

DISEÑO DE DOMO ESTÁTICO

Dr. Ing. Orlando Micolini, Ing. Luis O. Ventre, Ing. Gabriel Valenzuela, Julian Tantera, y Emiliano Zanotti

Laboratorio de Arquitectura de Computadoras, FCEFYN-Universidad Nacional de Córdoba Av. Velez Sarfield 1601, CP-5000, Córdoba, Argentina {luis.ventre, orlando.micolini, gabriel.valenzuela}@unc.edu.ar {julian.tantera, emiliano.zanotti}@mi.unc.edu.ar

Resumen En el presente trabajo se detalla el proceso de diseño de un Domo Estático, desarrollado en el Laboratorio de Arquitectura de Computadoras de la FCEFYN de la UNC. Este proyecto incorpora funcionalidades de vigilancia del entorno en tiempo real, como el reconocimiento de humanoides, medición de distancias y envío de alertas al usuario, a través de programas de software y hardware especializado. Utiliza la plataforma de desarrollo Jetson TX2, que dadas sus capacidades computacionales soporta los requerimientos del sistema, a la que se incorpora un arreglo de cámaras que permite una amplia cobertura del entorno. El resultado de este proyecto muestra la implementación de algoritmos sofisticados y hardware específico para el monitoreo sin utilizar sistemas mecánicos. Así mismo se han validado los criterios de diseño y se evaluaron parámetros de rendimiento, analizados cuantitativamente frente a una integración de componentes específica, obteniéndose un sistema estático y robusto con capacidad de vigilancia en tiempo real y exento de interrupciones.

Keywords: Detección de objetos · Sistemas de visión robótica · Video vigilancia · Sistemas Embebidos.

1. Introducción

A lo largo de los últimos años, el reconocimiento de imágenes y la detección de objetos ha sido el centro de atención en múltiples ámbitos, desde el sector doméstico, hasta el industrial y profesional. Esto permitió un desarrollo incremental con capacidades cada vez mayores gracias a las distintas investigaciones e inversiones de la comunidad.

En la actualidad existe una amplia variedad de domos utilizados para video-vigilancia, la mayoría de los cuales están equipados con una cámara motorizada única para abarcar un mayor ángulo de visión. A pesar de la abundancia de estos domos en el mercado y su sencilla operación, presentan ciertas desventajas. Por ejemplo, el mecanismo rotatorio de la cámara necesario para la cobertura de 360 grados presenta un punto débil. Además, la gran oferta de cámaras en el mercado representa una importante alternativa, que es aún más significativa si existe la necesidad de adquirir múltiples domos. Las deficiencias mencionadas, permiten visibilizar posibles mejoras e incorporación de nuevas funcionalidades útiles para la vigilancia.

En este trabajo se detalla el desarrollo de un domo que permite realizar diversas tareas de vigilancia a través de un arreglo de cámaras estático. Este proceso es llevado a cabo haciendo uso de la plataforma de desarrollo Jetson TX2, para evaluar su alcance y sus capacidades. El resultado de este proyecto es construir un sistema capaz de reconocer humanoides y medir la distancia a ellos a partir de algoritmos de inteligencia artificial y visión estereoscópica, posibilitando la toma de decisiones en tiempo real. De esta manera es posible obtener información del entorno, que permite realizar una vigilancia continua sin la intervención de un humano.

El contenido de este artículo comienza detallando la sección de materiales y métodos. A continuación, se exponen los Casos de Estudio, seguidos por la enumeración de los resultados obtenidos y su correspondiente explicación. Para finalizar, se presentan las conclusiones y los trabajos futuros.

2. Antecedentes

La propuesta se beneficia de la experiencia adquirida a través del proyecto Hermes III, especialmente en lo que concierne a su utilización de la plataforma de desarrollo Jetson TK1. Además, se ha incorporado conocimiento proveniente de algoritmos de inteligencia artificial de código abierto desarrollados por expertos en la materia. Este enfoque combinado de experiencia práctica y algoritmos especializados ha enriquecido significativamente la propuesta.

Además, para el desarrollo de este proyecto, se han adherido a los siguientes principios rectores: una óptima relación entre rendimiento y costos, capacidad de escalabilidad, facilidad de mantenimiento, posibilidades de uso tanto en el ámbito educativo como en el ingenieril, simplicidad de uso y acceso a información de carácter público. Se le ha dado prioridad a los criterios de funcionalidad, costo y simplicidad al diseñar y seleccionar los componentes.

3. Objetivos

El principal objetivo del proyecto es diseñar y construir un domo estático, el cual posea un arreglo de cámaras fijo capaz de abarcar un ángulo de visión de al menos 270 grados en todo momento. Como objetivos secundarios, se puede mencionar la implementación de un sistema de cámaras que desde un mismo punto cubra diferentes áreas, como también de un algoritmo que permita estimar distancias a personas y que envíe alertas ante intrusiones.

Además, esta iniciativa tiene como objetivo ir más allá del aprendizaje puramente basado en software y adentrarse en lo vinculado al propio concepto de los domos. Estas incluyen la actualidad del tema planteado, su capacidad para simplificar la supervisión del entorno y la posibilidad de desarrollar funcionalidades que potencien su rendimiento. Por otro lado, también se identifican motivaciones de índole académica que no pueden ser pasadas por alto.

4. Materiales y Métodos

La arquitectura general del sistema (Figura 1) estará compuesta por un entorno de desarrollo que será definido a continuación. Además, se adicionarán como periféricos un monitor para observar la salida del sistema con sus resultados, y las 4 cámaras utilizadas como entrada de vídeo.

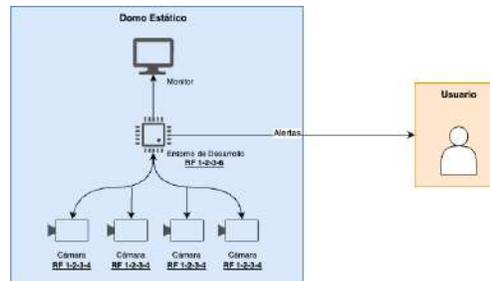


Figura 1. Diagrama de alto nivel del sistema.

Entorno de desarrollo Nvidia Jetson TX2 Luego de diferentes análisis sobre las ventajas y desventajas que presenta la Nvidia Jetson TK1 utilizada en el proyecto Hermes 3, se escogió como entorno de desarrollo la placa Nvidia Jetson TX2 (Figura 2), brindada por el Laboratorio de Arquitectura de Computadoras de la Facultad. Además, se optó por la instalación del Sistema Operativo Ubuntu 18.04. Uno de los principales beneficios que presenta la Jetson TX2 es su GPU integrada, lo cual permite procesar rápidamente los datos de las imágenes capturadas y tomar decisiones en tiempo real en función de lo detectado.

Además, se pueden destacar los siguientes beneficios:

1. Perfiles de energía dinámicos (Max Q y Max P).
La Jetson TX2 presenta dos configuraciones predefinidas: Max-Q, que opera a 7,5 W para ofrecer eficiencia máxima y Max-P, que alcanza un rendimiento máximo a menos de 15 W. Además, ofrece flexibilidad para crear configuraciones personalizadas que equilibren eficiencia y rendimiento según las necesidades de la aplicación.
2. Optimización de trabajo de CPU.
El complejo de CPU del TX2 incluye un superescalar NVIDIA Denver 2 de 7 vías y doble núcleo para un alto rendimiento de un solo subproceso para la optimización dinámica del código, y un Arm Cortex-A57 de cuatro núcleos para el procesamiento de subprocesos múltiples.
3. Mejoras de rendimiento de GPU y sensores.
La Jetson TX2 aprovecha dos multiprocesadores de transmisión Pascal, con 128 núcleos CUDA cada uno, optimizados para un rendimiento eficiente. Su arquitectura GPU Pascal ofrece mejoras notables en rendimiento y eficiencia.



Figura 2. Nvidia Jetson TX2.

La configuración habitual del pipeline involucra interfaces de alta velocidad como CSI, PCIe, USB3 y Gigabit Ethernet, con etapas de pre y postprocesamiento de CUDA que incluyen la conversión de gama y análisis estadístico de las salidas de la red neuronal.

4. Incremento de mejora y ancho de banda

Con el doble de memoria y ancho de banda que la Jetson TX1, la Jetson TX2 puede capturar y procesar simultáneamente flujos de datos adicionales de gran ancho de banda, incluidas cámaras estéreo y entrada y salida 4K UHD. A través de canalizaciones, el aprendizaje profundo y la visión por computadora fusionan múltiples sensores de diferentes fuentes y dominios espectrales para mejorar la percepción y la conciencia situacional.

Tarjeta PCI Express USB 3.0 de 4 puertos En primera instancia se optó por utilizar una tarjeta PCI Express la cual hace uso de un bus de datos compartido entre los 4 puertos USB. Al realizar pruebas se halló que esto es una limitante para operar con 4 cámaras a la vez, por lo que se debió utilizar una tarjeta desarrollada por StarTech (Figura 3) que nos permita tener un bus de datos exclusivo para cada puerto USB, asegurando así el correcto funcionamiento de las cámaras

El beneficio que presenta la tarjeta es que cada puerto al que se conectan las cámaras tiene su propio conjunto de recursos, lo que evita la congestión de datos y los conflictos que se presentaban al utilizar un único controlador en el hub USB. De esta manera se logra una distribución equitativa de la carga de datos, permitiendo un flujo eficiente y simultáneo de información desde cada cámara conectada, lo que resulta en una captura de imágenes fluida y en tiempo real.

Reconocimiento de humanoides y medición de distancias Se llevó a cabo una investigación sobre trabajos relacionados al reconocimiento de objetos en imágenes. Se exploraron diferentes proyectos y enfoques utilizados por la comunidad de



Figura 3. Tarjeta PCI Express.

investigación y desarrollo en este campo. A partir de esto, se escogieron dos proyectos que utilizan diferentes enfoques: Real Time Object Detection (Alexander Robles) y Object Detection Webcam (Gabriel Cassimiro). El primero de estos, lleva a cabo el reconocimiento de objetos a partir del algoritmo YOLO (You Only Look Once) y el modelo pre-entrenado “YOLOv3”, un conjunto de datos amplio y diverso para reconocer una amplia variedad de objetos en imágenes. Mientras que el segundo, emplea una biblioteca de TensorFlow que proporciona una amplia variedad de modelos de aprendizaje profundo pre-entrenados. En este caso, el proyecto utiliza el modelo “ssd-mobilenet-v2-coco” que ha sido entrenado en el conjunto de datos COCO (Common Objects in Context), que incluye una amplia gama de objetos comunes.

Al realizar un análisis exhaustivo de las alternativas escogidas, se determinó que el proyecto Object Detection Webcam presenta un modelo listo para usar, con una fácil implementación y configuración, permite ahorrar tiempo y recursos al no requerir el entrenamiento de un modelo desde cero como YOLO, cuya implementación es más compleja. Además brinda una gran flexibilidad, ya que es posible integrar modelos más livianos y veloces, lo que resulta en una mayor eficiencia y velocidad en el proceso de reconocimiento de objetos. Una ventaja que presenta el proyecto Object Detection Webcam para nuestra implementación es que utiliza un archivo de etiquetas, el cual proporciona información específica sobre las clases de objetos que el modelo está diseñado para detectar, permitiendo el filtrado y detección específica de los objetos de interés, siendo en este caso los humanos.

Una vez realizada la detección se desarrolla la funcionalidad de medición de distancias al humanoide, a partir de las imágenes obtenidas por el arreglo de cámaras. Para ello, se llevó a cabo una nueva investigación de proyectos, concluyendo con dos implementaciones de diferentes naturalezas Distance measurement using single camera (por Asadullah Dal), implementación que permite medir distancias utilizando una sola cámara y Stereo Vision Depth Estimation (por Nicolai Nielsen), proyecto que se centra en la estimación de profundidad utilizando técnicas de visión estéreo (múltiples cámaras), tomando como refe-

rencia el rostro de un humano. Esta fue una de las razones clave por las que nos inclinamos por este proyecto.

Para integrar un sistema de visión estereoscópica se ubican las cámaras de forma paralela (O, O'), como se observa en la Figura 4, superponiendo su ángulo de visión a una distancia conocida entre ellas (Baseline B). Este algoritmo funciona calculando la diferencia en la posición de un punto en la imagen (disparity), permitiendo de esta manera calcular la profundidad de los objetos en la escena.

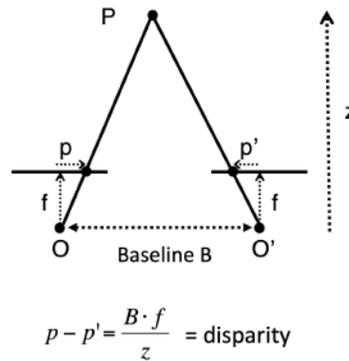


Figura 4. Ubicación de cámaras y cálculo de disparidad.

La ubicación horizontal alineada de las cámaras y su disposición separada se ha seleccionado debido a su similitud con la disposición de los ojos humanos, que están separados por una distancia que proporciona dos puntos de vista ligeramente diferentes.

Luego se realizó la integración de ambos proyectos de detección de humanos y medición de distancias. Al combinar la estimación de la profundidad estéreo y el reconocimiento de objetos, el sistema puede proporcionar información más detallada sobre la posición y la distancia de las personas en la escena. Esto es especialmente importante para aplicaciones de seguridad y seguimiento, donde se requiere una comprensión completa del entorno y la capacidad de identificar y rastrear humanos en tiempo real.

Se observó que los resultados obtenidos de esta implementación en un principio no seguían un comportamiento lineal (Figuras 5 y 6), por lo que se utilizó la función polyval de la biblioteca NumPy para generar un polinomio que ajuste las mediciones estimadas de distancia obtenidas por el sistema a las mediciones reales.

Cada distancia estimada por el sistema será ajustada por el polinomio, obteniendo mediciones más confiables y precisas en diferentes situaciones y condiciones. En la Figura 7 se puede observar el comportamiento de la obtención de



Figura 5. Medicion a una distancia aproximada de 1 metro.



Figura 6. Medicion a una distancia aproximada de 2 metros.

distancias, teniendo en cuenta las medidas reales y las estimadas por el sistema antes y después de aplicar el polinomio generado (azul y rojo, respectivamente).

Durante la ejecución del proyecto en la plataforma Jetson TX2, se identificaron deficiencias significativas en el procesamiento y ejecución de los códigos implementados. Se experimentó que el tiempo de arranque del proyecto era excesivamente largo y se observó un alto consumo de la unidad central de procesamiento (CPU) durante la ejecución.

El problema se manifestó en un uso continuo y constante del CPU cercano al 95 % de su capacidad máxima. Esto indica que el procesador se encontraba operando en su límite de carga, lo cual tiene un impacto negativo en el rendimiento general del sistema.

Para abordar esta deficiencia, se realizó un análisis exhaustivo del código implementado buscando oportunidades de optimización y mejoras en la eficiencia del procesamiento, considerando la implementación de algoritmos más eficientes desde el punto de vista computacional. La existencia de recursos limitados, presenta la necesidad de abordar un nuevo desarrollo centrado en la creación de un sistema de medición de distancias optimizado para la arquitectura Jetson TX2.

Esta propuesta se basa en los conocimientos adquiridos del proyecto denominado Jetson-Inference, una biblioteca de software de código abierto diseñada específicamente para la plataforma Jetson TX2. Este proyecto desarrollado por Dustin Franklin (NVIDIA Jetson Developer), ofrece una amplia gama de capacidades de inferencia de IA y visión por computadora para la detección y medición de distancias a humanos

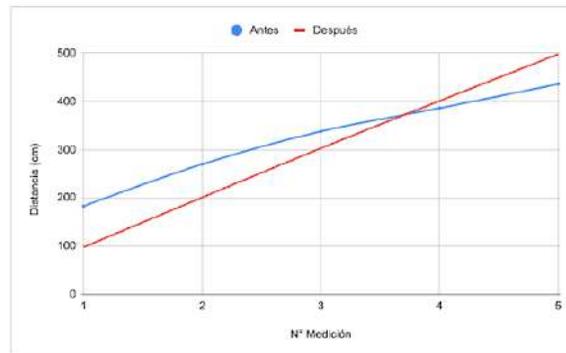


Figura 7. Comparación de medidas originales y ajustadas.

Jetson-Inference utiliza la biblioteca de optimización de modelos TensorRT para ejecutar modelos optimizados en las GPUs de la plataforma Jetson TX2, además de PyTorch como framework para su entrenamiento. La presencia de TensorRT en el proyecto es fundamental, ya que permite acelerar y optimizar la ejecución de los modelos de IA en las GPUs de la Jetson TX2, aprovechando al máximo la potencia de procesamiento paralelo de estas unidades de procesamiento gráfico. A continuación, en la Figura 8, se observa de manera gráfica el proceso de calificación de objetos con TensorRT.

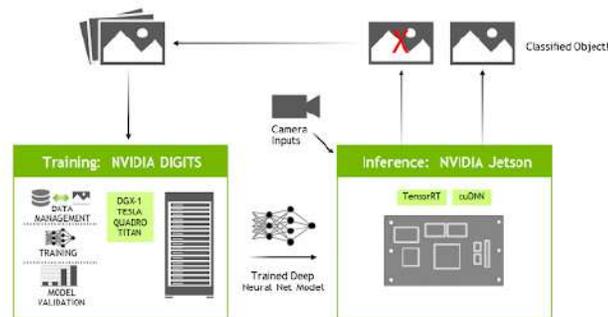


Figura 8. Clasificación de objetos con TensorRT.

Cabe destacar, que la biblioteca Jetson-Inference se basa en la arquitectura de red neuronal convolucional (CNN) y utiliza algoritmos de aprendizaje profundo para realizar tareas de detección y reconocimiento visual. Utiliza modelos pre-entrenados, como SSD-MobileNet y TAO PeopleNet, que han sido entrenados en grandes conjuntos de datos para detectar objetos y personas en imágenes y videos.

Para la detección y medición de distancias a humanos, se utilizaron técnicas de visión estereoscópica y triangulación lo que permite la medición precisa y en tiempo real de las distancias a humanoides en el entorno de la Jetson TX2 que integra el domo estático.

Además, se creó una nueva implementación de visión estereoscópica basada en el proyecto “stereoDepth” desarrollado por Ali Yasin Eser (ios developer), que permite calcular la profundidad de una escena utilizando imágenes estéreo. Este emplea técnicas de correspondencia estéreo para encontrar los puntos en común en las imágenes izquierda y derecha capturadas por las cámaras, y calcula la disparidad entre estos puntos, para luego estimar la profundidad de cada punto en la escena. Además, presenta una destacada funcionalidad que permite calibrar las cámaras de manera individual y luego realizar una calibración estéreo.

Sistema de generación de alertas Un sistema de alarma es una parte esencial en un sistema de vigilancia ya que proporciona una detección temprana, una respuesta rápida y una disuasión efectiva ante intrusiones. Para suplir esto, existen numerosos métodos de alerta y comunicación con el usuario.

Luego de analizar las diferentes alternativas, se optó por desarrollar un sistema de envío de alertas a la aplicación Telegram. Esta plataforma de mensajería instantánea ofrece una amplia gama de características y funcionalidades que lo hacen sobresalir en términos de versatilidad y eficiencia en la entrega de alertas. Además, ofrece numerosas características y beneficios para desarrollar bots de comunicación, como Telebot, herramienta indispensable para el sistema de alertas.

Para el desarrollo de este sistema, se utiliza la api denominada pyTelegram-BotAPI, es una biblioteca de Python que simplifica la interacción con la API de Telegram y facilita el desarrollo de bots de manera rápida y sencilla.

pyTelegramBotAPI se basa en el uso de solicitudes HTTP para realizar llamadas a la API de Telegram y recibir respuestas en formato JSON. Esto significa que se establece una conexión entre el bot y los servidores de Telegram, y se envían solicitudes para realizar diferentes acciones.

El bot es capaz de enviar alertas cuando se detecta un posible intruso a una distancia menor al límite configurable. También cuenta con la capacidad de interactuar con el usuario a través de comandos para ver las últimas distancias obtenidas, junto con las imágenes correspondientes.

Construcción de gabinete Para albergar todos los componentes de forma ordenada y fijar las cámaras de forma segura, se desarrolló un gabinete el cual fue construido utilizando filamento termoplástico y una impresora 3D.

El gabinete (9) cuenta con una abertura que permite situar la placa y permite un fácil acceso a los componentes. Se realizaron agujeros en la parte superior situados a 10 y 20 cm de separación entre sí, lo que permite acercar las cámaras si se desea para obtener una mejor coincidencia de puntos clave. También cuenta con un techo el cual protege a las cámaras de cualquier objeto que pudiera dañarlas, y un orificio central que permite organizar los cables de las cámaras y una correcta ventilación del sistema.



Figura 9. Diseño integrado del gabinete.

A continuación, se muestra una imagen del prototipo ya construido y armado con sus componentes (Figura 10).



Figura 10. Gabinete con componentes integrados.

5. Casos de Prueba

Dentro de los experimentos más importantes, se realizaron los siguientes casos de prueba.

Prueba 1: Medir distancia a una persona situada a una distancia de 2 metros.

Se sitúa la persona a una distancia previamente medida de 2 metros con respecto a las cámaras (Figura 11), y una vez estabilizada la medición se toman las últimas 3 distancias medidas, siendo:

- Medida 1: 1.92 metros
- Medida 2: 1.97 metros



Figura 11. Medición a una distancia de 2 metros.

- Medida 3: 2 metros

Vemos entonces que las medidas estimadas tienen una variación de ± 0.08 metros.

Prueba 2: Emisión de alerta cuando se detecta un intruso a una distancia menor al límite.

Observamos en la Figura 12 que se recibe correctamente la alerta cuando el domo detecta un intruso a una distancia menor a la límite.

6. Resultados

El domo fue capaz reconocer humanoides a través de algoritmos de aprendizaje profundo e inteligencia artificial, utilizando una biblioteca específica basada en una arquitectura de red neuronal convolucional (CNN). En cuanto a la medición de distancias, se han obtenido resultados satisfactorios, siempre que se cumpla con los rangos máximos y mínimos alcanzables por el sistema. Esto fue logrado gracias a la implementación de un sistema de visión estereoscópica que permite calcular la profundidad de una escena utilizando imágenes estéreo.

Se ha logrado una correcta integración y funcionamiento del hardware adicional a la plataforma, refiriéndose a las cámaras USB y la tarjeta PCI Express de 4 puertos.

Se alcanzó un funcionamiento eficiente de la plataforma de desarrollo Jetson TX2 a través de herramientas de reconocimiento específicas para arquitecturas Nvidia Jetson como Jetson-Inference y del modelo pre-entrenado para la detección SSD-MobileNet-V2, cuyo desempeño fue sustancialmente más rápido que otras alternativas. En cuanto a las pruebas sobre el desempeño del sistema en funcionamiento durante la detección de humanoides y medición de distancias con la totalidad de las cámaras, se observó que la TX2 se encuentra al límite de sus capacidades (Figura 13). Sin embargo, fue suficiente para ejecutar todos los procesos en simultáneo.



Figura 12. Comunicación con bot durante detección.

```

[Top MAXN] CPU 17.7% GPU 98.5%
Model: quill - [L4T 32.7.4] Jetpack NOT DETECTED
 1 [|||||] 55.0% 2.0GHz 4 [|||||] 77.5% 2.0GHz
 2 [|||||] 0.0% 345MHz 5 [|||||] 77.8% 2.0GHz
 3 [|||||] 0.0% 345MHz 6 [|||||] 80.8% 2.0GHz
Mem [|||||] [5.1G/7.7G] PMM [|||||] 69.8% 3664RPM
Dsp [|||||] 8k/3.8k Jetson Clocks: Inactive
Enc [|||||] 1.9GHz 0x NV Power[0]: MAXN
Optime: 0 days 1:25:31

GPU [|||||] 98.5% 1.3GHz
Dsk [|||||] 22.6G/27.4G

PID USER GPU TYPE PRI S CPU% MEM [GPU MEM] Command
17332 nvidia I G 20 R 148.2 396M 352M python3

[HW engines] [Sensor] [Temp] [Power] [Inst] [Avg]
APE: [OFF] BCPU 39.50C 1V8 IO 0mW 0mW
NVENC: [OFF] NVDEC: [OFF] GPU 38.00C 3V3 IO SLP 4.1W 4.1W
NVJPG: [OFF] SE: [OFF] MCPU 39.50C 3V3 SYS 4.2W 4.2W
PLL 39.50C 2V3 SYS M2 0mW 0mW
Tboard 33.00C 4V0 WIFI 57mW 45mW
Tdiode 37.25C 5V0 TO SYS 4.1W 4.1W
thermal 38.70C MUX 9.4W 9.4W
SYS CPU 1.9W 1.9W
SYS DDR 2.6W 2.6W
SYS GPU 2.1W 2.1W
    
```

Figura 13. Utilización de recursos de la placa.

Refiriéndonos a las alertas, se logró implementar un sistema de alarma a través de un bot en la mensajería Telegram, que notifica al usuario cuando se detectan intrusiones en un perímetro determinado en tiempo real, por medio de mensajes de texto, imágenes y logs correspondientes a la detección.

En la medición de distancias, es importante destacar el incremento de precisión en las mediciones obtenidas cuando se sitúan las cámaras a una distancia de 20 cm.

Finalmente, se diseñó y confeccionó un prototipo 3D del gabinete que almacenará los componentes del domo, conformando un armazón robusto para el desempeño en entornos cerrados.

7. Conclusiones

Como resultado de este trabajo, se ha diseñado, implementado y construido un domo estático que incorpora cuatro cámaras USB de bajo costo a la plataforma de desarrollo Nvidia Jetson TX2. Este domo está capacitado para la detección de intrusos, el envío de alertas ante intrusiones y la medición a este de distancias en tiempo real.

Se realizaron diferentes adaptaciones y reconfiguraciones de tecnologías y frameworks de acuerdo a los requerimientos planteados, tras lo cual se llevaron a cabo los experimentos pertinentes con el fin de determinar el cumplimiento de los objetivos del proyecto. Esta estrategia nos permitió obtener el diseño que satisfizo la detección y medición de distancias de múltiples objetivos con diferentes ángulos de visión. Para esto, se desarrolló un algoritmo eficaz, el cual ha sido validado experimentalmente, capaz de detectar intrusos y el envío de notificaciones en tiempo real.

A través de las diferentes pruebas, se descubrieron nuevos desafíos a resolver a futuro. La medición de distancias al objetivo presenta cierta inexactitud si no se realiza la calibración óptima. Esto da lugar a la búsqueda de nuevos algoritmos que permitan mayor precisión.

En resumen, este proyecto propone un sistema efectivo y autónomo, que permite la vigilancia continua del entorno a través de herramientas de inteligencia artificial, brindando la capacidad de reconocer imágenes en tiempo real y obtener información específica, abriendo nuevos caminos para resolver desafíos en diversos campos.

Referencias

1. Building a depth estimation system with stereo vision and opencv c++. <https://www.youtube.com/watch?v=uKDAVcSaNZA&t=1767s>
2. Computer vision using opencv. <https://morioh.com/a/b5dd599c7e4e/computer-vision-using-opencv>
3. Computervision. <https://github.com/niconielsen32/ComputerVision/tree/master/StereoVisionDepthEstimation>
4. Disparity and depth estimation from stereo camera. <https://www.digitalnuage.com/disparity-and-depth-estimation-from-stereo-camera>

5. Disparity and depth estimation from stereo camera. <https://www.digitalnuage.com/disparity-and-depth-estimation-from-stereo-camera>
6. Disparity map in stereo vision. <https://www.baeldung.com/cs/disparity-map-stereo-vision>
7. Distance measurement using single camera. https://github.com/Asadullah-Dal17/Distance_measurement_using_single_camera/tree/main
8. Machine learning vs. deep learning. <https://www.cdainfo.com/es/noticias/148-machine-learning-vs-deep-learning>
9. Nvidia jetson tx2 delivers twice the intelligence to the edge. <https://developer.nvidia.com/blog/jetson-tx2-delivers-twice-intelligence-edge/>
10. Object detection with tensorflow model and opencv. <https://towardsdatascience.com/object-detection-with-tensorflow-model-and-opencv-d839f3e42849>
11. Robot móvil hermes iii (rediseño del sistema de sensores y de control). https://www.researchgate.net/publication/356815496_Robot_Movil_Hermes_III_rediseño_del_sistema_de_sensores_y_de_control
12. Sistema inteligente de relevamiento de stock. https://www.researchgate.net/publication/356442991_Sistema_inteligente_de_relevamiento_de_stock
13. Stereo vision camera. https://github.com/AlexJinlei/Stereo_Vision_Camera
14. Stereo vision camera. http://vision.stanford.edu/teaching/cs231a_autumn1112/lecture/lecture10_multi_view_cs231a.pdf
15. Stereo-webcam-depth-detection. <https://github.com/mkokshoorn/Stereo-Webcam-Depth-Detection>
16. Tensorflow overview. <https://blog.finxter.com/tensorflow-overview/>