionan el sistema de coordenadas 3D con el plano de la imagen:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Donde:

- f_x, f_y : Longitud focal en píxeles.
- c_x, c_y : Coordenadas del centro óptico (punto principal).

Parámetros Extrínsecos Describen la posición y orientación de la cámara en el espacio 3D:

$$[R | t]$$

Donde:

- R : Matriz de rotación.
- t : Vector de traslación.

Modelo de Distorsión Corrige distorsiones radiales y tangenciales:

$$x_{\text{corr}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2)$$

$$y_{\text{corr}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2xy$$

Donde:

- k_1, k_2, k_3 : Coeficientes de distorsión radial.
- p_1, p_2 : Coeficientes de distorsión tangencial.

Preparación de la Calibración

Tablero de Ajedrez Utiliza un tablero de ajedrez para capturar imágenes, definiendo:

- Tamaño de cada cuadrado en cm (SQUARE_SIZE).
- Número de cuadros en las direcciones X y Y (NX, NY).

Captura de Imágenes

Captura imágenes desde diferentes distancias y ángulos:

- **Cercano al tablero:** El tablero debe cubrir casi todo el campo visual (FOV).
- **Mediana distancia:** El tablero debe cubrir aproximadamente el 40 % del FOV.
- **Distancia lejana:** El tablero cubre una pequeña parte del FOV.

Procesamiento de Calibración

Minimización del Error

El error de reproyección se calcula como:

$$\text{Error de Reproyección} = \frac{1}{N} \sum_{i=1}^N \|p_i^{\text{observado}} - p_i^{\text{proyectado}}\|^2$$

Cálculo de Matrices de Rectificación

Se obtienen las matrices para alinear las imágenes estéreo:

$$R_{\text{rect}}, T_{\text{rect}}$$

4.3.2. Cálculo de Profundidad (Cámaras Estéreo)

La profundidad (Z) se calcula a partir de la disparidad (d):

$$Z = \frac{f_x \cdot B}{d}$$

Donde:

- f_x : Longitud focal en píxeles.
- B : Distancia entre las cámaras.
- d : Disparidad (diferencia entre las coordenadas x en las imágenes izquierda y derecha).

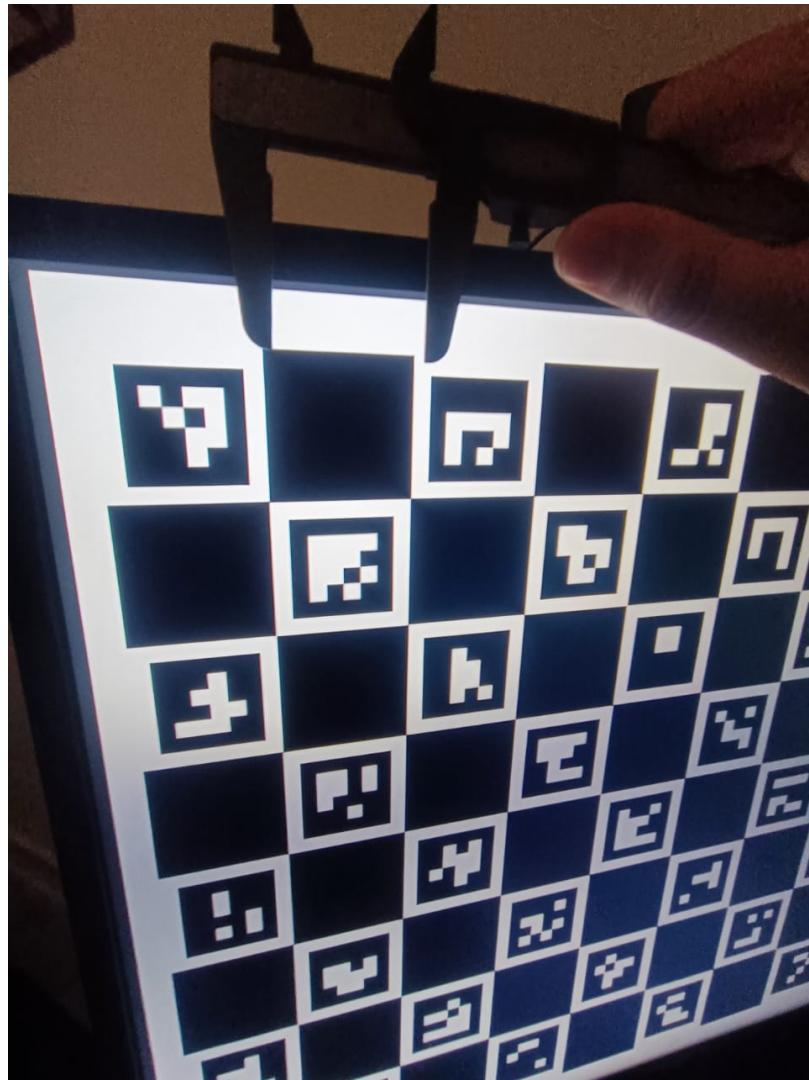


Figura 4.1: Proceso de Calibración de Cámara stereo.

Comandos y Scripts

```
git clone https://github.com/luxonis/depthai.git  
cd depthai  
git submodule update --init --recursive  
python3 install_requirements.py
```

Ejecución de la Calibración

Conforme al dato obtenido al medir con vernier el cuadro de ajedrez metemos el dato en el campo SQUARE SIZE IN CM, se cuentan los cuadros en el eje X y Y.

```
python3 calibrate.py -s [SQUARE_SIZE_IN_CM] --board [BOARD] -nx [NX] -ny [NY]
```

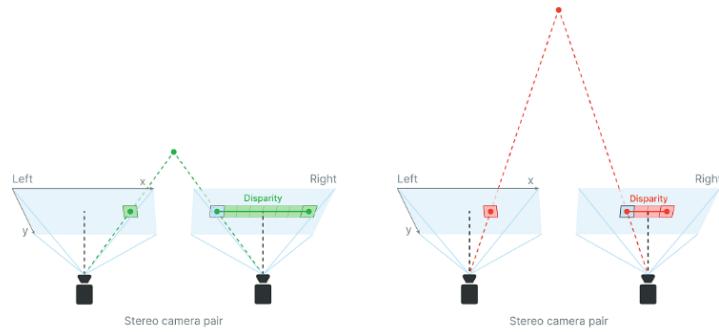


Figura 4.2: Stereo camera pair.

Profundidad de la disparidad

$$\text{Profundidad [mm]} = f_x[\text{px}] \frac{\text{línea de base [mm]}}{\text{Disparidad [px]}}$$

- depth cm - profundidad en centímetros
- f_x px - Distancia focal en píxeles
- baseline cm - Distancia entre dos cámaras del par de cámaras estéreo
- disparity px - Disparidad en píxeles

Distancia focal

La distancia focal es la distancia entre la lente de la cámara y el sensor de imagen. Cuanto mayor sea la distancia focal, más estrecho será el campo de visión.

Disparidad

La disparidad y la profundidad están inversamente relacionadas. A medida que la disparidad disminuye, la profundidad aumenta exponencialmente dependiendo de la línea de base y la distancia focal. Es decir, si el valor de la disparidad es cercano a cero, entonces un pequeño cambio en la disparidad genera un gran cambio en la profundidad. Del mismo modo, si el valor de la disparidad es grande, entonces algún cambio en la disparidad no conduce a un gran cambio en la profundidad (mejor precisión).

Capítulo 5

Experimentos y Resultados

Se realizaron pruebas exhaustivas con el sensor LiDAR A1M8 utilizando dos entornos diferentes: Python y el software RoboStudio 2.0.0. Estas pruebas tenían como objetivo evaluar la precisión y la eficacia del LiDAR en la detección de objetos en entornos simulados y en tiempo real.

5.0.1. Pruebas en Python LiDAR A1M8

Utilizando Python, se desarrolló un script que permite interactuar con el LiDAR A1M8 y recoger datos en tiempo real. Se emplearon bibliotecas como NumPy y Matplotlib para procesar y visualizar los datos obtenidos. Durante las pruebas, se registraron las coordenadas de varios objetos en el entorno y se generó un mapa en 2D de la escena detectada.

5.0.2. Pruebas en RoboStudio 2.0.0

El software RoboStudio 2.0.0 se utilizó para simular un entorno de prueba en el que se integró el LiDAR A1M8. Esta plataforma permitió visualizar en tiempo real la detección de objetos y analizar el rendimiento del sensor en diversas condiciones. Se realizaron diferentes configuraciones de prueba para observar cómo el LiDAR respondía a cambios en la disposición de los objetos y a variaciones en la iluminación.

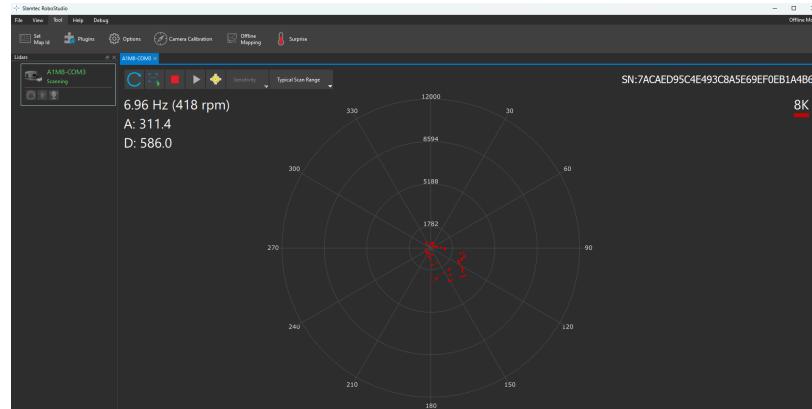


Figura 5.1: Resultados de la detección de objetos con el LiDAR A1M8.

5.0.3. Pruebas con cámara de profundidad

Se implementaron funciones:

1. Prueba de cámara RGB y de Profundidad.



Figura 5.2: Imagen RGB.

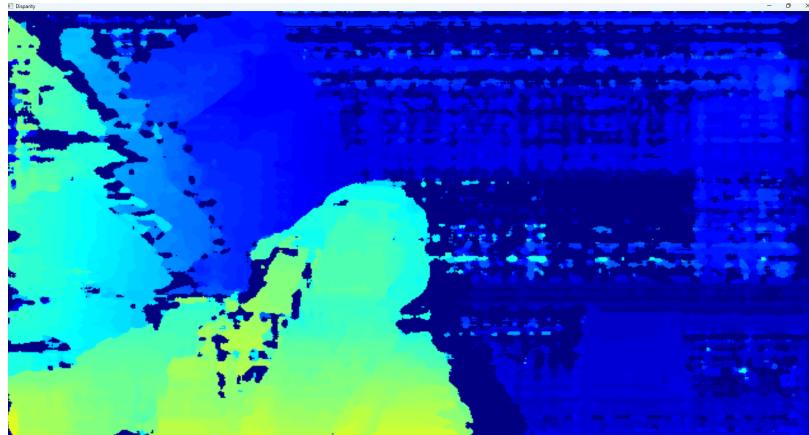


Figura 5.3: Imagen de profundidad.

2. Alineación de cámara RGB con la cámara stereo:

La alineación entre una cámara RGB y un par estéreo implica transformar los datos de profundidad obtenidos de la cámara estéreo para que coincidan espacialmente con la vista de la cámara RGB. Este proceso es fundamental para tareas como la fusión de imágenes RGB y profundidad.

Cada cámara tiene una matriz intrínseca que relaciona coordenadas 3D del mundo con coordenadas 2D en la imagen:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Donde:

- f_x, f_y : Longitud focal en píxeles.

- c_x, c_y : Coordenadas del centro óptico.

La posición y orientación de la cámara RGB respecto a la cámara estéreo se describe mediante una matriz de rotación (R) y un vector de translación (t):

$$[R | t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

Los puntos de profundidad del marco estéreo se proyectan al marco RGB mediante:

$$P_{\text{RGB}} = K_{\text{RGB}} \cdot (R \cdot P_{\text{Depth}} + t)$$

Donde:

- P_{Depth} : Coordenadas 3D en el sistema estéreo.
- K_{RGB} : Matriz intrínseca de la cámara RGB.
- R, t : Transformación extrínseca.

El proceso de Implementación es el siguiente, se calibra cada cámara para obtener sus parámetros intrínsecos y extrínsecos, utilizando un tablero de ajedrez.

Para la alineación de Marcos. Primero se calcula una matriz de transformación homogénea para proyectar coordenadas del espacio estéreo al espacio RGB. Despues se ajusta la resolución para que la profundidad coincida con el tamaño de la imagen RGB.

Para la proyección de Puntos Estéreo a RGB Dado un punto en coordenadas estéreo $P_{\text{Depth}} = (x, y, z)$, su transformación al marco RGB es:

$$P_{\text{RGB}} = \begin{bmatrix} u_{\text{RGB}} \\ v_{\text{RGB}} \\ 1 \end{bmatrix} = K_{\text{RGB}} \cdot \left(R \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t \right)$$

La profundidad (z) se obtiene de la disparidad (d):

$$z = \frac{f_x \cdot B}{d}$$

Donde:

- f_x : Longitud focal.
- B : Distancia entre cámaras estéreo.
- d : Disparidad entre imágenes izquierda y derecha.

Resultados:

3. Detección de personas con el modelo tiny Yolo.

El modelo Tiny YOLO es una variante eficiente de YOLO diseñada para dispositivos con recursos limitados. Su integración con cámaras OAK permite la detección de personas en tiempo real, aprovechando las capacidades de procesamiento de inteligencia artificial en el dispositivo.

Tiny YOLO es una red neuronal convolucional optimizada para detección de objetos. Ofrece un compromiso entre precisión y velocidad. Su arquitectura utiliza menos capas y filtros que el modelo completo de YOLO, reduciendo la carga computacional. La arquitectura de Tiny YOLO incluye:

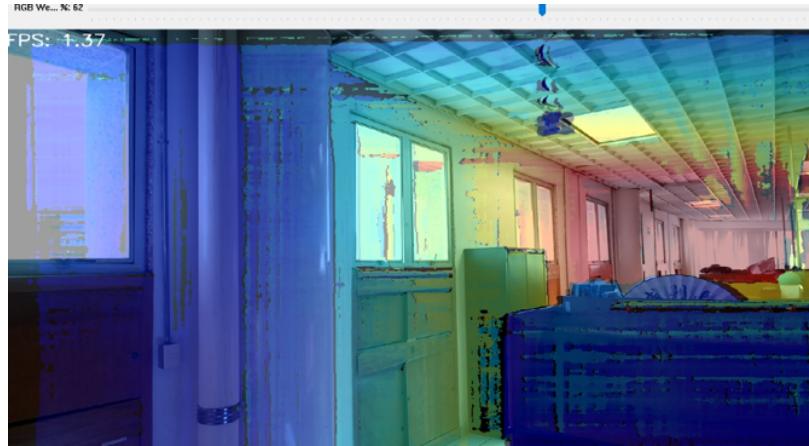


Figura 5.4: Alineación RGB y Profundidad.

- Capas convolucionales ligeras para la extracción de características.
- Capas de activación Leaky ReLU.
- Una capa final de detección basada en regresión para predicción de bounding boxes.

Tiny YOLO utiliza bounding boxes y anclas para detectar objetos. Cada cuadro delimitador predice:

$$\text{Bounding Box} = (x, y, w, h)$$

Donde:

- x, y : Coordenadas del centro del cuadro.
- w, h : Ancho y altura del cuadro.

El pipeline de la cámara OAK incluye los siguientes nodos:

- **ColorCamera**: Captura imágenes RGB.
- **YoloDetectionNetwork**: Ejecuta el modelo Tiny YOLO.
- **XLinkOut**: Envía las detecciones al host.

Se necesita descargar los siguientes archivos:

- Descarga un modelo preentrenado de Tiny YOLO.
- Documento COCO Names.

A continuación se muestra un diagrama del funcionamiento del programa para las detecciones.

El modelo Tiny YOLO identifica personas en tiempo real. Cada detección incluye:

- **Etiqueta**: Clase del objeto detectado (e.g., persona).
- **Confianza**: Valor entre 0 y 1 que indica la certeza de la detección.
- **Coordenadas**: Bounding box en el marco RGB.

Métricas de Rendimiento

- Velocidad de procesamiento: ~ 30 FPS.

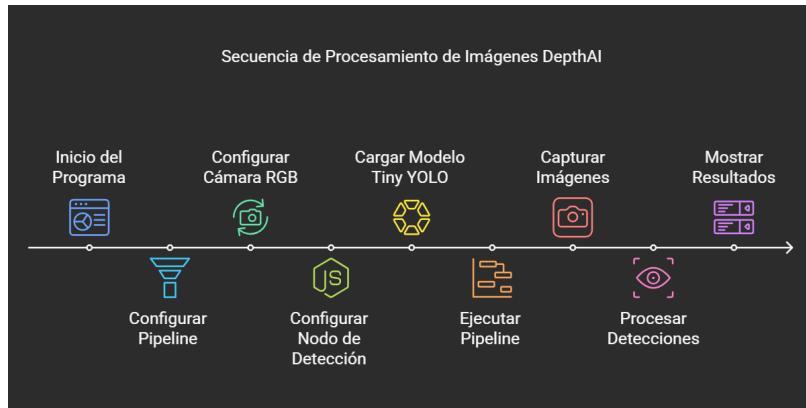


Figura 5.5: Diagrama funcionamiento Yolo.

- Precisión: > 90 % en detecciones de personas con iluminación adecuada.

La integración de Tiny YOLO con la cámara OAK permite una detección eficiente y en tiempo real de personas. Esta capacidad es útil para aplicaciones como vigilancia, conteo de personas y sistemas de seguridad.

Pruebas con Yolo y imagen de profundidad para obtener la distancia

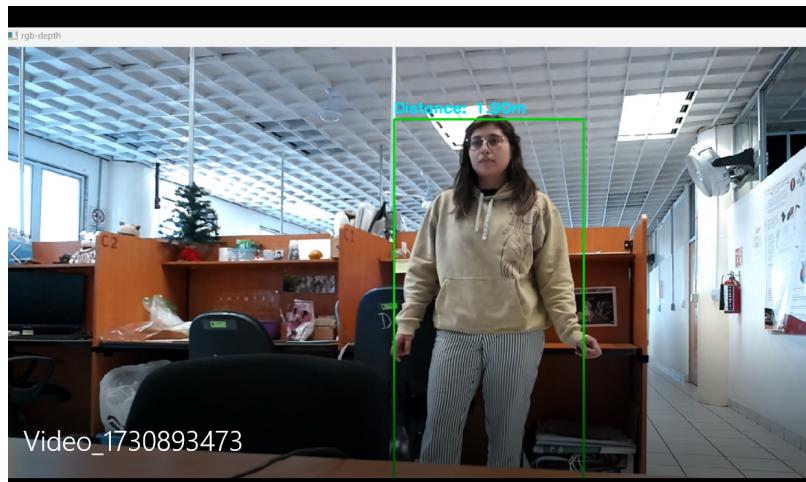


Figura 5.6: Yolo Detección de persona.

4. Cálculo de profundidad en el pixel central del objeto.

El programa utiliza la cámara OAK en combinación con el modelo Tiny YOLO para detectar personas y calcular la distancia a la que se encuentran. Inicialmente, se carga el modelo Tiny YOLO, diseñado para reconocer objetos como personas, animales y vehículos. Las cámaras izquierda, derecha y RGB de la OAK trabajan juntas para capturar imágenes y calcular la profundidad en 3D. El sistema calibra las cámaras para corregir distorsiones y garantizar la precisión de las imágenes. La cámara RGB captura imágenes en tiempo real, mientras que las cámaras estéreo calculan la profundidad, midiendo qué tan lejos están los objetos de la cámara. A continuación, las imágenes se procesan con el modelo Tiny YOLO, que identifica personas basándose en patrones predefinidos. Para cada detección, se calcula la posición (encerrando a la persona en un rectángulo) y se asigna

un nivel de confianza que indica la certeza de la detección. Usando los datos de profundidad, el programa mide la distancia entre la cámara y cada persona detectada. Los resultados se visualizan en pantalla, mostrando un cuadro verde alrededor de cada persona con la distancia calculada en metros. Todo esto ocurre en tiempo real, actualizándose continuamente mientras el programa está en ejecución. Finalmente, el programa permite al usuario detener la ejecución presionando la tecla q, cerrando las ventanas y liberando recursos. A continuación, muestro un diagrama del funcionamiento del Programa:

Presento algunos resultados que se obtuvieron al hacer pruebas en el exterior.

5.1. Modelo de Seguimiento de Mirada y Estimación de Profundidad

Trabaje con un enfoque que combina el seguimiento de la mirada (*gaze tracking*) y ademas con la estimación de profundidad a partir de imágenes monoculares, con el fin de obtener la profundidad de un objeto en una imagen y usarlo para estimar la distancia del objeto en conjunto con el vector de la mirada. Para ello, se integraron dos modelos de aprendizaje profundo: *GazeL2CS* [1] para la estimación de la dirección de la mirada y *Depth Pro* [2] para la predicción de mapas de profundidad a partir de imágenes 2D.

5.1.1. Seguimiento de la Mirada con GazeL2CS

El modelo *GazeL2CS* se basa en la arquitectura de redes neuronales convolucionales y un enfoque de clasificación y regresión simultánea denominado *Look Two-ways Classification and Regression* (L2CS). Este modelo permite estimar la dirección de la mirada con alta precisión mediante la combinación de múltiples técnicas de clasificación de ángulos discretizados y regresión continua. La entrada del modelo consiste en imágenes faciales preprocesadas, mientras que la salida se compone de los ángulos de orientación de la mirada en coordenadas esféricas (*yaw* y *pitch*).

En este trabajo, el modelo se entrenó utilizando un conjunto de datos previamente etiquetado, asegurando que las predicciones de la dirección de la mirada sean robustas frente a variaciones en la iluminación, posiciones de la cabeza y otras condiciones del entorno.

5.1.2. Estimación de Profundidad con Depth Pro

El modelo *Depth Pro* permite obtener mapas de profundidad precisos. lo que lo hace adecuado para aplicaciones en tiempo real. La arquitectura del modelo emplea una combinación de redes neuronales convolucionales y mecanismos de atención espacial para preservar los detalles estructurales en la predicción de la profundidad.

A diferencia de otros métodos de estimación de profundidad que requieren información estéreo o sensores adicionales, *Depth Pro* infiere la distancia relativa de los objetos en la escena a partir de una única imagen RGB. En este estudio, la salida del modelo se utilizó para complementar la estimación de la mirada, proporcionando un contexto tridimensional que mejora la comprensión de la escena.

5.1.3. Integración de GazeL2CS y Depth Pro

La combinación de ambos modelos permite una representación más rica del entorno, donde la dirección de la mirada no solo se estima en el espacio 2D de la imagen, sino que también se contextualiza dentro de la estructura tridimensional del espacio a partir de los mapas de profundidad. Esto facilita la identificación de objetos de interés y mejora la interpretación de la intención visual del

usuario en aplicaciones como interfaces basadas en la mirada y navegación en entornos de realidad aumentada.

Para la integración, se sincronizaron las salidas de ambos modelos de la siguiente manera:

- Se normalizaron las coordenadas de la mirada en el espacio 2D de la imagen.
- Se mapearon las coordenadas estimadas de la mirada sobre el mapa de profundidad generado por *Depth Pro*.
- Se aplicó un modelo de interpolación para determinar la profundidad aproximada del punto de fijación de la mirada dentro de la escena.

Esta metodología permite una estimación más precisa del punto de atención del usuario en entornos tridimensionales, abriendo nuevas posibilidades en aplicaciones interactivas y análisis del comportamiento visual.

5.2. Visualización de Resultados en Pantalla

El sistema presenta una interfaz visual que permite observar en tiempo real la información procesada por el modelo de detección de mirada. En cada fotograma capturado por la cámara, se muestra:

- El video en vivo de la webcam.
- **Rectángulos verdes** que enmarcan los rostros detectados (bounding boxes).
- **Textos informativos** con los valores de pitch (inclinación vertical) y yaw (inclinación horizontal) de la mirada.
- Un **punto verde** y un **punto rojo** que representan la posición estimada donde el usuario está mirando en la pantalla.
- Una **flecha roja** que indica gráficamente la dirección de la mirada estimada, calculada a partir de los ángulos de pitch y yaw.
- Mensajes como:
 - Si el usuario está mirando a la pantalla.
 - El ángulo estimado del dispositivo.
 - La tolerancia configurada para la detección de atención.
 - La profundidad estimada (en caso de usar sensores adicionales).
 - El punto de mirada estimado.
 - La tasa de cuadros por segundo (FPS).
- Un parpadeo de alerta en rojo si el usuario no está prestando atención a la pantalla.

5.2.1. Dibujo de la Flecha de Mirada

La flecha que indica la dirección de la mirada se genera a través de la función `draw_gaze()`, la cual se encarga de representar gráficamente el ángulo de orientación de la mirada.

El procedimiento para calcular y dibujar la flecha es el siguiente:

1. Se obtiene la posición del rostro a partir de la *bounding box*, calculando el centro de esta como punto inicial de la flecha.
2. Se recuperan los valores de **pitch** y **yaw**, que corresponden a los ángulos vertical y horizontal de la mirada, respectivamente.
3. Se ajustan los signos de estos ángulos para que coincidan con el sistema de coordenadas de la imagen:
 - Pitch se invierte, ya que mirar hacia arriba corresponde a valores negativos.
 - Yaw se invierte, ya que mirar hacia la derecha corresponde a valores positivos.
4. Se calcula el ángulo de orientación en grados utilizando:

$$\theta = \text{atan2}(-\sin(\text{yaw}), \sin(\text{pitch}) \cdot \cos(\text{yaw}))$$

5. Se ajusta el ángulo para que 0° represente mirar al frente:

$$\theta = (\theta + 90) \pmod{360}$$

6. Se convierte el ángulo a radianes y se calculan las componentes horizontales y verticales de la flecha:

$$dx = l \cdot \cos(\theta), \quad dy = l \cdot \sin(\theta)$$

donde l es la longitud deseada de la flecha.

7. Finalmente, se dibuja una flecha desde el centro del rostro hasta el punto $(x + dx, y + dy)$ utilizando la función `cv2.arrowedLine()` de OpenCV.

Este mecanismo permite visualizar de manera clara y precisa la dirección en la que el usuario está enfocando su atención visual, siendo útil para sistemas de atención, control o interacción por mirada.



Figura 5.7: Diagrama de funcionamiento del programa para obtención de distancia de personas.

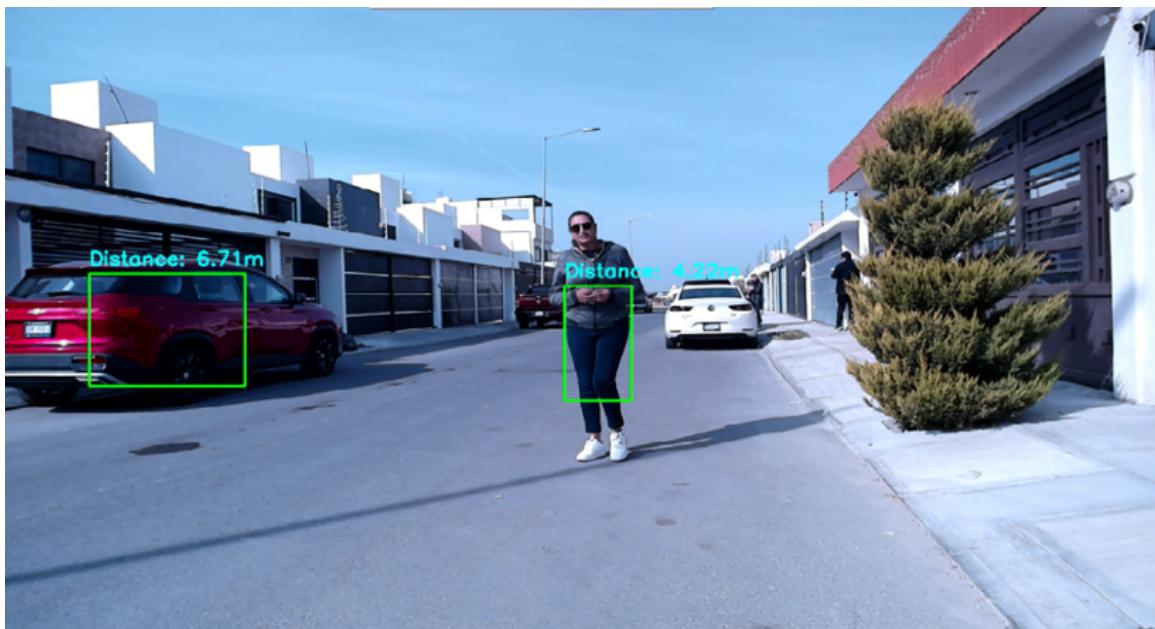


Figura 5.8: Detección de persona y automóvil.



Figura 5.9: Detección de bicicleta y automóvil.



Figura 5.10: Detección de Automóviles.

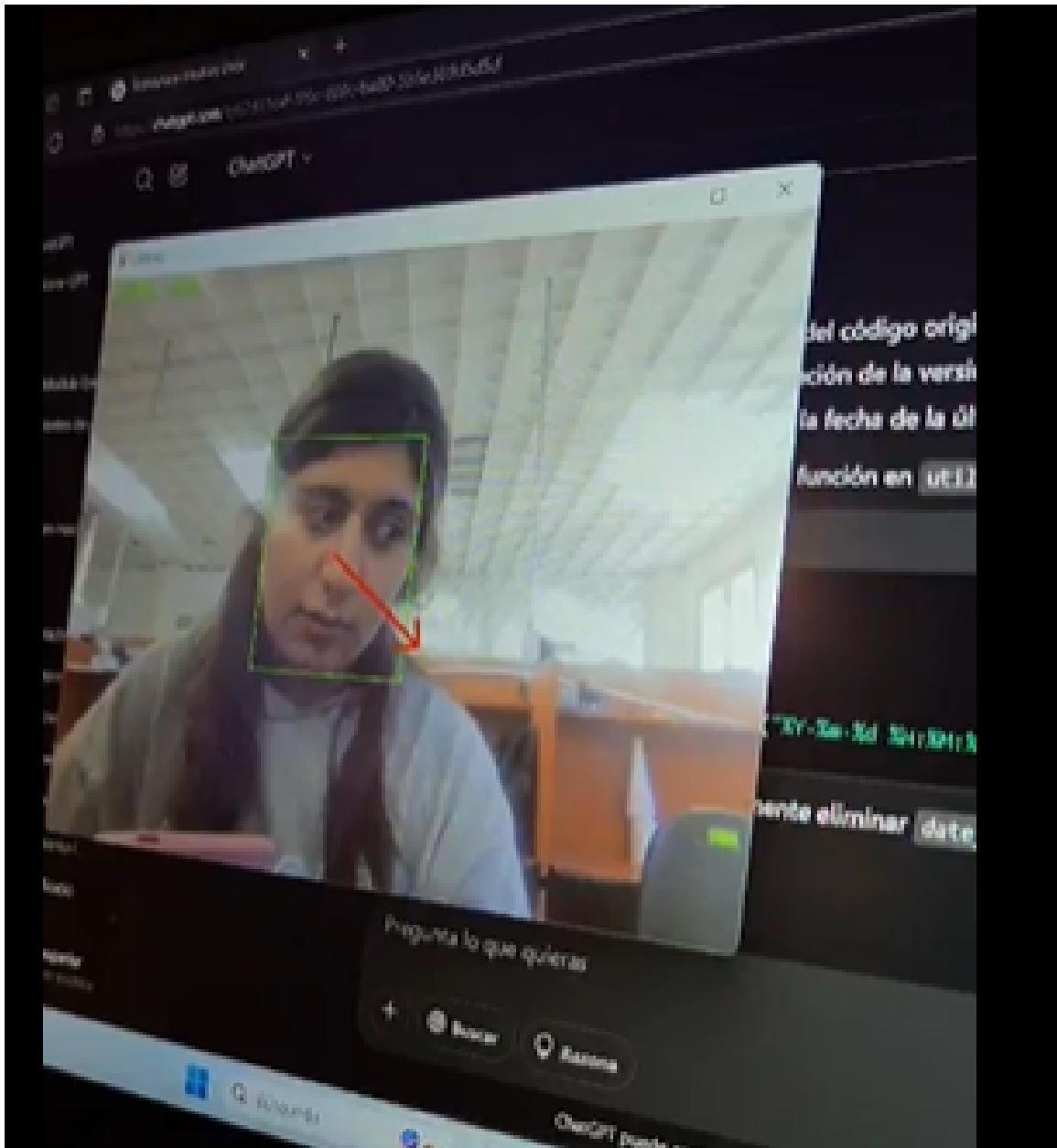


Figura 5.11: Detección de Automóviles.

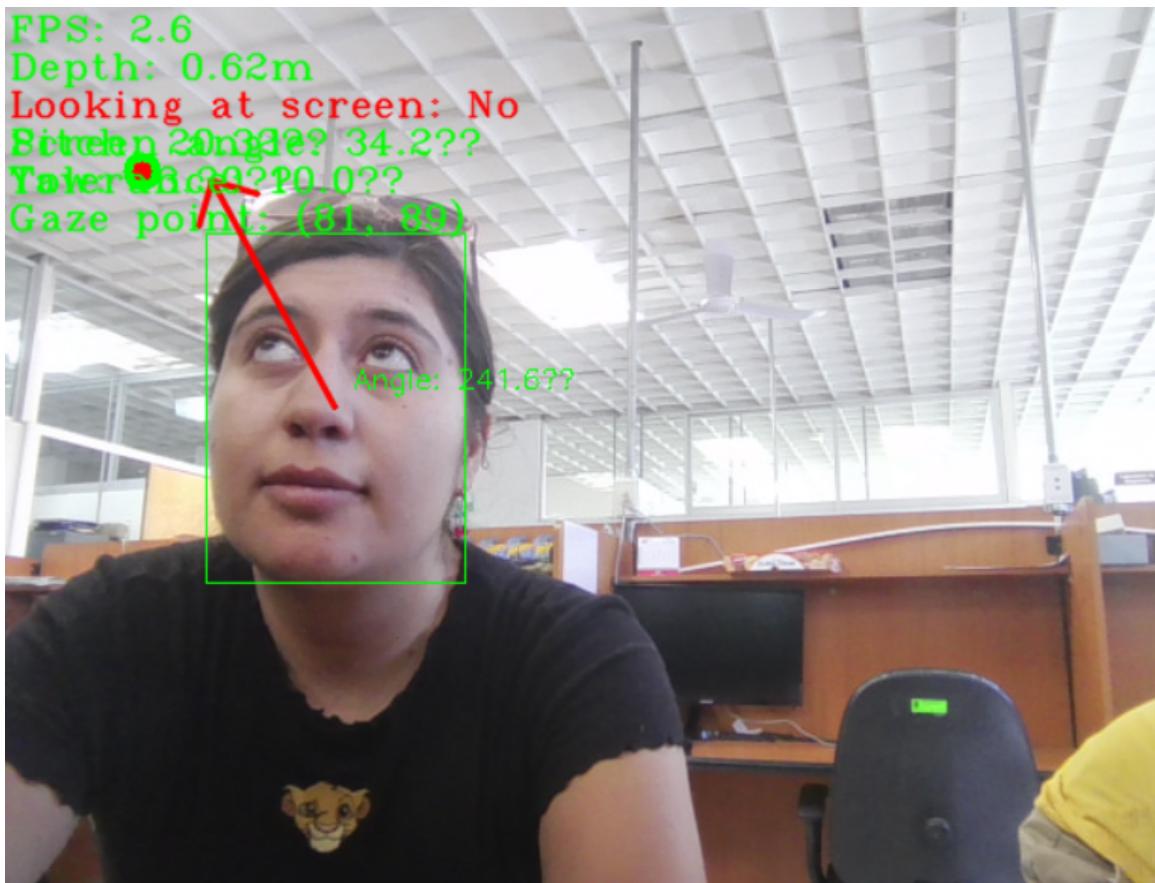


Figura 5.12: prueba gaze.

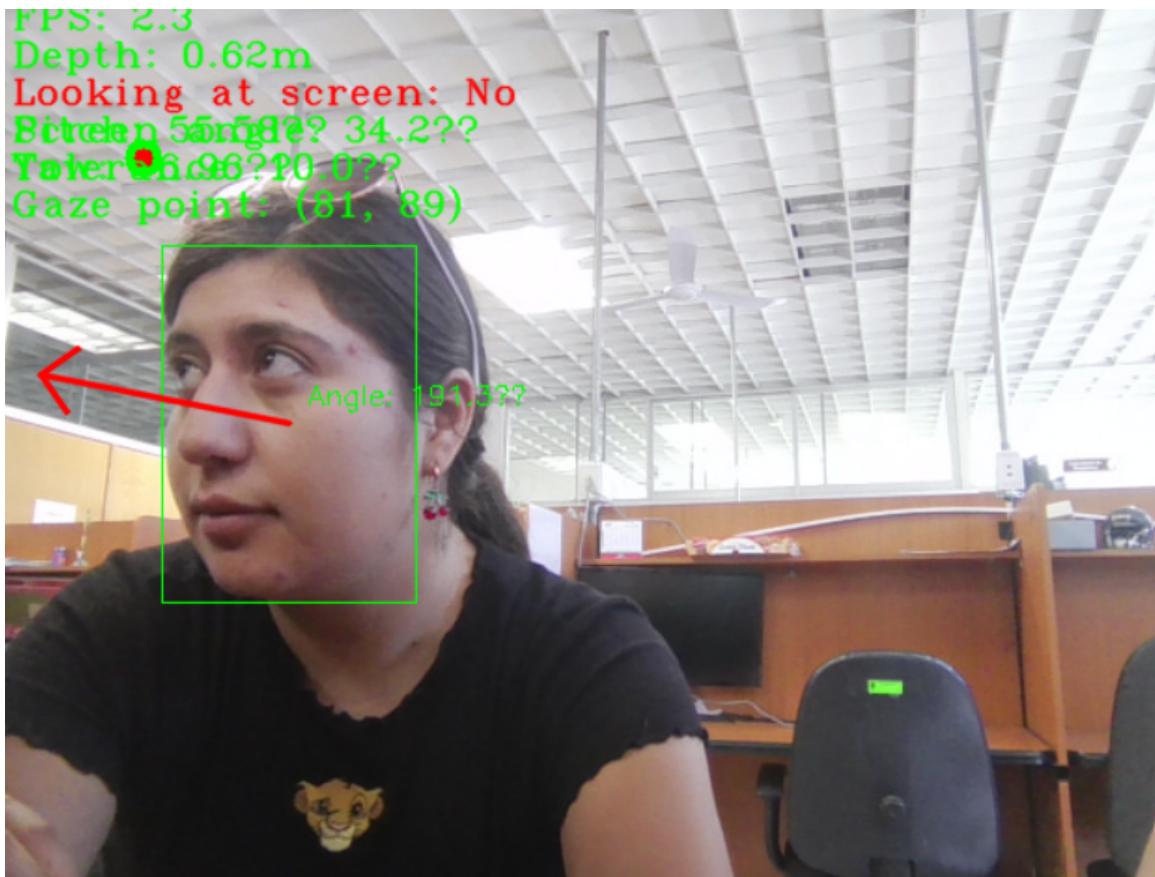


Figura 5.13: pruba gaze.

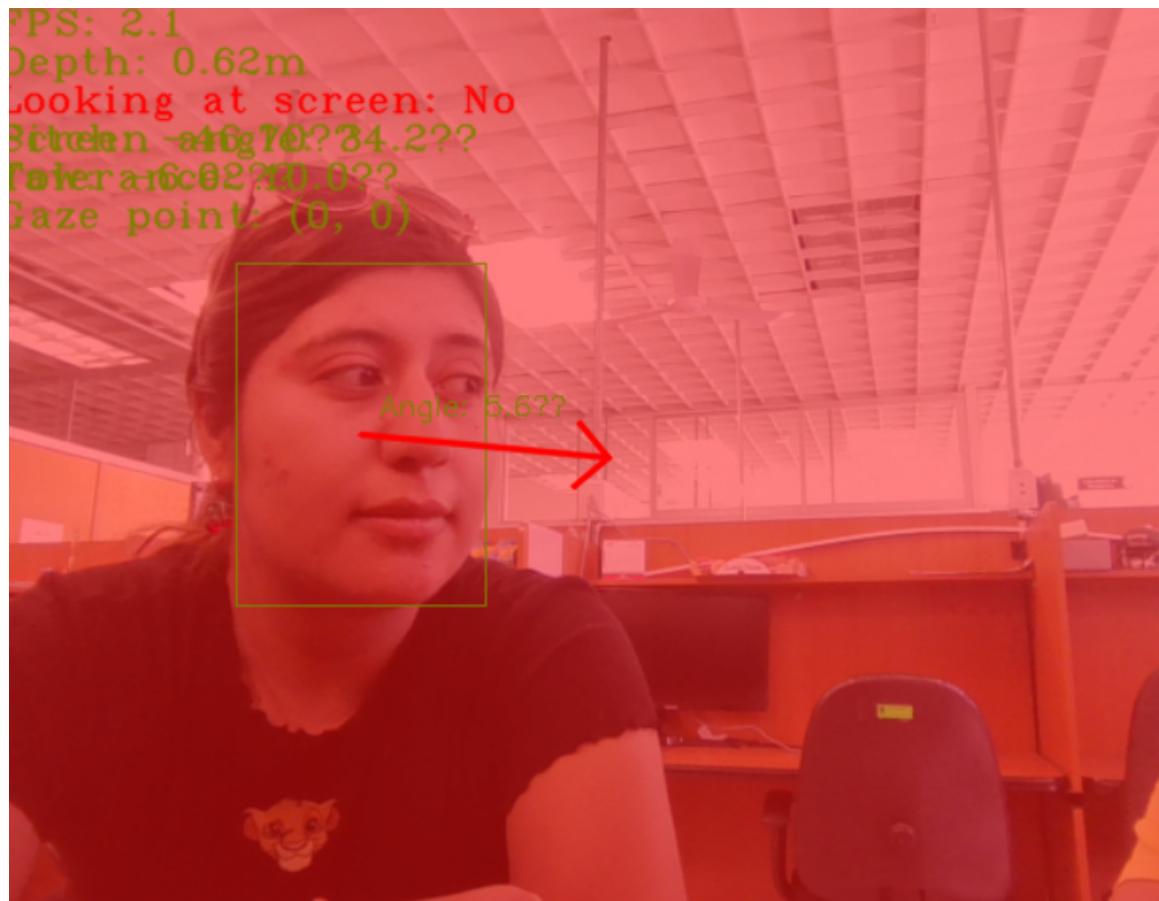


Figura 5.14: pruba gaze.



Figura 5.15: pruba gaze.

Capítulo 6

Conclusiones

El proyecto fue exitoso en la implementación de un sistema capaz de calcular la distancia entre la cámara y las personas detectadas. Este logro demostró la eficacia de combinar las capacidades de las cámaras estéreo de OAK con el modelo Tiny YOLO para la detección de objetos y el cálculo de profundidad en tiempo real. Sin embargo, se identificaron ciertas limitaciones en el modelo Tiny YOLO, ya que en algunas ocasiones no detectaba correctamente personas o vehículos presentes en la escena, lo que resultaba en la falta de datos en esos casos. Este problema destaca la necesidad de explorar alternativas más potentes para la detección de objetos. Como avance futuro, se propone investigar e implementar redes neuronales más robustas y especializadas en la detección de personas y vehículos, con el objetivo de mejorar la precisión y la cobertura del sistema. Esto permitirá superar las limitaciones observadas y potenciar el rendimiento del proyecto en aplicaciones del mundo real.

Bibliografía

- [1] L. C. Reveles-Gomez, H. Luna-Garcia, and J. M. Celaya-Padilla, “Development of an automotive safety system for pedestrian detection by fusing information from reversing camera and proximity sensors using convolutional neural networks,” in Proceedings of the XI Latin American Conference on Human Computer Interaction, CLIHC ’23, (New York, NY, USA), Association for Computing Machinery, 2024.
- [2] A. Singh and N. Patwari, “Online learning for dynamic impending collision prediction using fmcw radar,” ACM Transactions on Internet of Things, vol. 5, 08 2023.
- [3] F. Li, X. Li, Q. Liu, and Z. Li, “Occlusion handling and multi-scale pedestrian detection based on deep learning: A review,” IEEE Access, vol. 10, pp. 19937–19957, 2022.
- [4] J. Smith and J. Doe, “Lidar for autonomous driving: The future of road safety,” Journal of Autonomous Vehicles, vol. 34, pp. 123–134, 2021.

Bibliografía

- [1] Autor(es). *GazeL2CS: Precise Gaze Tracking*. Disponible en: <file:///C:/Users/beatr/OneDrive/Escritorio/maestria/proyecto/2do%20semestre/GazeL2CS/2203.03339v1.pdf>.
- [2] Autor(es). *Depth Pro: Sharp Monocular Metric Depth in Less Than a Second*. Publicado en octubre de 2024.