



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Sistema Embebido para Asistencia de Conducción
Basado en Lógica Difusa Tipo-2

presentada por

Ing. Ángel Arturo Rendón Castro

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis
Dr. Dante Mújica Vargas

Cuernavaca, Morelos, México. Enero de 2020.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO.

Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor., **09/enero/2020**

OFICIO No. DCC/001/2020
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

C. DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Ing. Ángel Arturo Rendón Castro**, con número de control M18CE013, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado "**Sistema embebido para asistencia de conducción basado en Lógica Difusa Tipo-2**" y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

Dr. Dante Mújica Vargas
Doctor en Comunicaciones y Electrónica
09131756
Director de tesis

Dra. Andrea Magadán Salazar
Doctora en Ciencias Computacionales
10654097
Revisor 1

Dr. Andrés Blanco Ortega
Doctor en Ciencias en Ingeniería Eléctrica
6559298
Revisor 2

C.c.p. Depto. Servicios Escolares.
Expediente / Estudiante
JGGS/Imz

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490 Cuernavaca, Morelos.
Tel. (01) 777 3 62 77 70, ext. 3202, e-mail: dcc@cenidet.edu.mx
www.tecnm.mx | www.cenidet.tecnm.mx

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico





Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección académica

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor.,
No. de Oficio:
Asunto:

09/enero/2020
SAC/006/2020
Autorización de
impresión de Tesis

ING. ÁNGEL ARTURO RENDÓN CASTRO
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "Sistema embebido para asistencia de conducción basado en Lógica Difusa Tipo-2", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.



ATENTAMENTE
Excelencia en Educación Tecnológica®
"Conocimiento y tecnología al servicio de México"

DR. GERARDO VICENTE GUERRERO RAMÍREZ MICA
SUBDIRECTOR ACADÉMICO

SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN

C.p. M.E. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/ego

Resumen

En este trabajo se presenta el diseño y desarrollo de un asistente de conducción basado en Lógica Difusa Tipo-2 y Visión por Computadora para la detección de nivel de riesgo. El trabajo se divide en tres apartados principales: Adquisición de información, inferencia e implementación. En Adquisición de información se abordan técnicas de Visión por Computadora que permiten una detección fiable en entornos no controlados (detección de puntos de referencia faciales y detección de objetos con margen-máximo). En inferencia se utiliza Lógica Difusa Tipo-2, esta permite trasladar la experiencia humana y el manejo de incertidumbre a un sistema de control, en este trabajo se usa un sistema Lógico Difuso de Intervalo Tipo-2 que realiza una inferencia de nivel de riesgo con la información obtenida del apartado anterior. Por último, se implementa la parte de adquisición e inferencia en la plataforma embebida Nvidia Jetson TX2, creando un prototipo que se equipó en un automóvil donde se realizaron pruebas en ambientes reales.

El sistema de asistencia propuesto se evaluó con métricas de detección (*Recall*, Precisión y *F1-Score*). Se adaptaron las métricas a tres experimentos que evalúan la detección de vehículo frontal, detección de nivel de riesgo por inatención al camino y detección de nivel de riesgo con detección de vehículo frontal en conjunto con la de inatención. Para los experimentos se utilizaron videos obtenidos del sistema además de utilizar las bases de datos *YawDD*, *UTA Real-Life Drowsiness* y *TME Motorway*.

Los resultados cuantitativos y cualitativos, muestran que el enfoque propuesto resulta eficiente para la detección de nivel de riesgo, teniendo un rendimiento mayor a 0.90 en las métricas de evaluación y una implementación en tiempo real con un promedio de 24.63 cuadros por segundo.

Palabras Clave: Lógica Difusa Tipo-2, Visión por Computadora, Asistencia al Conductor y Sistemas Embebidos.

Abstract

In this work is proposed a driving assistant based on fuzzy logic type-2 and computer vision for risk level detection. The work is divided in three main sections: Acquisition of information, inference and implementation. In Acquisition of information, computer vision techniques are approached that allow reliable detection in uncontrolled environments (Detection of facial reference points and max-margin object detection). In inference, fuzzy logic type-2 is used, this allows transfer the human experience and the uncertainty management to a control system. In this work, a Interval Type-2 Fuzzy Logic System is used to perform a risk level inference with the information obtained from the previous section. Finally, the acquisition and inference part was implemented in the embedded Nvidia Jetson TX2 platform, creating a prototype that was equipment in a car where tests were conducted in real environments.

The proposed assistance system was evaluated with detection metrics (Recall, Precision and F1-Score). The metrics were adapted to three experiments that evaluate the detection of frontal vehicle, detection of risk level due to inattention to the road and detection of risk level with detection of frontal vehicle in conjunction with that of inattention. For the experiments, videos obtained from the system were used in addition to using the YawDD, UTA Real-Life Drowsiness and TME Motorway databases.

The quantitative and qualitative results show that the proposed approach is efficient for the detection of risk level, having an average yield above 0.90 in the evaluation metrics and a real-time implementation with an average of 24.63 frames per second.

Keywords: Fuzzy logic type-2, Computer Vision, Driver Assistance and Embedded Systems.

Dedicatoria

A mis padres *Arturo Rendón García y Guadalupe Castro Uriostegui*.
Por su amor y apoyo incondicional.

A *Katy Reyes Bastida*. Te amo, eres el amor de mi vida.

A mi hermana *Nayla Melina Rendón Castro* por ser mi mejor amiga.

A mis *amigos* por permitirme aprender y crecer junto a ellos.

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo brindado durante mis estudios de maestría mediante su sistema de becas de posgrado. Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), perteneciente al TECNM, por haberme permitido realizar los estudios de una Maestría en Ciencias en sus instalaciones.

Agradezco a mi director de tesis Dr. Dante Mújica Vargas por su apoyo, esfuerzo, dedicación y guía brindada durante mi formación. A el comité revisor, Dra. Andrea Magadán y al Dr. Andrés Blanco, por sus aportaciones y esfuerzo para el desarrollo de esta investigación.

Índice General

| | |
|--|-----------|
| Resumen | I |
| Nomenclatura | x |
| 1. Introducción | 1 |
| 1.1. Descripción del Problema | 2 |
| 1.1.1. Delimitación del Problema | 3 |
| 1.1.2. Complejidad del Problema | 3 |
| 1.2. Objetivos | 4 |
| 1.2.1. Objetivo General | 4 |
| 1.2.2. Objetivos Específicos | 4 |
| 1.3. Alcances y Limitaciones | 4 |
| 1.3.1. Alcances | 4 |
| 1.3.2. Limitaciones | 5 |
| 1.4. Justificación | 5 |
| 1.5. Organización de la Tesis | 5 |
| 2. Marco Teórico | 7 |
| 2.1. Sistemas Avanzados de Asistencia de Conducción | 7 |
| 2.2. Visión por Computadora | 8 |
| 2.3. Lógica Difusa | 10 |
| 2.4. Sistemas Embebidos Basados en Cómputo Heterogéneo | 16 |
| 2.5. Discusión | 17 |
| 3. Estado del Arte | 18 |
| 3.1. Antecedentes | 18 |
| 3.2. Trabajos Relacionados | 20 |
| 3.2.1. Discusión del Estado del Arte | 31 |

| | |
|---|-----------|
| 4. Metodología | 40 |
| 4.1. Sistema Propuesto | 40 |
| 4.1.1. Adquisición de la Información | 40 |
| 4.1.2. Inferencia | 46 |
| 4.1.3. Implementación | 53 |
| 4.2. Discusión | 57 |
| 5. Pruebas y Resultados | 58 |
| 5.1. Diseño e Implementación de Pruebas | 58 |
| 5.1.1. Métricas | 58 |
| 5.1.2. Experimentos | 59 |
| 5.2. Resultados | 64 |
| 5.2.1. Detección de Vehículos | 64 |
| 5.2.2. Detección de Inatención | 66 |
| 5.2.3. Detección de Nivel de Riesgo | 68 |
| 5.3. Discusión | 69 |
| 6. Conclusiones | 70 |
| 6.1. Objetivos y Alcances Logrados | 70 |
| 6.2. Resultados del Trabajo | 71 |
| 6.2.1. Aportaciones | 72 |
| 6.2.2. Conclusiones | 72 |
| 6.2.3. Trabajo futuro | 74 |
| A. Documentos correspondientes a participaciones | 80 |

Índice de Figuras

| | |
|--|----|
| 2.1. Recolección de datos con diferentes sensores (<i>Klette and CITR, 2015</i>). | 7 |
| 2.2. Etapas de un sistema de visión por computadora (<i>Inchaurbe Chaterina, 2014</i>). | 9 |
| 2.3. Partes de una función de pertenencia (<i>Ponce-Cruz et al., 2016</i>). | 11 |
| 2.4. Diagrama de bloques de un sistema lógico difuso tipo-1 (<i>Mendel, 2001</i>). | 13 |
| 2.5. Representación 3D de T2FS (<i>Ponce-Cruz et al., 2016</i>). | 14 |
| 2.6. La área sombreada representa la FOU delimitada por la MF superior e inferior. (<i>Ponce-Cruz et al., 2016</i>). | 15 |
| 2.7. Diagrama de bloques de un T2FLS con Reducción de tipo + Defuzzificación (<i>Mendel, 2001</i>). | 15 |
| 3.1. Escenario donde la vista del conductor no está alineada con el vehículo que está delante (<i>Chien et al., 2015</i>). | 20 |
| 3.2. Diagrama de bloques de metodología basada en HOG, bordes, simetría y SVM (<i>Nieto et al., 2014</i>). | 28 |
| 3.3. Detección de vehículos representada en una vista de ave (<i>Palazzi et al., 2017</i>). | 30 |
| 3.4. Representación gráfica de la red de proyección densa con conciencia semántica (<i>Palazzi et al., 2017</i>). | 31 |
| 4.1. Coordenadas de los puntos de referencia facial p contenidas en el vector S (<i>Kazemi and Sullivan, 2014</i>). | 41 |
| 4.2. Incrementó de forma (<i>Kazemi and Sullivan, 2014</i>). | 41 |
| 4.3. Proceso de regresión, (<i>Kazemi and Sullivan, 2014</i>). | 42 |
| 4.4. Predicción de puntos: (a) 68 puntos, (b) 22 puntos. | 43 |
| 4.5. Posición de los puntos de referencia para el cálculo de EAR y MAR. | 43 |
| 4.6. Arquitectura de CNN MMOD. | 45 |
| 4.7. Proceso de detección de la CNN MMOD. | 45 |

| | |
|---|----|
| 4.8. Prueba de recorte y detección con CNN MMOD. | 46 |
| 4.9. Tipos de conjuntos difusos. | 47 |
| 4.10. Diagrama de un IT2FLS. | 48 |
| 4.11. Tipos de fuzzificación. | 48 |
| 4.12. Conjuntos difusos definidos para las diferentes entradas. | 49 |
| 4.13. Conjuntos difusos definidos para la salida. | 49 |
| 4.14. Ilustración de FOU de salida del sistema | 51 |
| 4.15. Ilustración de FOU de salida del sistema. | 53 |
| 4.16. Diagrama del IT2FLS para el asistente de conducción propuesto. . . | 53 |
| 4.17. Diagrama de conexión del sistema. | 55 |
| 4.18. Implementación de las cámaras en el vehículo. | 55 |
| 4.19. Implementación del sistema de asistencia en vehículo. | 56 |
| 4.20. Diseño de Jetson TX2 J21 GPIO (<i>JetsonHacks, 2018</i>). | 56 |
| 4.21. Circuito de conexión de <i>Buzzer</i> | 57 |
| | |
| 5.1. Representación gráfica de <i>IOU</i> | 60 |
| 5.2. Resultados de detectores en secuencias de video no controladas I. . | 60 |
| 5.3. Resultados de detectores en secuencias de video no controladas II. . | 61 |
| 5.4. Detecciones de diferentes niveles de riesgo | 62 |
| 5.5. Pruebas realizadas de detección de nivel de riesgo del sistema de asistencia. | 63 |
| 5.6. Pruebas realizadas de detección de nivel de riesgo del sistema de asistencia. | 64 |
| | |
| A.1. Reconocimiento como ponente en la Escuela de Inteligencia Artificial y Robótica 2018, UTEZ. | 81 |
| A.2. Reconocimiento como participante de poster en Primera Jornada de Ciencia y Tecnología Aplicada, CENIDET 2018. | 82 |
| A.3. Reconocimiento como concursante en ICMEAE, 2018. | 83 |
| A.4. Reconocimiento como co-autor en artículo publicado en CONIELECOMP, UDLAP 2019. | 84 |
| A.5. Constancia de autor de artículo publicado en la Segunda Jornada de Ciencia y Tecnología Aplicada, CENIDET 2018. | 85 |
| A.6. Reconocimiento como ponente en el 1er coloquio " <i>Artificial Intelligence</i> ", Tecnológico de Iguala 2019. | 86 |
| A.7. Artículo enviado a la Escuela de Inteligencia Artificial y Robótica, UTEZ 2019. | 87 |

Índice de Tablas

| | |
|--|----|
| 2.1. Funciones de pertenencia convencionales (<i>Ponce-Cruz et al., 2016</i>). . . | 12 |
| 2.2. Plataformas de la familia <i>Jetson</i> de NVIDIA. | 17 |
| 3.1. Comparación de los resultados del análisis de seguridad entre redes neuronales artificiales y reglas difusas, de un total de 343 casos (<i>Chien et al., 2015</i>). | 21 |
| 3.2. Tiempo de procesamiento (por frame) de ejecución de métodos en diferentes plataformas (<i>Nieto et al., 2014</i>). | 27 |
| 3.3. La exactitud y alerta de LDWS, FCWS y OVDS (<i>Dai et al., 2016</i>). . . | 29 |
| 3.4. Definición de métricas de desempeño (<i>Velez and Otaegui, 2016</i>). . . . | 29 |
| 3.5. Resumen de artículos. | 33 |
| 4.1. Algoritmo EIASC. | 52 |
| 4.2. Características de Jetson TX2. | 54 |
| 4.3. Especificaciones de elementos de Hardware. | 54 |
| 5.1. Resultados para la base de datos <i>TME Motorway</i> | 65 |
| 5.2. Resultados para la base de datos de secuencias de video propias. . . | 65 |
| 5.3. Resultados para la base de datos <i>UTA Real-Life Drowsiness</i> | 66 |
| 5.4. Resultados para base de datos <i>YawDD</i> | 67 |
| 5.5. Resultados para secuencias de video propias. | 67 |
| 5.6. Resultados de detección de nivel de riesgo para secuencias de video propias. | 68 |
| 6.1. Objetivos y alcances realizados. | 70 |

Nomenclatura

| | |
|----------------------------------|--|
| A | Conjunto difuso. |
| $\mu_A(x)$ | Grado de pertenencia de x . |
| \tilde{A} | Conjunto difuso tipo-2. |
| $\mu_{\tilde{A}}(x, u)$ | Grado de pertenencia de x y u . |
| Jx | Pertenencia primaria de x . |
| $\bar{\mu}_{\tilde{A}}(x)$ | Función de pertenencia superior . |
| $\underline{\mu}_{\tilde{A}}(x)$ | Función de pertenencia inferior . |
| S | Vector de forma . |
| $\hat{S}^{(t)}$ | Estimación actual de S . |
| $S_{\pi i}$ | Forma objetivo de S . |
| ΔS_i^0 | Desplazamiento de forma de S . |
| rt | Regresor. |
| I | Imagen de entrada. |
| yl | Punto de cambio de los límites superiores de los intervalos de disparo a los límites inferiores. |
| yr | Punto de cambio de los límites inferiores de los intervalos de disparo a los límites superiores. |
| y | Salida de sistema difuso. |

Capítulo 1

En este Capítulo se abordan los aspectos fundamentales que servirán de soporte, tales como: la descripción, delimitación y complejidad del problema, los objetivos, alcances y limitaciones, así como la justificación.

Introducción

Los sistemas avanzados de asistencia al conductor son herramientas desarrolladas para automatizar y mejorar sistemas vehiculares con el fin de reducir situaciones de riesgo. Estas herramientas buscan minimizar el error humano a través de alertas y recomendaciones al conductor sobre situaciones de riesgo que puedan presentarse y ser difíciles de identificar por un ser humano. La alerta o recomendación idónea es generada comúnmente por algoritmos o métodos que tienen como entrada la fusión de información proveniente de diversos sensores, los sensores utilizados con más frecuencia suelen ser las cámaras por la diversidad y cantidad de información que puede ser obtenida dependiendo de los algoritmos utilizados (Velez and Otaegui, 2016). Los sistemas de asistencia basados en visión procesan imágenes en tiempo real del entorno en busca de eventos de riesgo (obstáculos, colisiones con otros vehículos, estado de inatención del conductor, etc.) y generan como salida una alerta o recomendación correspondiente.

Un inconveniente en los asistentes basados en visión reside en la inherente incertidumbre de la información generada por características de entornos no controlados, como la presencia de variaciones en iluminación, movimiento por estado precario del camino u otros factores. Un enfoque de solución es usar técnicas robustas de detección en conjunto con una inferencia basada en lógica difusa, en especial del Tipo-2 que permite un manejo de información imprecisa y la exportación de la experiencia humana a través de reglas definidas por un experto.

En esta tesis, se presenta un sistema embebido de asistencia de conducción basado en Lógica Difusa Tipo-2 (FLT2) y Visión por Computadora. Este asistente utiliza información del camino (detección de vehículos frontales) y del monitoreo del conductor (detección de nivel de inatención) para una posterior inferencia con base en FLT2 del nivel de riesgo.

La implementación de asistente propuesto se realiza en el sistema embebido *Jetson TX2* que lleva a cabo las tareas de detección e inferencia en tiempo real. La elección del sistema embebido corresponde a la cantidad y complejidad de información necesaria para realizar las tareas de detección con técnicas robustas de visión por computadora e inferencia con un sistema de control basado en Lógica Difusa Tipo-2 que requiere la realización de una cantidad considerable de operaciones.

1.1. Descripción del Problema

Una de las razones del aumento de accidentes de tráfico se debe a la falta de atención del conductor al camino y a diversos obstáculos ignorados que pueden presentarse a simple vista o en puntos ciegos. La inatención al camino puede ser generada por varios factores, tales como el estado de somnolencia del conductor y el enfoque de atención del conductor en otras actividades, esta inatención con frecuencia conlleva a situaciones de riesgo que abarcan desde una pérdida material hasta accidentes mortales. Con el objetivo de tratar de minimizar el error humano en la actividad de conducción se implementan sistemas para asistencia que brindan alertas y/o recomendaciones con base a la detección de obstáculos en el entorno y el monitoreo del conductor en tiempo real, de tal forma que se detecten y alerten posibles situaciones de riesgo que representen un peligro para el conductor y los pasajeros del vehículo.

Este trabajo de tesis tiene por objeto desarrollar un sistema embebido para asistencia al conductor en tiempo real que pueda trabajar con información que contenga incertidumbre inherente a entornos reales. El sistema propuesto se basa en Lógica Difusa Tipo-2 para la inferencia y Visión por Computadora para la obtención de información.

Hipótesis: El uso de Lógica Difusa Tipo-2 puede ser mejor para el manejo de información obtenida de entornos reales, además de que podría permitir una evaluación del nivel de riesgo más certera, tal como lo haría un humano. Las técnicas de Visión por Computadora robustas para detección podrían obtener información del estado de alerta del conductor (bostezos/parpadeos y posición de la cabeza) y detección de objetos (vehículos frontales) en tiempo real y de una forma eficiente. Considerando a su vez una implementación del sistema en una plataforma embebida basada en cómputo heterogéneo para un procesamiento óptimo de las operaciones que conlleva la Lógica Difusa Tipo-2 y las técnicas de Visión por Computadora empleadas.

1.1.1. Delimitación del Problema

En este trabajo se realiza un sistema para asistencia de conducción en un sistema embebido, obtiene la información a procesar mediante técnicas de Visión por Computadora y el sistema infiere situaciones de riesgo bajo el enfoque de Lógica Difusa Tipo-2. En el aspecto de la visión existen diversas técnicas comprobadas eficazmente en el Estado del Arte, por lo cual se realizó la experimentación con diversas técnicas para la detección de eventos/objetos de interés (por ejemplo bostezos, parpadeo, posición de la cabeza, objetos en el camino, etc.). El objetivo de este trabajo no es desarrollar una nueva técnica de Visión por Computadora, simplemente se considera el uso y modificación de estas técnicas para obtener la información deseada. Este trabajo se limita a la inferencia del nivel de riesgo mediante el monitoreo del estado de inatención del conductor y la detección de vehículos frontales en el camino.

1.1.2. Complejidad del Problema

Las complejidad del problema radica en los siguientes puntos:

- Diseño y desarrollo de un controlador basado en Lógica Difusa Tipo-2 para la detección de nivel de riesgo en la tarea de conducción.
- Desarrollo de un sistema de asistencia, que trabaje lo más cercano en tiempo real, con el procesamiento de la técnicas de visión y el controlador basado en Lógica Difusa Tipo-2.
- Diseño y desarrollo de un prototipo del sistema de asistencia para su posterior implementación para pruebas en entornos reales.

- La inexistencia de una comparación justa con otros asistentes de conducción debido a que no existe ningún procedimiento estándar para la recopilación y etiquetado de los conjuntos de datos de video utilizados para calcular métricas; además, de que la longitud y la variedad de los conjuntos de datos difieren significativamente de un trabajo a otro.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un sistema embebido para asistencia de conducción basado en Lógica Difusa Tipo-2 y Visión por Computadora.

1.2.2. Objetivos Específicos

- Analizar la Lógica Difusa tipo-2 para el desarrollo del sistema asistente de conducción.
- Experimentación con técnicas de Visión por Computadora para obtención e interpretación de secuencias de imagen.
- Analizar los sistemas embebidos en computación heterogénea.
- Realizar un sistema de alertas sonoro para el sistema.
- Implementar el sistema desarrollado en el sistema embebido seleccionado.
- Evaluar los resultados obtenidos con métricas del estado del arte.

1.3. Alcances y Limitaciones

1.3.1. Alcances

- Experimentar con técnicas de Visión por Computadora para la identificación de obstáculos y monitoreo del conductor.
- Desarrollo del sistema propuesto usando Lógica Difusa Tipo-2.
- Realizar pruebas experimentales del sistema en ambientes reales.
- Evaluar el desempeño del sistema propuesto.

1.3.2. Limitaciones

Aspectos quedan fuera de su cobertura para este trabajo de tesis:

- Parámetros circunferenciales:
 - Clima (lluvia, niebla, nieve, etc.)
 - Condiciones de iluminación demasiado variantes.
 - Oclusión prominente en el monitoreo del conductor por el uso de accesorios o rasgos físicos (barba, cicatrices, etc.).
- Condiciones demasiadas precarias de caminos.

1.4. Justificación

Los sistemas de asistencia basados en visión son los más populares debido a su bajo costo y la diversa cantidad de información que puede obtenerse dependiendo de las técnicas o algoritmos implementados. Sin embargo, al estar basados en visión pueden presentar niveles altos de incertidumbre debido a factores presentes en entornos reales, tales como: variaciones de iluminación, vibración, oclusión, posición de cabeza del conductor, etc. Se utiliza Lógica Difusa Tipo-2 debido a que posee la propiedad de trasladar la experiencia humana a un sistema de control y de trabajar con información con incertidumbre. En este trabajo la información se obtiene con técnicas de Visión por Computadora, por lo cual se consideran técnicas que son tolerantes a entornos no controlados. Implementar este enfoque de solución podría brindar resultados aceptables, pero debido al procesamiento requerido para ejecutar las técnicas de visión y el controlador difuso tipo-2 se contempla el uso de una plataforma embebida de alto desempeño para despliegue de aplicaciones de inteligencia artificial.

1.5. Organización de la Tesis

La presente tesis se organiza de la siguiente manera:

En el Capítulo 2 se describen los conceptos teóricos necesarios para comprender las técnicas de visión utilizadas y los fundamentos de Lógica Difusa Tipo-2.

En el Capítulo 3 se presenta un análisis de los trabajos más recientes relacionados con los sistemas de asistencia de conducción, fusión de sensores, técnicas de visión y controladores difusos.

En el Capítulo 4 se describe a detalle el sistema de asistencia propuesto. Se divide en tres partes principales: adquisición de información, inferencia e implementación.

En el Capítulo 5 se describen experimentos y resultados obtenidos de la evaluación conforme a conceptos y métricas de detección.

Finalmente, en el Capítulo 6, se exponen las conclusiones obtenidas, recomendaciones para trabajos futuros, así como un análisis de los objetivos completados y de los alcances planteados en esta investigación.

Capítulo 2

Marco Teórico

En esta sección se detallan diversos conceptos básicos y específicos que son necesarios para el correcto entendimiento y desarrollo del tema de tesis propuesto.

2.1. Sistemas Avanzados de Asistencia de Conducción

Los sistemas avanzados de asistencia de conducción son desarrollados para automatizar, adaptar y mejorar sistemas vehiculares para un mayor nivel seguridad y mejora de manejo. El sistema automatizado que implementa un sistema de asistencia en el vehículo es previsto para reducir accidentes, minimizando el error humano (*Hamid et al., 2017*). Estos sistemas usualmente hacen uso de diversos sensores como cámaras, Lidar, radar, GPS, etc (como se muestra en la Figura 2.1), procesando la información recolectada para una oportuna toma de control del automóvil o la emisión de una alerta.

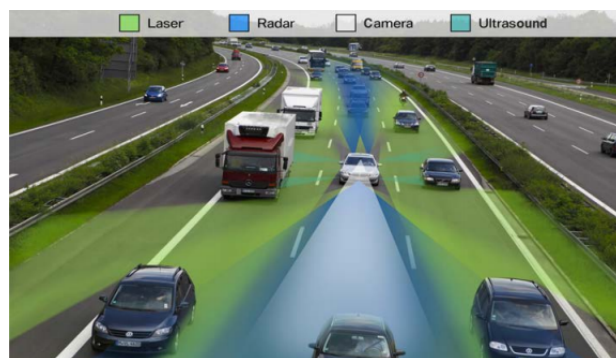


Figura 2.1: Recolección de datos con diferentes sensores (*Klette and CITR, 2015*).

Sistemas de Asistencia de Conducción Basados en Visión

Los Sistemas de Asistencia de Conducción Basados en Visión (VB-DAS) pertenecen a la clase general de sistemas de asistencia al conductor, estos usan cámaras como sensores. Los VB-DAS comenzaron como soluciones para la supervisión de salida de carril o puntos ciegos. Con el desarrollo de técnicas de visión más complejas se adaptaron más usos para el despliegue de diversas tareas relacionadas con la conducción en tiempo real, como por ejemplo: detección de carril, somnolencia, distracción, colisiones, obstáculos en el camino, señales de tráfico, etc. Los VB-DAS se conforman por una o varios tipos de cámaras, una unidad de procesamiento e interfaces/componentes del vehículo relacionados con el control o la comunicación entre el conductor y el vehículo que permiten detectar, alertar y actuar ante situaciones de riesgo (*Klette and CITR, 2015*).

En este trabajo de investigación se desarrolla un VB-DAS, por lo que son necesarios algunos temas relacionados de Visión por Computadora, que son explicados en la siguiente sección.

2.2. Visión por Computadora

La Visión por Computadora consiste en adquirir, procesar, analizar y comprender imágenes del mundo real con el fin de conocer el contenido que hay en ellas para la obtención de información. Este proceso se realiza mediante una serie de transformaciones aplicadas a la imagen de manera que la información de interés predomine en la imagen por encima de la información que no es relevante (*Inchaurbe Chaterina, 2014*). En la Figura 2.2 se pueden observar diversas etapas de que componen comúnmente un sistema de Visión por Computadora y que son descritas a continuación:

- **Adquisición de Imagen:** Proceso por el cual se obtiene una imagen visual desde una cámara.
- **Pre-Procesamiento:** Conjunto de técnicas que facilitan el procesado posterior. Incluye técnicas tales como la reducción de ruido y realce de detalles. Las cámaras monoculares, que son el sensor más común, con este tipo de sensor se procesa la información en un espacio bidimensional, espacio imagen (proyección 2D de una escena 3D).

- **Segmentación:** Proceso que divide a una imagen en objetos de interés para el estudio.
- **Descripción:** Proceso mediante el cual se obtienen características para diferenciar un tipo de objeto de otro.
- **Reconocimiento e Interpretación:** Proceso que identifica a los objetos de una escena y asocia un significado a un conjunto de objetos reconocidos.

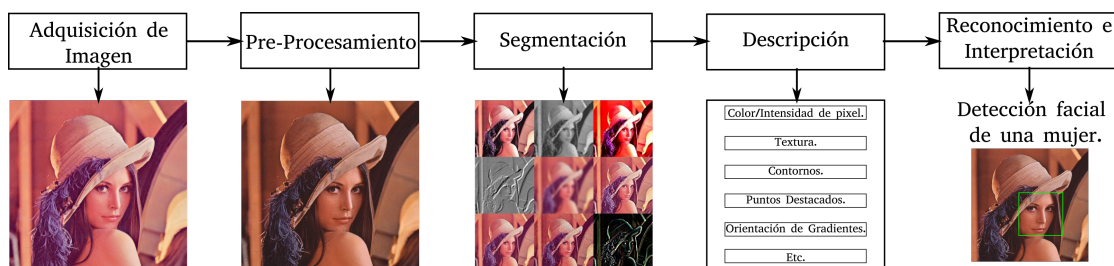


Figura 2.2: Etapas de un sistema de visión por computadora (Inchaurbe Chaterina, 2014).

Los sistemas de visión pueden encontrarse en distintos tipos de nivel de procesamiento, esta clasificación representa la complejidad de abstracción de la información contenida en la imagen. Usualmente los niveles de visión se dividen en tres, el primero para mejora y realce de detalles, el segundo para obtener características y el tercero para análisis e interpretación. Según (Inchaurbe Chaterina, 2014) los tres niveles mencionados se describen de la siguiente forma:

- **Visión de bajo nivel:** Comprende la captación y el preprocesamiento. Ejecuta algoritmos típicamente de filtrado, restauración de la imagen, realce, extracción de contornos, etc.
- **Visión de nivel intermedio:** Comprende la segmentación, descripción y reconocimiento, con algoritmos típicamente de extracción de características, reconocimiento de formas y etiquetado de éstas.
- **Visión de alto nivel:** Comprende la fase de interpretación, normalmente estos algoritmos se refieren a la interpretación de los datos generalmente mediante procedimientos típicos de la Inteligencia Artificial para acceso a bases de datos, búsquedas, razonamientos aproximados, etc.

Las técnicas de Visión por Computadora seleccionadas, que conforman el sistema de adquisición del sistema, brindan la información que define las entradas que ingresan al sistema de inferencia del asistente para detectar el nivel de riesgo. En esta investigación se utiliza un sistema de control basado en Lógica Difusa tipo-2 para la inferencia, los conceptos básicos para comprender esta teoría de conjuntos se abordan en la siguiente sección.

2.3. Lógica Difusa

La lógica difusa es una teoría de conjuntos propuesta por (Zadeh, 1987). En esta teoría se definen conjuntos difusos como una clase cuyos elementos cuentan con grados de pertenencia desde 0 a 1, entre más cercano sea el grado de pertenencia a 1 representa una mayor pertenencia y entre más cercana a 0 el caso contrario, (Ponce-Cruz *et al.*, 2016). Los conjuntos difusos tratan de modelar la incertidumbre relacionada al razonamiento natural humano, que se expresa con palabras y oraciones lingüísticas en lugar de expresiones matemáticas, el razonamiento difuso es un modo de razonamiento que no es exacto ni inexacto, para entenderlo, se deben comprender tres conceptos básicos:

- **Variable Lingüística.** Variable cuyos valores son palabras u oraciones en un lenguaje natural o artificial en lugar de numérico. Por ejemplo, la temperatura se puede describir como se presenta a continuación.

$$TS(\text{temperatura}) = \{\text{Frío}, \text{Tibio}, \text{Caliente}\} = \{F, T, C\}.$$

- **Proposición difusa.** Declaración expresada en un lenguaje natural o artificial. A diferencia de las proposiciones lógicas clásicas, puede adoptar un valor de verdad del intervalo $[0, 1]$. Por ejemplo, la temperatura es caliente.
- **Regla Lingüística.** Sentencia IF-THEN que se compone de dos partes: causa= IF {Proposición difusa}, consecuencia= THEN {Proposición difusa}.

Un conjunto difuso es caracterizado por la función de pertenencia 2.1, la cual asigna a cada elemento un grado de pertenencia. Un conjunto difuso A es definido por un conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad y \quad \mu_A \in [0, 1] \quad (2.1)$$

Donde x es un elemento del universo U y μ_A es la función de pertenencia que se asigna a un grado de pertenencia $\mu_A(x)$ para cada elemento x de A (Mendel, 2001).

Conjunto Difuso Tipo-1 (FST1)

Usualmente, el razonamiento humano en la toma de decisiones no es definido con métodos matemáticos, así que, los números difusos pueden ser usados para resolver problemas sencillos y avanzados que lidian con condiciones ambiguas. Los conjuntos difusos se utilizan para describir la falta de claridad en función de los grados de membresía y se pueden usar en muchas situaciones reales con términos lingüísticos (Ponce-Cruz et al., 2016).

Función de Pertenencia

La función de pertenencia transforma cada elemento de X a un grado de pertenencia entre 0 y 1, (Mendel, 2001). Para entender mejor las funciones de pertenencia, se definen las siguientes nomenclaturas que se observan en la Figura 2.3 y se definen de la siguiente forma:

- **Soporte:** el soporte de un conjunto difuso X , es el conjunto de todos los puntos x en X tal que $\mu_A(x) > 0$:

$$\text{Soporte}(A) = \{x | \mu_A(x) > 0\}. \quad (2.2)$$

- **Núcleo:** es el conjunto de todos los puntos x en X tal que $\mu_A(x) = 1$:

$$\text{Núcleo}(A) = \{x | \mu_A(x) = 1\}. \quad (2.3)$$

- **Fronteras:** definen como aquellas regiones del universo que contienen elementos que tienen una pertenencia distinta de cero pero no una pertenencia completa.

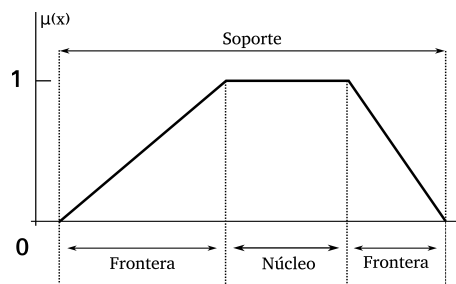


Figura 2.3: Partes de una función de pertenencia (Ponce-Cruz et al., 2016).

Tipos de Funciones de Pertenencia

Existen varias funciones de pertenencia (MF) que se pueden usar en T1FS; por ejemplo, las funciones normales de membresía presentadas anteriormente. Un conjunto fuzzy normal o convencional es aquel cuya MF tiene al menos un elemento x en el universo, cuyo valor de pertenencia, es uno. La representación matemática para algunas MFs convencionales se presentan en la Tabla 2.1.

Tabla 2.1: Funciones de pertenencia convencionales (*Ponce-Cruz et al., 2016*).

| Función de Pertenencia | Parámetros | Forma |
|------------------------|--|-------|
| Triangular | $\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x \leq a \vee x \geq c \end{cases}$ | |
| Trapezoidal | $\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x \leq a \vee x \geq d \end{cases}$ | |
| Forma-S | $\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \\ 0 & x \leq a \end{cases}$ | |
| Forma-Z | $\mu_A(x) = \begin{cases} 1 & x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x \geq d \end{cases}$ | |
| Gaussiana | $\mu_A(x) = e^{-\frac{1}{2}(\frac{x-\epsilon}{\sigma})^2}$ | |
| Singleton | $\mu_A(x) = \begin{cases} 1 & x = c \\ 0 & \text{otros} \end{cases}$ | |
| Intervalo | $\mu_A(x) = \begin{cases} 1 & a \leq x \leq b \\ 0 & \text{otros} \end{cases}$ | |

Sistema Lógico Difuso Tipo-1 (T1FLS)

Un sistema lógico difuso tipo-1 basado en reglas contiene cuatro componentes principales: fuzzificación, base de conocimiento, inferencia y defuzzificación, que están interconectados como se muestra en la Figura 2.4 y se describen a continuación.

- Fuzzificación: convierte un valor de entrada discreto a un dominio difuso continuo por medio de una MF.
- Base de conocimiento: reglas canónicas conformadas por antecedentes y consecuentes de estructura *IF-THEN*; son establecidas de forma coherente por un experto; además, determina las formas, cantidad y umbrales de los conjuntos que se trabajaran para cada variable.
- Inferencia: este bloque asigna a la entrada difusa una salida difusa de acuerdo con las reglas establecidas y operaciones de conjuntos difusos.
- Defuzzificación: asignación de una entrada difusa de tipo-1 en una salida discreta que puede ser interpretada en forma de palabras (Por ejemplo: muy frío, tibio o demasiado caliente).

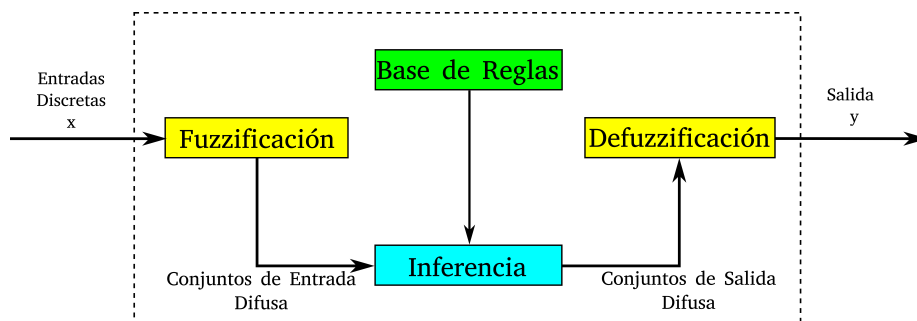


Figura 2.4: Diagrama de bloques de un sistema lógico difuso tipo-1 (Mendel, 2001).

Lógica Difusa Tipo-2

La Lógica Difusa Tipo-2 es una extensión del tipo-1. Son la respuesta a problemas donde se cuenta con entradas con incertidumbre, ruido o imprecisión estadística y los conjuntos de lógica difusa tipo-1 no son adecuados o suficientes para la implementación y obtención de resultados óptimos. Los conjuntos difusos tipo-2 (FST2) proporcionan información adicional en una MF secundaria, esto significa, que no solo se depende sobre una variable " x " si no también en una variable " u " adicional, estos conjuntos difusos son representados por:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X \quad \forall u \in U \leftarrow [0, 1]\} \quad (2.4)$$

Donde \tilde{A} represente el conjunto difuso tipo-2 y $\mu_{\tilde{A}}(x, u)$ el grado de pertenencia que depende de las variables x y u . Los FST2 son usualmente representados en dos dimensiones y una dimensión extra auxilia a la representación de estos conjuntos *Karnik and Mendel (2001)*, en la Figura 2.5 puede observarse esta representación.

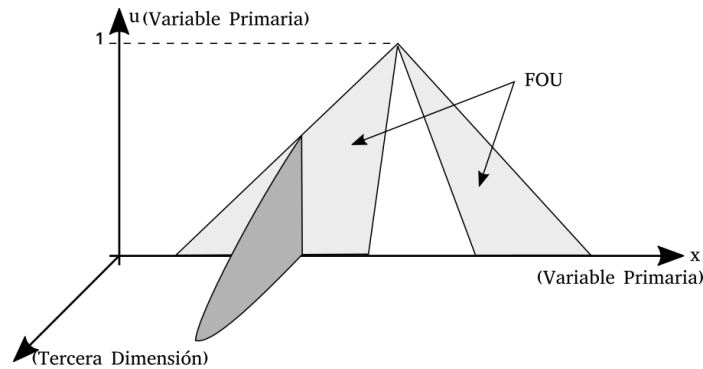


Figura 2.5: Representación 3D de T2FS (*Ponce-Cruz et al., 2016*).

La dimensión adicional brinda un grado extra de incertidumbre y, por lo tanto, algunas posiciones de la MF tienen un mayor grado que otros.

Huella de Incertidumbre

La huella de incertidumbre (FOU) es la unión de la MF primaria y la región delimitada que representa la incertidumbre en la MF principal de un T2FS. Se definen las MFs superior e inferior que son los límites de la FOU, estas funciones se representan por:

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x \quad (2.5)$$

Donde J_x es un intervalo $[0, 1]$ que representa la pertenencia primaria de x y la MF superior e inferior son dos funciones de tipo-1, los cuales rodean la T2FS (ver Figura 2.6).

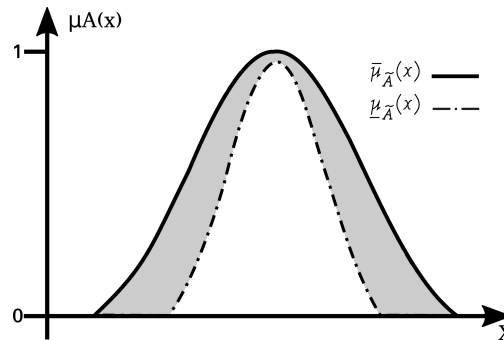


Figura 2.6: La área sombreada representa la FOU delimitada por la MF superior e inferior. (Ponce-Cruz et al., 2016).

La MF superior 2.6 es denotada por $\bar{\mu}_{\tilde{A}}(x)$ y la MF inferior 2.7 es denotada por $\underline{\mu}_{\tilde{A}}(x)$

$$\bar{\mu}_{\tilde{A}}(x) = \max(J_x) \quad \forall x \in X \quad (2.6)$$

$$\underline{\mu}_{\tilde{A}}(x) = \min(J_x) \quad \forall x \in X \quad (2.7)$$

Sistema Lógico Difuso Tipo-2 (T2FLS)

La Figura 2.7 presenta el diagrama de bloques para un T2FLS que es similar a un sistema T1FLS, con la diferencia de que se trabaja con conjuntos y operaciones difusas tipo-2. En este tipo de sistemas de control se introducen los siguientes componentes:

- Reducción de tipo: para algunos sistemas, se requiere transformar las salidas difusas de tipo 2 del motor de inferencia a T1FS y el resultado se denomina conjunto reducido de tipo.
- Defuzzificación: una vez que las salidas se han reducido, el bloque de Defuzzificación determina el valor puntual que se introducirá en algún actuador.

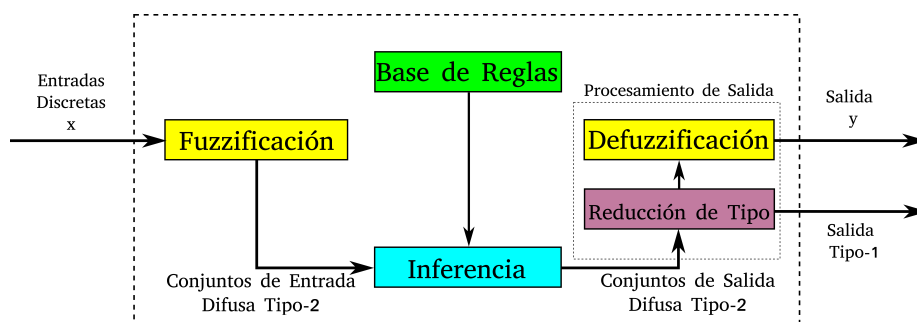


Figura 2.7: Diagrama de bloques de un T2FLS con Reducción de tipo + Defuzzificación (Mendel, 2001).






Para la implementación en tiempo real, de la adquisición de información e inferencia, en un prototipo funcional que pueda operar con procesamientos complejos se contempla el uso de sistemas embebidos. Para el entendimiento de estos sistemas embebidos se abordan conceptos esenciales en la sección contigua.

2.4. Sistemas Embebidos Basados en Cómputo Heterogéneo

Los sistemas embebidos heterogéneos son la respuesta de los sistemas integrados a las altas demandas para satisfacer requerimientos de aplicaciones modernas con el fin de obtener un mayor poder de cómputo, estos sistemas hacen uso de una arquitectura heterogénea distribuida en múltiples y diferentes unidades de procesamiento que son capaces de responder a diversas tareas de nivel complejo de procesamiento. Hacen uso comúnmente de un CPU para actividades secuenciales y una GPU para procesamiento de operaciones en paralelo. En sistemas heterogéneos, la arquitectura de cada nodo puede diferir, por lo que las características de los nodos pueden variar. Cada nodo puede tener diferente velocidad de núcleo máxima y mínima, o un nivel de consumo de energía diferente (*Huang et al., 2017*).

Los sistemas embebidos de la línea *Jetson*, producidos por la empresa NVIDIA, son ejemplos de sistemas basados en computación heterogénea de baja potencia que son diseñados para implementar aplicaciones de inteligencia artificial o de alta complejidad en prototipos funcionales. Los modelos Jetson contienen un procesador de arquitectura ARM que trabaja en conjunto con una unidad gráfica de procesamiento (GPU) para realización de procesamiento de aplicaciones complejas y de alta demanda computacional. En la Tabla 2.2 pueden observarse diferentes tipos de modelos de sistemas Jetson y sus características más relevantes.

Tabla 2.2: Plataformas de la familia *Jetson* de NVIDIA.

| | TK1 | NANO | TX1 | TX2 (TX2 4GB/TX2/TX2i) | | AGX XAVIER |
|-------------------------|---|---|--|---|---|--|
| |  |  |  |  |  | |
| Desempeño | 326 GFLOPS | 472 GFLOPs | 1TOPs | 1.3TOPs | | 32TOPs |
| GPU | 192-cores Kepler | 128-core Maxwell | 256-core Maxwell | 256-core Pascal | | 512-core Volta con 512 Tensor Cores |
| CPU | Quad-core ARM Cortex-A1 | Quadcore ARM A57 MPCore | Quadcore ARM A57 MPCore | Dual-Core Denver y Quadcore ARM A57 MPCore | | 8-Core ARM 8.2 Carmel |
| Memoria | 2GB | 4GB | MPCore | 4GB | 8GB | 16GB |
| Almacenamiento | 16GB | 16GB | 16GB | 16GB | 32GB | 32GB |
| Consumo | 1.6W/4.7W | 5W/10W | 10W | 7.5W/15W | 10W/20W | 10W/15W/30W |
| Acelerador de visión | - | - | - | - | - | Si |
| Wi-Fi | No | No | Si | No | Si | No |

2.5. Discusión

Este capítulo presentó un panorama general sobre los temas relacionados con el trabajo de investigación que se desarrolla en este tema de tesis, abarcando los conceptos básicos de sistemas de asistencia de conducción basados, Visión por Computadora, Lógica Difusa Tipo-2 y sistemas embebidos. La información presentada en este capítulo es requerida para en el entendimiento fundamental del diseño, adquisición de información, método de inferencia e implementación que será de vital importancia en capítulos siguientes.

Capítulo 3

Estado del Arte

En este capítulo se presenta una breve descripción de trabajos de investigación relevantes relacionados con los temas de asistentes de conducción, Lógica Difusa Tipo-2, técnicas de Visión por Computadora y sistemas embebidos.

3.1. Antecedentes

Los trabajos mostrados a continuación fueron desarrollados en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) y abordan temas relacionados con este trabajo de investigación.

Modelado del Comportamiento de Conducción de Vehículos de Transporte y/o Carga Aplicando Redes Neuronales Artificiales y Visión Artificial (*Siordia, 2009*).

En este trabajo se propone una aproximación al modelado de conductores de camiones de transporte y/o carga. Se utiliza un clasificador de comportamiento basado en Redes Neuronales Artificiales (RNAs). Es alimentado con datos provenientes de ejercicios de conducción realizados en una cabina de simulación del proyecto Cabina Inteligente para el Transporte por Carretera (CABINTEC) y es apoyado con un Sistema de Visión por Computadora que le proporciona información del comportamiento del conductor, en específico la posición de las manos del conductor durante las sesiones simuladas.

Para el sistema de visión se utilizaron cuatro sensores CMOS de cinco megapíxeles con un convertidor analógico a digital de 12 bits, capaz de entregar más de 200

imágenes por segundo en una resolución de 320x240. Se convierten las imágenes obtenidas al formato RGB, dando a los valores de los dos canales faltantes de cada píxel, un valor o promedio resultante de los vecinos más cercanos que contienen información de color de los canales.

Fusión de Datos de Sensores Mediante lógica difusa, Aplicación en Robótica Móvil (Gómez, 2014).

En esta tesis se describe la navegación de un robot móvil autónomo, mediante el uso de fusión de sensores y lógica difusa tipo-1. Se realizó la fusión de un dispositivo Kinect y sensores ultrasónicos para plantear un esquema de minimización de errores sistemáticos y no sistemáticos.

En el trabajo propuesto se implementó lógica difusa como un método de toma de decisiones para el robot móvil, programado en un sistema embebido. El robot móvil realizó la navegación reactiva, mediante la toma de información proveniente de la fusión de sensores y reglas difusas establecidas, se determinaba la acción a tomar.

Extracción de Superpíxeles y Fusión Local Mediante un Algoritmo de Agrupamiento Difuso Intuicionista (López, 2019).

En este trabajo de tesis se propone un método para la extracción de superpíxeles basado en Agrupamiento Difuso Intuitivo. Se utiliza el enfoque difuso para el manejo de incertidumbre. Se aborda el problema del tratamiento de imágenes complejas, en específico, las imágenes de resonancia magnética cerebral. El método se comparó con cinco algoritmos para extraer superpíxeles. Para la evaluación se utilizó dos estudios de resonancia magnética cerebral del simulador BrainWeb. El método se evaluó con las métricas: Rec (Re-cuerdo de límites), UE (Error de subsegmentación) y EV (Variación Explicada). Los resultados cuantitativos y cualitativos, demostraron que el enfoque propuesto generó superpíxeles con alta adherencia a los bordes naturales de la imagen, teniendo un rendimiento similar y en algunos casos mejor que los algoritmos de comparación.

Discusión de los Antecedentes.

Estos trabajos se conforman de investigaciones que brindan un primer panorama de comprensión y orientación que sirven como una referencia local sobre los conceptos

básicos de los temas de fusión de sensores, lógica difusa, visión por computadora y sistemas embebidos.

3.2. Trabajos Relacionados

En esta sección se revisaron los trabajos más relevantes encontrados en la literatura relacionados a los temas de interés de esta investigación. También se presenta una breve descripción de las metodologías de desarrollo, implementación y evaluación utilizadas.

Un sistema de asistencia al conductor basado en reglas difusas (*Chien et al., 2015*).

En este trabajo se propuso el uso de módulos basados en reglas difusas para la detección de obstáculos en el camino, puntos ciegos, detección de maniobras ilegales y determinación del ángulo de vista del conductor. Se implementó un clasificador *Cost-Sensitive AdaBoost (CS AdaBoost)*, con el fin de llevar a cabo el desarrollo de un VB-DAS que clasifica tres niveles diferentes de seguridad: Seguro, Riesgoso y Peligroso. El sistema desarrollado fue puesto a prueba en un escenario real, imágenes de la experimentación se muestran en la Figura 3.1.



(a) Ángulo de vista del conductor. (b) Vehículo delantero (c) Activación de reglas detectado.

Figura 3.1: Escenario donde la vista del conductor no está alineada con el vehículo que está delante (*Chien et al., 2015*).

Los resultados experimentales demostraron un 94.16% de eficacia de identificación de vehículo con *CS AdaBoost* contra un 90.15% del *AdaBoost* convencional. Además, comparan la exactitud del método de análisis de seguridad del sistema basado en reglas difusas contra redes neuronales artificiales, tal y como se observa en la Tabla 3.1.

Tabla 3.1: Comparación de los resultados del análisis de seguridad entre redes neuronales artificiales y reglas difusas, de un total de 343 casos (*Chien et al., 2015*).

| Método de análisis de seguridad | Precisión | Falsa alarma/fallo | Exactitud |
|---|-----------|--------------------|-----------|
| ANN(9 nodos ocultos) | 312 | 31 | 90.9% |
| ANN(12 nodos ocultos) | 314 | 29 | 91.4% |
| Extracción automática de reglas difusas | 321 | 22 | 94.5% |

Sistema integrado de asistencia al conductor con múltiples sensores para una maniobra segura de adelantamiento (*Chiang et al., 2014*).

Este trabajo propuso el uso de un procesador de señal digital embebido de bajo costo para la construcción de un asistente de conducción, el cual se implementó en un vehículo comercial. El sistema obtuvo datos de múltiples entradas como lo son: sistema de visión en tiempo real, comunicación vehículo a vehículo y sensores del vehículo. El sistema propuesto brindó la posibilidad de facilitar la toma de decisiones y la realización de tareas de conducción. Para la realización de maniobras automáticas, dirección y velocidad, se usó control difuso. El control fue llevado a cabo por dos controladores difusos, cada uno con cuatro componentes principales: una base de reglas, lógica de toma de decisiones, interfaz de entrada de fuzzificación y una interfaz de salida de defuzzificación. Los controladores se implementaron usando reglas de tipo Mamdani.

Optimización de sistema de asistencia basado en visión por computadora: estudio de caso de detección de vehículos (*Nieto et al., 2016*).

En este trabajo se presentó el diseño y desarrollo de una metodología que se enfocó en dos retos principales para los desarrolladores de asistentes de conducción: la optimización del algoritmo de desempeño y del ciclo de desarrollo. Esta metodología propuso la optimización de los algoritmos de visión sin la necesidad de re-codificar o incluir procedimientos de optimización. La metodología propuesta se ilustró con un caso de estudio de detección de vehículos, haciendo uso de la optimización del ciclo de desarrollo mediante el uso de un sistema operativo de acceso libre, en el cual se escribía y verifica el código con capacidades multiplataforma. La optimización del algoritmo de desempeño se centró en una

serie de nueve prácticas aplicables al contexto mediante el impacto esperado y el nivel/costo de implementación, en la cual se utilizó HOG, *AdaBoost* y SVM. El desarrollo de la metodología con los aspectos comentados resultó en cuatro prototipos de metodología del caso de estudio: multiescala, perspectiva multiescala, perspectiva de pre-detección multiescala y cambio de día/noche.

Detección de vehículos a escala múltiple basada en visión y seguimiento relevante de distancia combinada para sistemas de asistencia de conductor (Gu et al., 2015).

Los autores del artículo presentaron un método de detección y rastreo de vehículos a escala múltiple basada en visión para sistemas de asistencia al conductor. Proponen una técnica de transformación de perspectiva inversa, donde mediante el uso de un operador morfológico multiescala se obtiene la detección de la placa y esqueleto del vehículo que produce un sistema de coordenadas lineales para la detección del vehículo completo; además, de utilizar SVM con características HOG para la detección robusta en áreas cerradas. Como resultado un clasificador se entrenó para detectar vehículos de largo alcance con características similares a Haar.

Diseño e implementación de cámara vehicular en tiempo real para asistencia al conductor y estimación de congestión de tráfico (Son and Baek, 2015).

Este trabajo propuso la implementación de un sistema de asistencia en una plataforma NVIDIA *Jetson* basado en Tegra Kepler1 para el reconocimiento de peatones, matrícula y señales de tráfico. Presentaron dos nuevos métodos: regiones de interés dinámicas para la reducción de área de búsqueda y salto de cuadro dinámico para descartar procesamiento innecesario. El proceso de reconocimiento es realizado con la ayuda de un HOG que calcula los cambios en cada pixel y después se hizo uso de una SVM para comparar el valor de la gradiente obtenido. También introdujeron un método para estimar la congestión del tráfico utilizando la distancia entre vehículos, la velocidad del vehículo y el número de vehículos vecinos.

Detección de señales de tráfico basadas en FPGA para sistemas de cámaras automotrices (Schwiegelshohn et al., 2015).

En este trabajo se propuso la introducción de un sistema de procesamiento de imágenes RGB en formato FullHD, para la única tarea de detección de señales de tráfico y de límite de velocidad, implementado en una plataforma Spartan-6-FPGA. En la implementación de el espacio de color RGB, se propuso una conversión a espacio de color YC_bC_r con el fin de prevenir las diferencias en los componentes de color que son resultado de variaciones en diversas condiciones de luz. La extracción de características se realizó con un algoritmo de desborde donde los valores horizontales y verticales mínimos y máximos son guardados para después representar un cuadro delimitador que detecta la presencia de características y si están dentro o fuera de las regiones de características esperadas.

Implementación y evaluación del algoritmo de reconocimiento de imagen para un vehículo inteligente utilizando SoC heterogéneo de múltiples núcleos (Ozaki et al., 2015).

Este trabajo tuvo por objetivo el desarrollo de un sistema en un chip multi-núcleo heterogéneo TMPV750 que se especializó en reconocimiento de imagen. Se implementaron tareas de advertencia de colisión frontal, advertencia de salida de carril y detección de peatones. El sistema utilizó un procesador multi-núcleo que cuenta con cuatro núcleos de conjunto reducido de instrucciones de computadora (RISC), además se implementaron en paralelo para las tareas típicas en aplicaciones de reconocimiento de imagen dos filtros, un histograma de co-ocurrencia de gradientes orientados (CoHOG), un afinado y aceleradores coincidentes. El algoritmo CoHOG y los diversos aceleradores resultaron en una estimación del rango de detección entre los 20 y 30 metros con un clima favorable y ejecución en tiempo de entre 32 y 48 milisegundos.

PerSEE: unidad de control electrónica de fusión de sensores centrales para el desarrollo de ADAS basado en percepción (Gruyer et al., 2015).

El artículo presenta un sistema de asistencia de conducción en tiempo real llamado PerSEE, el cual mediante una fusión central de sensores (radar, cámara, ultrasonido y lidar) realizaba la estimación de carriles y obstáculos en el camino. Se utilizó un algoritmo para detección y estimación de múltiples carriles basado en procesos de extracción, detección y valoración; mientras que para la visión estéreo se utilizó un estimador de espacio V-disparitive y para el sistema de rastreo se

implementó un algoritmo basado en teoría de creencias. El sistema propuesto se implementó en una placa iMX6Q con procesador ARM9Q de cuatro núcleos, con 800MHz de frecuencia. Las propiedades multi-núcleos de la placa permitieron la capacidad de distribuir los algoritmos en diferentes núcleos.

Snap-DAS: sistema de asistencia de conducción basado en visión en una plataforma embebida Snapdragon TM (Satzoda et al., 2015).

Este artículo se centra en el desarrollo de un sistema de asistencia de conducción basado en visión, implementado en un sistema embebido snapdragon 600. Aborda el tema de discusión de cómo en los últimos años estas plataformas móviles brindan una opción accesible y de procesamiento de cómputo óptimo para este tipo de aplicaciones. Los autores se centran en la cuestión de detección, análisis y desvío de carril, implementando el algoritmo *Lane analysis using selective regions* (LAsER) que fue desarrollado para su implementación embebida. Se obtuvo como resultado métricas de tiempos de ejecución aceptables así como la desviación estándar y media de evaluación de desempeño.

Sistema de monitoreo de fatiga y distracción eficaz para el conductor basado en OpenCV y Android (Manoharan and Chandrakala, 2015).

Este artículo propuso la implementación de un sistema de monitoreo de conductor en los aspectos de fatiga y distracción. Utilizaron la plataforma móvil *xiaomi redmi 1s* con 1.6GHz y 1GB de RAM, ejecutándose sobre el sistema operativo Android 4.3 *Jelly Bean*. Se implementaron detectores de clasificación de objetos para la detección del rostro y detectores de características Haar-like para la detección de los ojos. Se logró la diferenciación de estados de distracción y fatiga (parpadeo, frotamiento de ojo, detección de boca y detección de bostezo).

Detección rápida de peatones para dispositivos móviles (Costea et al., 2015).

En este artículo se presentó una solución robusta y rápida para la detección de peatones, que demostró poder ser usada en tiempo real e incluso en dispositivos móviles con limitado poder de procesamiento. Se propuso una optimización de los esquemas de detección multiescala basados en características de canal utilizando ocho modelos de detección para cada escala de media octava. Las características de

la imagen se procesaron únicamente una vez cada media octava, descartando la necesidad de una aproximación de características utilizando funciones cuadradas multiescala para entrenar los clasificadores peatonales multiresolución. El proceso de detección usando HOG y ocho modelos de clasificadores derivados de *AdaBoost* fue capaz de correr sobre 20 fotogramas por segundo en los dispositivos móviles Samsung Galaxy Tab Pro T325 (Quad-core 2.3GHz Krait 400 CPU) y Sony Xperia Z1 (Quad-core 2.2 GHz Krait 400 CPU).

Detección de peatones en tiempo real en escenarios urbanos (*Varga et al., 2014*).

En este artículo se presentó el desarrollo de un método de detección de peatones que fue probado en una computadora con Intel Core i7, 3.5 GHz, 4 núcleos, 8 procesadores lógicos y 16 GB de RAM. El método propuesto usó un enfoque de ventana deslizante con 6 dimensiones de ventana fija y una relación de aspecto constante. El algoritmo de detección sigue un formato de canal establecido con los siguientes pasos: conversión Luv, cómputo de gradiente, agrupación de histogramas y sumas rectangulares de área. El método fue desarrollado con un procedimiento de entrenamiento para la generación de modelos de clasificación, este entrenamiento fue realizado con un conjunto de datos capturado con un *smartphone*, el conjunto fue presentado en los formatos mp4 y png, con una duración de vídeo de 15 minutos (27666 imágenes). El método fue evaluado con el conjunto de datos adquirido y el INRIA, comparando la compensación de error de detección obteniendo un solapamiento mínimo del 20% en ambos conjuntos de datos, demostrando que el método es competitivo. El método presentado fue exportado para dispositivos Android e implementado en un dispositivo móvil con capacidad multi-núcleo que permitió la paralización del código. Las especificaciones del dispositivo son las siguientes: NVIDIA Tegra 3 T30L chipset, Quad-core 1.2GHz ARM Cortex-A9 CPU.

Enfoque basado en visión estéreo para detectar y rastrear carril y obstáculos delanteros en dispositivos móviles (*Petrovai et al., 2015*).

El trabajo presentó un enfoque de detección-rastreo de carril y obstáculos delanteros basado en visión estéreo implementado en dispositivos móviles. El dispositivo utilizado fue el LG Optimus V900 PAD con núcleo doble Cortex A9 Tegra 2 y cámara dual que permite la visión estéreo. El rastreo de carril y obstáculos se realizó usando como base un filtro discreto Kalman, que usa una serie de mediciones a lo largo del tiempo, para la realización de las predicciones de variables desconocidas. La detección de obstáculos fue realizada mediante la combinación del algoritmo SLIC para segmentación de super píxeles con información 3D. Se obtuvo como resultado un asistente de conducción con implementación en una aplicación Android que introdujo intervalos de visión monocular y visión estéreo con el fin de ejecutar diferentes algoritmos y lograr un sistema robusto y de alta velocidad.

Gestión de recursos de tiempo de ejecución segura y eficiente en sistemas heterogéneos para la conducción automatizada (*Borrmann et al., 2015*).

El trabajo presentado en este artículo se enfocó en presentar una nueva gestión de recursos en tiempo real para sistemas de asistencia multitarea que se implementó en la plataforma Xilinx Zynq 7000. Se utilizó una segmentación multi-escala con un enfoque piramidal, que dependiendo de la distancia a la que se esperan los objetos en diferentes regiones se discrimina de acuerdo a los diferentes tamaños y posiciones. Posteriormente con una SMV, con previo entrenamiento, se realizó la detección de luces de semáforo, bicicletas, señales de tráfico y peatones. Se presentó un mecanismo operativo de avería que proporcionó una conducción cómoda en condiciones normales, permitiendo una reacción inmediata en diferentes estados de emergencia, los cuatro posibles estados son los siguientes: estado de reacción de confort (estado de arranque), reacción de emergencia, estado de avería y estado de emergencia.

Identificación de salida de carril para asistencia avanzada de conducción (*Gaikwad and Lokhande, 2015*).

En este artículo se propuso una nueva técnica de identificación de salida de carril, la cual utilizó tres parámetros relacionados con el carril, se basó en

la transformada de distancia euclidiana para calcular la salida del carril de un vehículo. Los parámetros fueron tratados con una función de estiramiento lineal por partes (PSFL), segmentación de regiones de interés y enfoque de transformación a distancia. Se lograron resultados experimentales mejores a comparación a las métricas del estado del arte presentado en el mismo artículo, mejoras observadas en el tiempo de procesamiento de cómputo y reducción en la detección de falsos positivos en líneas de carril rectas y curvas.

Creación de sistemas avanzados de asistencia al conductor basados en visión (Nieto et al., 2014).

Este trabajo, en una primera instancia, enfatizó que debe tener un sistema de asistencia basado en visión para una implementación exitosa, mostrando una investigación de mercado sobre los asistentes de conducción. En el resto del trabajo se propone una metodología de software y hardware con un enfoque basado en histogramas orientados a gradientes, bordes y simetría para la extracción de características. Se utilizaron máquinas de soporte vectorial y *Adaboost* para la clasificación, el diagrama de bloques de la metodología puede ser observado en la Figura 3.2.

Los métodos implementados fueron para advertencia de salida de carril y detección de vehículos/peatones. Esta metodología propuesta fue implementada en diferentes plataformas para comparar sus tiempo de procesamiento, el resultado puede ser observado en la Tabla 3.2.

Tabla 3.2: Tiempo de procesamiento (por frame) de ejecución de métodos en diferentes plataformas (Nieto et al., 2014).

| Métodos | PC | PC Industrial | HW Embebido #1 | HW Embebido #2 |
|--------------------------|------|---------------|----------------|----------------|
| | (ms) | | | |
| Alerta Salida de Carril | 0.5 | 12 | 18.9 | - |
| Detección de Vehículo | 4 | 40-50 | 20-40 | - |
| Detección de Peatón | 16 | 85 | - | - |
| Detección de Somnolencia | 7 | 37 | - | 30-70 |

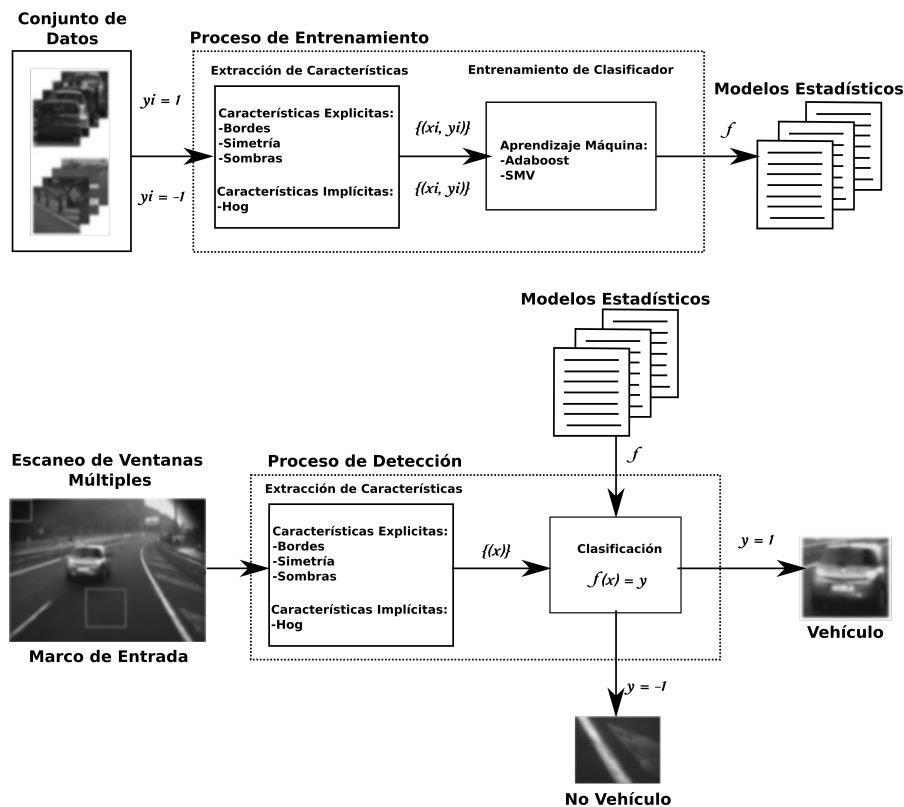


Figura 3.2: Diagrama de bloques de metodología basada en HOG, bordes, simetría y SVM (Nieto et al., 2014).

Sistema de asistencia de conducción con técnicas de detección de vehículos basadas en visión (Dai et al., 2016).

Este sistema de asistencia de conducción basado en técnicas de visión presentó un acercamiento integrado al incluir tres diferentes sistemas: “alerta de salida de carril (LDWS)”, “alerta de colisiones delanteras (FCWS)” y “detección de adelantamiento de vehículos (OVDS)”. Los autores tomaron como objetivo desarrollar sistemas, lo más simple posible, para ser capaces de hacer el cómputo en tiempo real, buscando adaptación a variedades de caminos, escenarios urbanos y condiciones de clima. El algoritmo implementado para la detección de carril es basado en características de imagen de bajo nivel definiendo la área de interés (ROI). Para detección de colisión propusieron un algoritmo de análisis de simetría para mejorar sus inconvenientes en escenarios complejos, en adición se desarrolló a un sistema de detección de vehículos usando la técnica de flujo óptico de Lucas-Kanade. Los resultados fueron obtenidos mediante pruebas experimentales implementadas en una PC con sistema operativo Windows 7, CPU Intel de 3.60

GHz. Los resultados de los sistemas implementados pueden ser observados en la Tabla 3.3.

Tabla 3.3: La exactitud y alerta de LDWS, FCWS y OVDS (Dai et al., 2016).

| Prueba de vídeo | TP | FP | FN | Precisión | Sensibilidad |
|-----------------|------|----|-----|-----------|--------------|
| LDWS | 2423 | 53 | 80 | 97.9% | 96.8% |
| FCWS | 1510 | 0 | 911 | 100% | 62.4% |
| OVDS | 40 | 0 | 2 | 100% | 95.2% |

Sistemas avanzados de asistencia al conductor embebidos basados en visión: un estudio (Velez and Otaegui, 2016).

Este trabajo brinda un estudio de diversos VB-DAS implementados en diversas plataformas embebidas. Se presentan aspectos significativos a considerar en el desarrollo de un VB-DAS, como los niveles de integridad de seguridad automotriz y cuadros por segundo necesarios para un buen desempeño (desviación estándar del tiempo de procesamiento y resolución). También se presentan métricas de detección comunes para la evaluación de desempeño de asistentes de conducción, las métricas se presentan en la Tabla 3.4.

Tabla 3.4: Definición de métricas de desempeño (Velez and Otaegui, 2016).

| Métrica de Desempeño | Definición |
|--------------------------|-----------------------------|
| Exactitud | $\frac{TP+TN}{TP+TN+FP+FN}$ |
| Recall/Tasa de Detección | $\frac{TP}{TP+FN}$ |
| Precisión | $\frac{TP}{TP+FP}$ |
| Tasa de Falsos Negativos | $\frac{FN}{FN+TP}$ |
| Tasa de Falsos Positivos | $\frac{FP}{FP+TN}$ |

También se lleva a cabo la comparación entre trabajos de VB-DAS en cuanto a resultados de desempeño, donde se mencionan los desafíos para la realización

de una comparación justa debido a la inexistencia de una estandarización en la recolección/procesamiento de datos, técnicas y sensores usados. Este trabajo brinda información relevante para el diseño, elaboración evaluación de un VB-DAS.

Aprendiendo a mapear vehículos en una vista de ojo de ave (Palazzi et al., 2017).

Este artículo propuso una solución a un paradigma de los sistemas de conducción basados en visión. El paradigma consiste en un acercamiento al entendimiento del escenario alrededor del vehículo, transformando los vehículos a través de diferentes vistas, en particular la conversión de una vista de una cámara en el tablero, a una vista de ojo de ave. La tarea que se propone se puede observar en la Figura 3.3 .

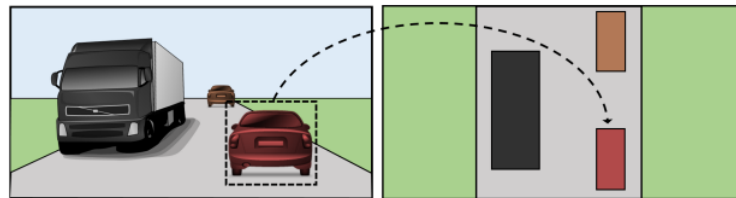


Figura 3.3: Detección de vehículos representada en una vista de ave (Palazzi et al., 2017).

Una característica interesante es que presentaron un conjunto de datos sintéticos de alta calidad recolectados de la librería Script Hook V, que permite usar funciones nativas del videojuego Grand Theft Auto V. El modelo empleado (Ver Figura 3.4) extrae representaciones profundas con la red profunda ResNet50 y transforma la vista en dos planos, permitiendo concatenar los datos para poder predecir la posición de los vehículos en la vista de ave. Los resultados experimentales se basaron en un modelo que emplea transformación proyectiva para estimar el mapeo correspondiente entre puntos en dos vistas, además de la recolección de resultados en escenarios reales con predicciones razonables.

Escenarios para el desarrollo, prueba y validación de vehículos automatizados (Menzel et al., 2018).

Este trabajo tuvo por objetivo analizar los requerimientos para la representación de escenarios en diferentes etapas de proceso definidas en la norma ISO 26262. Esta norma representa el estado del arte para un desarrollo guiado de sistemas

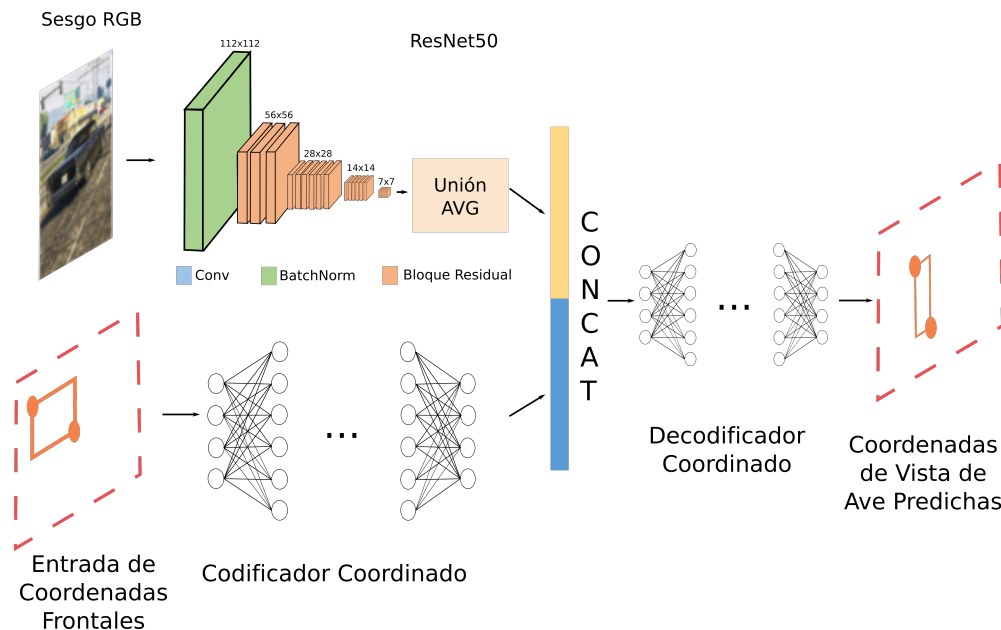


Figura 3.4: Representación gráfica de la red de proyección densa con conciencia semántica (Palazzi et al., 2017).

eléctricos/electrónicos seguros para vehículos. Proponen una terminología para la identificación de diferentes niveles de abstracción para la demostración de escenarios que pueden ser sistemáticamente evolucionados a lo largo de las fases del proceso de desarrollo descrito en la norma. Los tres niveles de abstracción propuestos son: “escenarios funcionales” que operan en un nivel semántico que incluye una descripción lingüística y consistente de relaciones y entidades, “escenarios logísticos” que representan un nivel de estado espacial entre las entidades al ser representadas en una notación formal y “escenarios concretos” donde se representaron relaciones y entidades con valores concretos para diferentes parámetros.

3.2.1. Discusión del Estado del Arte

En los trabajos y artículos mencionados anteriormente se trataron diversos temas de interés referentes a los sistemas asistentes de conducción: la adquisición de información, algoritmos implementados y plataformas utilizadas. Una breve discusión de estos temas se muestra a continuación:

- Adquisición de la información.

En la mayor parte de los trabajos revisados se hizo uso de diversos tipos

cámaras para la adquisición de información, obteniendo secuencias de vídeo del camino y/o monitoreo del conductor, estos dispositivos demuestran la ventaja de poder extraer información de diferentes áreas de interés mediante la implementación de diversas técnicas y algoritmos de visión por computadora según las necesidades requeridas. También en algunas aplicaciones se utilizó adicionalmente información de sensores del tipo ultrasónicos y *LIDAR* para complementar algunas tareas de detección.

- Algoritmos implementados.

En lo que refiere a algoritmos de visión predominan el uso de clasificadores basados en características Haar y en histogramas de gradientes orientados, utilizados mayormente por su desempeño en tiempo real. En la parte de inferencia se predomina el uso de control difuso tipo-1, redes neuronales y máquinas de soporte vectorial para la detección de diversas situaciones correspondientes a la tarea de un VB-DAS.

- Plataformas de implementación.

En la mayor parte de aplicaciones robustas demostradas se implementó en equipos de cómputo no embebidos (Laptops, computadora de escritorio, etc.), que demostraron un buen desempeño en tiempos y exactitud de detección, mientras que en las plataformas embebidas se busco el uso de algoritmos simples que tuviesen una respuesta óptima en tiempo real para una implementación versátil. En algunas implementaciones se tuvo un enfoque utilizando plataformas embebidas heterogéneas que demostraron tener las ventajas de ser versátiles por su enfoque dedicado y potente computacionalmente, debido al uso de un CPU para tareas secuenciales y una GPU para realización de tareas en paralelo.

Una comparación más descriptiva que incluye información relevante de cada trabajo de los artículos se muestra en las Tabla 3.5.

Tabla 3.5: Resumen de artículos.

| Trabajo | Objetivo | Algoritmos/Métodos | Aportación |
|---|---|---|--|
| Sistema de asistencia al conductor basado en reglas difusas (Chien et al., 2015). | Proponer un VB-DAS que pueda emitir alertas cuando el vehículo esté en situaciones inseguras utilizando un conjunto de módulos basados en reglas difusas. | Cost-Sensitive AdaBoost | Mejora de 4.1% de identificación de vehículo con CS AdaBoost en comparación a AdaBoost, métricas de comparación con lógica difusa tipo-1 y redes neuronales artificiales de 9 nodos. |
| Sistema integrado de asistencia al conductor con múltiples sensores para una maniobra segura de adelantamiento (Chiang et al., 2014). | Desarrollo de un sistema de asistencia en plataforma embebida de bajo costo basado en información de múltiples sensores y empleando control difuso. | Algoritmos de privados de las empresas Mazda y Honda, fusión de datos de sensores, control lógico difuso. | Procesador de señal digital capaz de realizar maniobras de conducción como mantenimiento de carril, cambio de carril y adelantamiento con lógica difusa tipo-1. |
| Optimización de sistema de asistencia basado en visión por computadora: estudio de caso de detección de vehículos (Nieto et al., 2016). | Diseño y desarrollo de optimización de algoritmo de rendimiento y optimización del ciclo de desarrollo. | HOG, SVM y clasificadores AdaBoost | Presentación de cuatro prototipos de metodologías de detección de vehículo: multiescala, perspectiva multiescala, perspectiva pre-detección multiescala y cambio de día/noche. |

Resumen de artículos (Continuación).

| Trabajo | Objetivo | Algoritmos/Métodos | Aportación |
|---|--|---|--|
| Detección de vehículos a escala múltiple basada en visión y rastreo de mezcla de distancia relevante para el sistema de asistencia al conductor (<i>Gu et al., 2015</i>). | Mejora de robustez en detección y rastreo de vehículos a escala múltiple basados en visión. | Extracción de características <i>Haar</i> y detección de vehículos basado en AdaBoost (mapeo de perspectiva inversa). SVM con características HOG para detección robusta. | Método de sistema de coordenadas lineales para detección de vehículos con base en matriz y esqueleto del vehículo. |
| Diseño e implementación de cámara vehicular en tiempo real para asistencia al conductor y estimación de congestión de tránsito (<i>Son and Baek, 2015</i>). | Diseño e implementación de un sistema de información de tráfico en tiempo real y un dispositivo de cámara inteligente para sistema de asistencia . | HOG, SVMs. Algoritmo Dilate, Transformada de Hough, salto de cuadro dinámico (DFS) y región de interés dinámica(D-ROI). | Métodos D-ROI y DFS en procesamiento en paralelo y estimación de congestión vehicular implementado en NVIDIA <i>Jetson</i> Tegra Kepler1. |
| Detección de señales de tráfico basadas en FPGA para sistemas de cámaras automotrices (<i>Schwiegelshohm et al., 2015</i>). | Introducir un sistema de procesamiento de imagen RGB para detección de señales de tráfico | Algoritmo de desborde y clasificación espacial de color RGB. | Introducción de un sistema de asistencia enfocado en la detección de señales de tráfico en Spartan-6-FPGA utilizando espacio de color RGB. |

Resumen de artículos (Continuación).

| Trabajo | Objetivo | Algoritmos/Métodos | Aportación |
|---|---|---|---|
| Implementación y Evaluación del Algoritmo de Reconocimiento de Imagen para un Vehículo Inteligente utilizando SoC Heterogéneo de Varios Núcleos (Ozaki et al., 2015). | Desarrollo de un sistema en chip multi-núcleo especializado en reconocimiento de imágenes para advertencia de colisión frontal y Advertencia de salida de carril. | Co-correncia de histograma de gradientes orientados (CoHOG) | Metodología de un sistema en chip multi-núcleo heterogéneo para el reconocimiento de imagen utilizando 4 aceleradores y procesos en paralelo. |
| PerSEE: una unidad de control electrónica de fusión de sensores centrales para el desarrollo de un sistema de asistencia basado en percepción (Gruyer et al., 2015). | Propuesta del sistema en tiempo real PerSEE para el desarrollo de una fusión central de sensores de una unidad de control electrónica. | Algoritmo para detección y estimación de múltiples carriles basado en un proceso de tres pasos: extracción, detección y estimación. | Fusión de sensores en una plataforma iMX6Q con ARM9Q con procesamiento en paralelo. |
| Snap-DAS: un Sistema de Asistencia de Conducción Basado en Visión en una Plataforma Embebida Snapdragon TM (Satzoda et al., 2015). | Desarrollar un sistema de asistencia basado en visión en una plataforma Snapdragon centrándose en análisis de carril. | Análisis de carril usando regiones selectivas (LASER) | Metodología de implementación del algoritmo LASER en una plataforma embebida Snapdragon. |

Resumen de artículos (Continuación).

| Trabajo | Objetivo | Algoritmos/Métodos | Aportación |
|--|---|--|--|
| Sistema de monitoreo de fatiga y distracción eficaz para conductor basado en OpenCV de Android (<i>Manoharan and Chandrakala, 2015</i>). | Lograr un acercamiento novedoso a un sistema de monitoreo del conductor con detección de rasgos faciales y oculares implementado en un dispositivo Android. | Clasificación en cascada <i>Haar, AdaBoost</i> . | Metodología de implementación del sistema en un dispositivo <i>Xiaomi Redmi 1s</i> con monitoreo basado en <i>AdaBoost</i> y <i>Haar</i> . |
| Detección rápida de peatones para dispositivos móviles (<i>Costea et al., 2015</i>). | Detección de peatones en dispositivos móviles a 20 FPS en tiempo real. | HOG, clasificador de características integrales de canal y 8 modelos de clasificadores usando <i>AdaBoost</i> . | Nuevo esquema de detección de peatones basado en 8 modelos multirresolución dirigido a dispositivos móviles. |
| Detección de peatones en tiempo real en escenarios urbanos (<i>Varga et al., 2014</i>). | Desarrollo de método de detección de peatones con un enfoque de ventana deslizante con 6 alturas de ventana fija y una relación de aspecto constante. | Algoritmo de detección de peatones con: conversión Luv, computo de gradiente, agrupación de histogramas y sumas rectangulares de región. | Implementación en tiempo real del método en PC de un algoritmo de detección de peatones, posibilidad de exportación a un dispositivo móvil Android |

Resumen de artículos (Continuación).

| Trabajo | Objetivo | Algoritmos/Métodos | Aportación |
|--|--|--|--|
| Enfoque basado en visión estéreo para detectar y rastrear carril y obstáculos delanteros en dispositivos móviles (<i>Petrovai et al., 2015</i>). | Propuesta de método para el seguimiento de carriles y obstáculos basado en la visión estéreo limitada para dispositivos móviles. | Filtro Kalman y SLIC para segmentación de super píxeles con información 3D. | Metodología de despliegue del sistema en LG Optimus V900 con doble núcleo Cortex A9 Tegra 2 y cámara dual para modelos 2D/3D. |
| Gestión de recursos de tiempo de ejecución segura y eficiente en sistemas heterogéneos para la conducción automatizada (<i>Borrmann et al., 2015</i>). | Propuesta de gestión de recursos en tiempo de ejecución real para sistemas de asistencia multitarea heterogéneos. | Segmentación multi-escala usando enfoque piramidal y SMV para entrenamiento de clasificadores. | Optimización en la ejecución de múltiples aplicaciones en una plataforma heterogénea Xilinx Zynq SoC e implementación de un mecanismo operativo de avería. |
| Identificación de salida de carril para asistencia avanzada de conducción (<i>Gaikwad and Lokhande, 2015</i>). | Proponer una técnica para la identificación de la salida de carril no deseada de un vehículo que viaja en una carretera. | Transformada de Hough para estimamiento lineal por partes (PSFL), segmentación de regiones de interés y enfoque de transformación a distancia. | Métricas de detección de carril (Líneas curvas y rectas). |

Resumen de artículos (Continuación).

| Trabajo | Objetivo | Algoritmos/Métodos | Aportación |
|--|---|--|--|
| En la creación de sistemas avanzados de asistencia al conductor basados en la visión (Nieto et al., 2014). | Tratar temas relevantes para la implementación exitosa de un VS-DAS, incluyendo una propuesta de metodología de <i>Software</i> y <i>Hardware</i> . | Uso de SVM como clasificador y <i>AdaBoost</i> para extracción de características. | Implementación y comparación métricas en diferentes plataformas como PC, PC industrial, y dos plataformas MAC. |
| Un sistema de asistencia de manejo con técnicas de detección de vehículos basados en la visión (Dai et al., 2016). | Desarrollo integral para detección de salida de carril simple e implementación un sistema de alerta de colisión y detección. | Técnica de flujo óptico Lucas-Kanade, muestra aleatoria de consenso (RANSAC) y etiquetado de componentes conectados (CCL). | Métricas para detección de salida de carril, sistema de alerta de colisión y sistema de detección de vehículo. |
| Sistemas Avanzados de Asistencia de Conducción Basados en Visión Embebidos: a survey (Velez and Otaegui, 2016). | Mostrar el campo de los sistemas de asistencia basados en visión embebidos, revisando las diferentes opciones de HW y SW, consideraciones de diseño, desarrollo y evaluación. | NA | Métricas de 34 sistemas de asistencia de conducción embebidos en diferentes plataformas: desempeño(exactitud, FP, TP, FN y Precisión). frames/s, opciones de <i>Hardware</i> y <i>Software</i> . |

Resumen de artículos (Continuación).

| Trabajo | Objetivo | Algoritmos/Métodos | Aportación |
|---|--|--|---|
| Aprendiendo a mapear vehículos en una vista de ojo de ave (<i>Palazzi et al., 2017</i>). | Desarrollo de un modelo para mapear vehículos a vista de ave. | Red Profunda ResNet50. | Un conjunto de datos de alta calidad sintético y un modelo neuronal basado en aprendizaje profundo para mapear escenas en una vista de ave. |
| Escenarios para el desarrollo, prueba y validación de vehículos automatizados (<i>Menzel et al., 2018</i>). | Discutir requerimientos de representación de escenarios en diferentes etapas del desarrollo de sistemas de seguridad electrónicos para vehículos definido por la ISO26262. | Proceso de desarrollo <i>V-model-based</i> . | Propuesta de de tres niveles de abstracción de escenarios para satisfacer requerimientos de la ISO 26262. |

Capítulo 4

Metodología

4.1. Sistema Propuesto

Como se mencionó en la introducción, el sistema se compone de 3 módulos esenciales: adquisición de información basada en técnicas de Visión por Computadora, inferencia basada en Lógica Difusa Tipo-2 y la implementación en tarjeta *Jetson* TX2. En este capítulo se presentan los algoritmos estudiados e implementados en cada módulos para cumplir lo establecido en el objetivo general y en los objetivos específicos.

4.1.1. Adquisición de la Información

Para la tarea de monitoreo del conductor se utilizó la estimación de puntos de referencia facial y relaciones de aspecto para la detección de eventos de interés, como la detección de bostezo, parpadeo y posición de cabeza. Mientras que para la detección de vehículos en la parte frontal del camino se basó en la Detección de Objeto de Margen Máximo (MMOD), este enfoque de detección funciona mediante una red neuronal convolucional que sustituye la supresión no máxima en su proceso de entrenamiento por una función de margen máximo, esto con el fin de simplificar el entrenamiento y obtener una detección más robusta en entornos no controlados.

Estimación de puntos de referencia

Para el monitoreo del conductor se basó en el trabajo de estimación de puntos de referencia introducida en el trabajo de (*Kazemi and Sullivan, 2014*), para la obtención de zonas o puntos de interés dentro de una imagen. Consiste en una detección

previa del rostro, usualmente *Haar* (Viola and Jones, 2001) o HOG (Dalal and Triggs, 2005), seguido de un predictor entrenado con un ensamble en cascada de árboles de regresión para realizar la estimación de puntos de referencia de un vector S que contiene 68 puntos. El predictor consiste en una cascada que en cada nivel T una serie de regresores débiles, denotados por $S = (x_1^T, x_2^T, \dots, x_p^T)^T$, representan las coordenadas los 68 puntos de referencia faciales p en una imagen I , ver Figura 4.1.

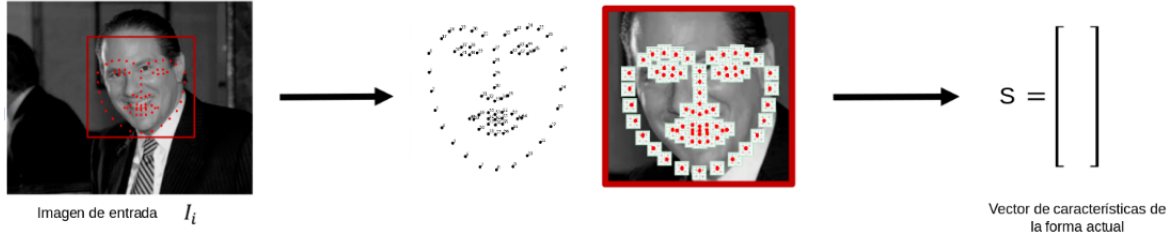


Figura 4.1: Coordenadas de los puntos de referencia facial p contenidas en el vector S (Kazemi and Sullivan, 2014).

Se usa $\hat{S}^{(t)}$ para indicar la estimación actual de S , cada regresor r_t en la cascada predice un vector de actualización de la imagen y $\hat{S}^{(t)}$ se agrega a la estimación de forma actual $\hat{S}^{(t)}$ para mejorar la estimación:

$$\hat{S}^{(t+1)} = \hat{S}^{(t)} + r_t(I, \hat{S}^{(t)}) \quad (4.1)$$

En la cascada el regresor r_t hace sus predicciones basadas en características, como los valores de intensidad de píxel, calculados desde I e indexados en relación con la estimación de forma actual $\hat{S}^{(t)}$. Para el entrenamiento del predictor se tienen los datos $(I_1, S_1), \dots, (I_n, S_n)$ donde cada I_i es una imagen de la cara y S_i es un vector de forma correspondiente a la imagen. Para la primera función de regresión r_0 en la cascada se crea la tripleta de datos de entrenamiento que consisten en una imagen del rostro, una estimación de forma inicial y el paso de actualización objetivo, es decir, $(S_{\pi i}, \hat{S}_i^0, \Delta S_i^0)$, ver Figura 4.2.

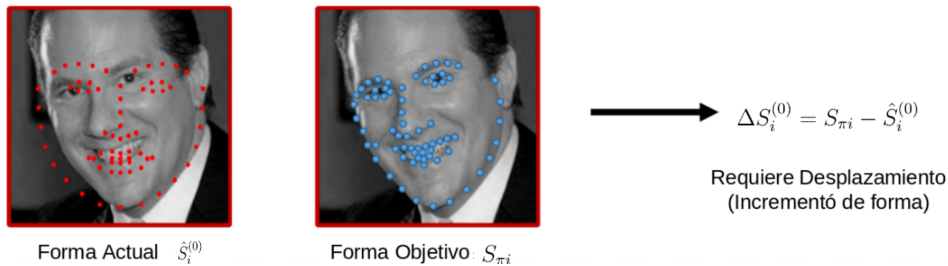


Figura 4.2: Incremento de forma (Kazemi and Sullivan, 2014).

para $i = 1, \dots, N$. Se establece el número total de esta tripleta en $N = n_R$ donde R es el número de inicializaciones utilizadas por imagen I_i . A partir de estos datos, se obtiene la función de regresión r_0 . La tripleta de entrenamiento se actualiza para proporcionar los datos de entrenamiento, $(S_{\pi i}, \hat{S}_i^{(1)}, \Delta S_i^{(1)})$, para el siguiente regresor r_1 en la cascada estableciendo (con $t = 0$):

$$\hat{S}_i^{(t+1)} = \hat{S}_i^{(t)} + r_t(I_{\pi}, \hat{S}_i^{(t)}) \quad (4.2)$$

$$\Delta S_i^{(t+1)} = S_{\pi i} - \hat{S}_i^{(0)} \quad (4.3)$$

Este proceso se repite hasta que se obtiene una cascada de T regresores r_0, r_1, \dots, r_{T-1} que combinados dan un nivel suficiente de precisión, esperando una cascada de regresión con un proceso como el de la Figura 4.3.

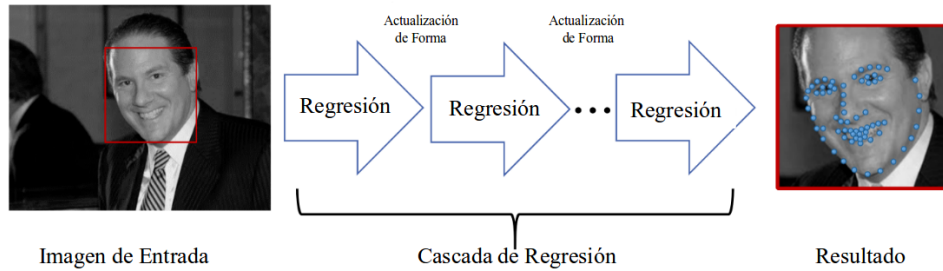


Figura 4.3: Proceso de regresión, (Kazemi and Sullivan, 2014).

Detección de eventos de Inatención

Teniendo como base los puntos de referencia que detectan las características faciales, se entrenó un predictor de 22 puntos de referencia faciales, vector de la Figura 4.4b. Se utilizó un cuarto (5321 imágenes con 68 anotaciones de marcas de referencia faciales) del repertorio IBUG (Sagonas et al., 2013) que se conforma de los datos de otras bases de datos como: LFPW (Belhumeur et al., 2013), FWL (Yang et al., 2016), HELEN (Le et al., 2012). El predictor fue entrenado con la ayuda de la librería Dlib (King, 2009), de la cual también se utilizó un detector robusto basado en 5 detectores HOG de vista frontal, izquierda, derecha, frontal girada a la izquierda y frontal girada a la derecha. El predictor resultante obtuvo un error medio de 8.40 respecto a la distancia promedio de los puntos de referencia predichos y donde deberían estar de acuerdo a el *Ground – truth*, siendo considerado como bueno con un error por debajo de 10.

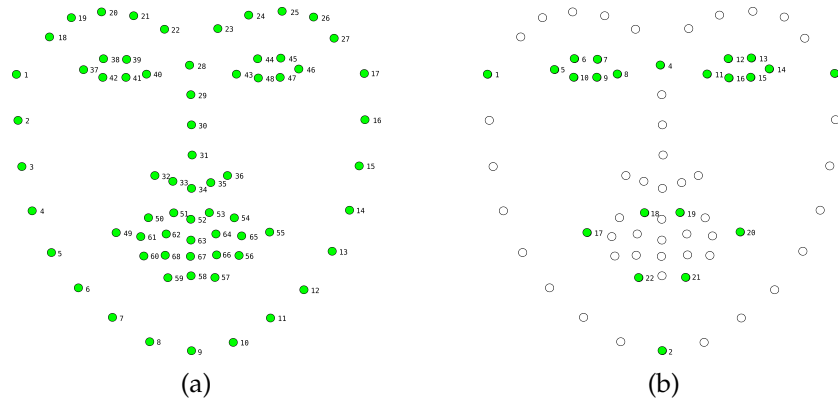


Figura 4.4: Predicción de puntos: (a) 68 puntos, (b) 22 puntos.

Una forma de encontrar eventos de interés en la imagen es utilizando la relación de aspecto del ojo (*EAR*) de la ecuación 4.4, introducida en Soukupova and Cech (2016).

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||} \quad (4.4)$$

Donde $p1, \dots, p6$ son coordenadas x, y en la imagen, el orden de estos puntos de muestra en la Figura 4.5a. En el numerador se calcula la distancia entre los puntos de referencia vertical, mientras que el denominador calcula la distancia entre los puntos de referencia horizontal. La relación de aspecto es aproximadamente constante mientras el ojo está abierto, pero decae rápidamente a cero cuando se produce un parpadeo, permitiendo evitar técnicas de procesamiento complejas y simplemente interpreta *EAR* para determinar si una persona parpadea.

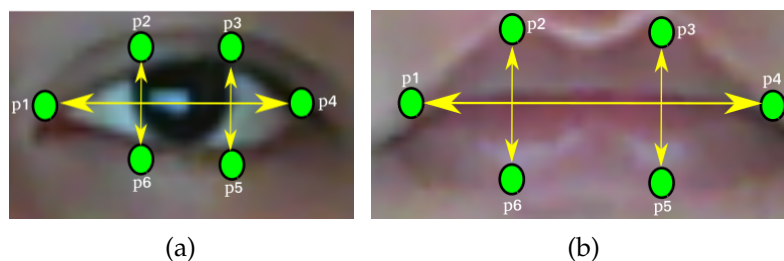


Figura 4.5: Posición de los puntos de referencia para el cálculo de *EAR* y *MAR*.

Se adaptó el mismo enfoque para poder calcular la relación de aspecto de la boca (*MAR*) mediante los puntos de la Figura 4.5b y aplicando la ecuación 4.5. Siendo que en esta ocasión las relaciones son constantes mientras la boca está

cerrada o realiza gestos menores, pero incrementa rápidamente a un valor cercano a uno cuando se detecta un bostezo.

$$MAR = \frac{||p2 - p6|| + ||p3 - p5||}{||p1 - p4||} \quad (4.5)$$

La posición de cabeza se calculó mediante las distancias de las coordenadas de la figura 4.4b. Siendo al punto central 4 de donde se obtienen las diferencias de distancia respecto a los puntos 1 (CI = Centro-Izquierda), 3 (CD = Centro-Derecha) y 2 (CA = Centro-Abajo).

Detección de Objeto de Margen Máximo (MMOD)

Para la detección de vehículos en la parte frontal del camino se utilizó el trabajo de (King, 2015) que mejora la detección de objetos al reemplazar la supresión no máxima con una función objetivo de margen máximo en una red neuronal convolucional seguida de un filtro lineal.

Esencialmente, los detectores de ventanas deslizantes escanean todas las ubicaciones de imágenes y buscan un objeto en específico, la parte trasera de un automóvil en este caso. Si existe un automóvil en una ubicación específica de una imagen se producen varias ubicaciones de ventanas deslizantes ligeramente diferentes, por lo que es común que los detectores incluyan un enfoque de supresión no máxima que toma la ventana con la detección más fuerte y elimina las demás. Sin embargo, para definir la cercanía entre las ventanas en una red neuronal convolucional se necesita aprender a generar pequeños puntajes de detección para pequeñas áreas de la imagen que no se supriman. Cuanto más pequeña es la región de supresión no máxima, más tiene que aprender la red neuronal convolucional y más difícil se hace el entrenamiento. Para lidiar con este problema se introduce el concepto de margen máximo como función de pérdida en la red neuronal convolucional, donde se utilizan todas las sub-ventanas como parte del entrenamiento, que brinda una robustez que permite hacer detecciones más confiables. La red neuronal convolucional consiste en una arquitectura que cuenta con dos capas convolucionales, una con un filtro 5x5 y la otra de 3x3, se realiza una disminución de ocho veces mediante el uso de estas capas simultáneamente para cada uno de los tres canales de la imagen. Con estos, se obtiene un mapa de características de 32 dimensiones. Estas características pasan por 4 capas convolucionales de 3x3. La última capa tiene un solo canal y los valores de este son

grandes cuando la red encuentra un objeto en una ubicación particular. El gráfico de la red se muestra en la Figura 4.6.

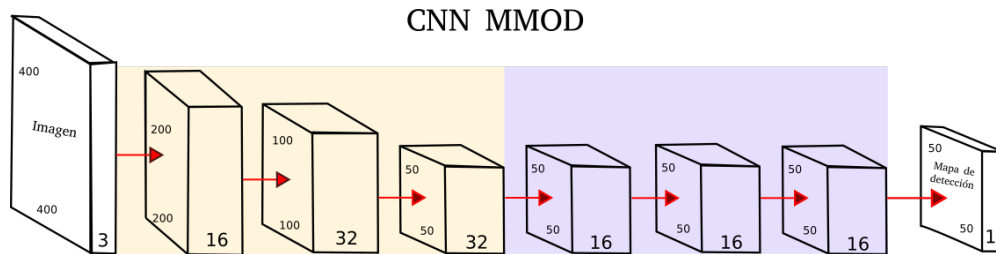


Figura 4.6: Arquitectura de CNN MMOD.

El modelo resultante está entrenado con 2217 imágenes del conjunto de datos *lib_rear_end_vehicles* de la base de datos *Caraffi et al. (2012)*. En este modelo se utiliza un tamaño de lote de 150 y tasa de aprendizaje inicial de 0.1 que al llegar a un valor inferior a 0.0001 detiene el entrenamiento. Los pasos de procesamiento de alto nivel del detector funcionan de la siguiente forma:

1. Se crea una imagen piramidal y se comprime en una imagen más grande.
2. Se ejecuta la imagen de la pirámide en mosaico a través de la CNN. La CNN emite una nueva imagen donde los píxeles brillantes en la imagen de salida indican la presencia de automóviles.
3. Busca píxeles en la imagen de salida de CNN con un valor > 0 . Esas ubicaciones son las detecciones preliminares de automóviles.
4. Realiza la supresión no máxima en las detecciones preliminares para producir la salida final.

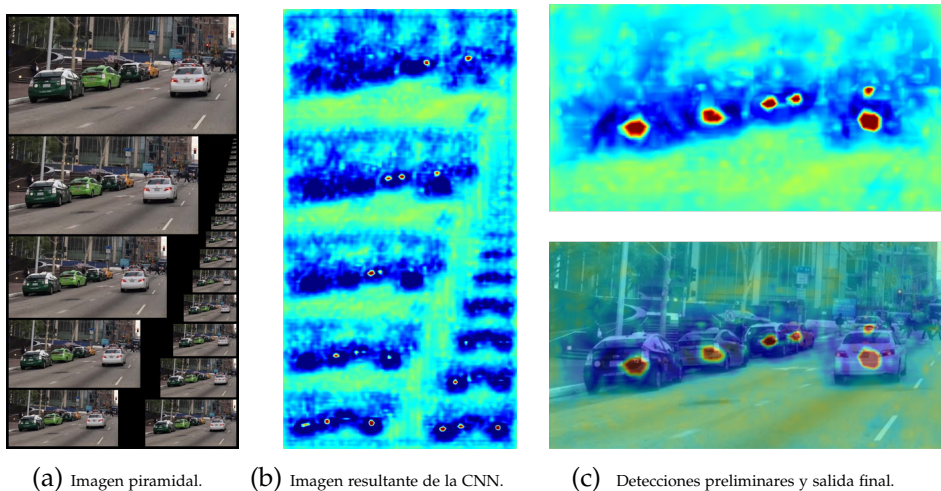


Figura 4.7: Proceso de detección de la CNN MMOD.

En el proceso de detección final se tomó el área producida por el cuadro delimitador como valor de entrada *DisCars*, este valor representa la cercanía de un vehículo frontal, siendo que entre más cercano el valor a 1 significa una mayor cercanía de un vehículo frontal y en caso contrario entre más cerca a 0 significa que se detectan vehículos lejos. Los valores para la clasificación de *DisCars* se definió mediante una serie de aproximaciones realizadas en fotogramas de prueba de la cámara utilizada para el asistente. El cuadro de dimensiones 640x480 es recortado y re-dimensionado a 200x150 para la obtención de la sección de interés, esto con el objetivo de reducir el área de búsqueda y realizar un procesamiento más rápido. En caso de realizar dos o más detecciones se considera la más cercana. El proceso de detección implementado se observa en la Figura 4.8.



Figura 4.8: Prueba de recorte y detección con CNN MMOD.

4.1.2. Inferencia

En este módulo se realiza la inferencia mediante un controlador difuso tipo-2, elegido por su característica de trabajar con situaciones de incertidumbre, trasladar la experiencia humana a través de su base de reglas y ser un sistema de múltiples entradas que toma los valores de las relaciones de aspecto *EAR*, *MAR*, *CI*, *CD*, *CA* y el valor aproximado de la detección *DisCars*. El controlador se realizó con la librería PyIT2FLS de (Haghras and Ghaemi, 2019) y se describe a continuación.

Sistemas de control difusos

La implementación práctica del concepto de Lógica Difusa Tipo-2 se encuentra en los sistemas difusos. Estos son utilizados para el control en situaciones de incertidumbre teniendo la característica de permitir trasladar la experiencia humana mediante una serie de reglas y trabajar con múltiples entradas y salidas. Estos

sistemas reciben su nombre con base a los conjuntos utilizados, los sistemas lógicos difusos tipo-1 hacen uso de conjuntos tipo-1 (Figura 4.9a), los intervaluados lógicos tipo-2 conjuntos de intervalo tipo-2 (Figura 4.9b) y los generales tipo-2 de conjuntos generales tipo-2 (Figura 4.9c).

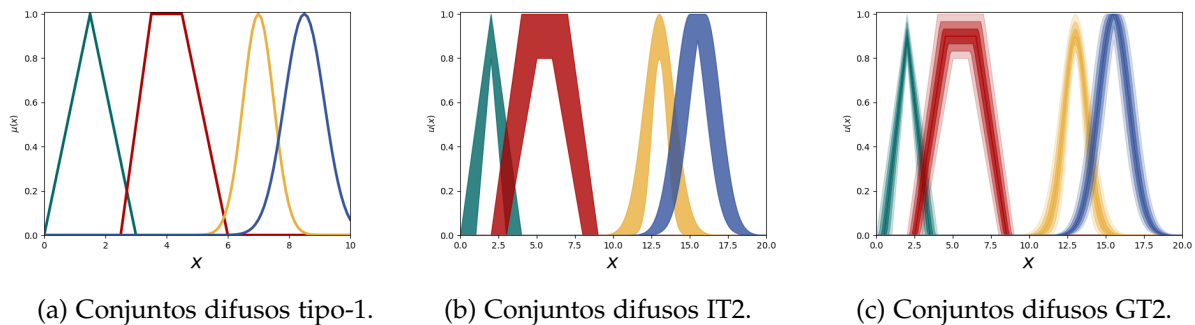


Figura 4.9: Tipos de conjuntos difusos.

Para el desarrollo de este trabajo de tesis se optó por el uso del Sistema Lógico Difuso de Intervalo Tipo-2, justificando su uso y detallando su diseño en la siguiente subsección.

Sistemas Lógicos Difusos de Intervalo Tipo-2

Planteado por (Gorzalczany, 1987) y retomados por (Liang and Mendel, 2000) como una solución para lograr una mayor velocidad de cómputo en los sistemas difusos tipo-2, permiten una aplicación de tareas en tiempo real y/o de gran complejidad ya que reducen drásticamente los requerimientos de computación, siendo este el motivo de ser seleccionado para el desarrollo del sistema de asistencia propuesto. El tiempo de ejecución se reduce, pero por otra parte se pierde el potencial que puede obtenerse de la unión con otro sistema difuso, a pesar de esto se ha demostrado que en varias aplicaciones los IT2FLSs tienen un mejor desempeño que el T1FS ante el ruido y situaciones no controladas, como se explica en (Jerry, 2019). Los IT2FLSs se componen principalmente por 5 elementos principales: Fuzzificación, Inferencia, Base de conocimiento, Reducción de tipo y Defuzzificación. En la Figura 4.10 puede observarse cómo se relacionan los elementos del IT2FS y en el resto de esta sección se explica la configuración de elementos de forma detallada para el diseño del sistema de asistencia.

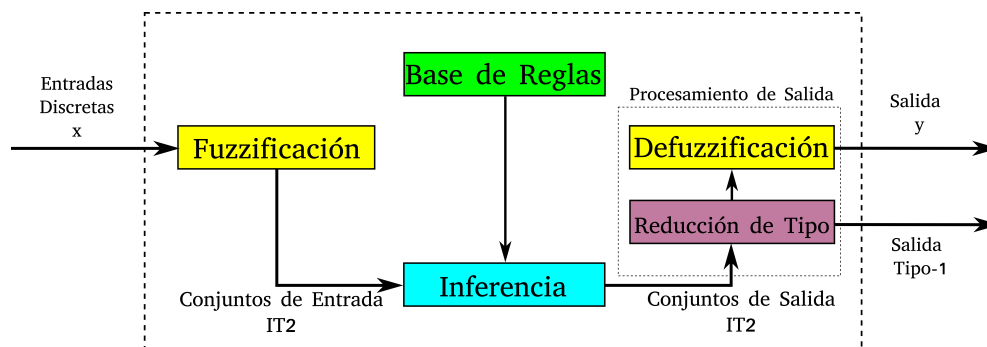


Figura 4.10: Diagrama de un IT2FLS.

Fuzzificación

Para los IT2FLSs existen tres tipos de fuzzificadores posibles: *Singleton*, *No-Singleton* tipo-1 y *No-Singleton* IT2 (ver Figura 4.11). Para este caso se utilizó una fuzzificación *No-Singleton* tipo-2 que transforma la medición $x_i = x'_i$ en un número difuso tipo-2, es decir es un IT2FS cuyas MFs superior e inferior de la huella de incertidumbre es un número difuso tipo-1.

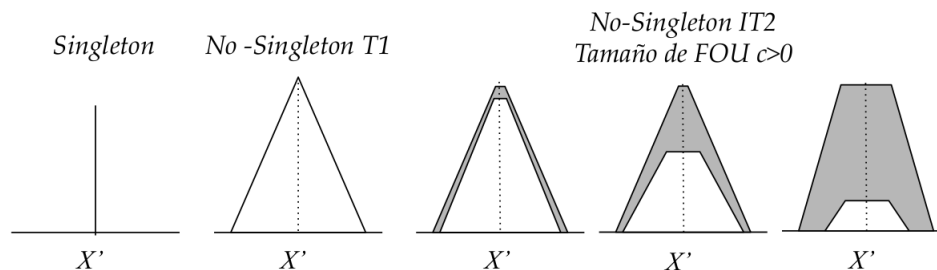


Figura 4.11: Tipos de fuzzificación.

Se optó por usar una fuzzificación con tamaño de FOU variable, dependiendo del impacto e importancia deseado para los conjuntos, se eligió usar MFs de tipo gaussiano que demuestran manejar de mejor forma comportamientos de la vida real y en situaciones no controladas, tal como se aprecia en los trabajos de (Yao *et al.*, 2012) y (Yao *et al.*, 2016). Para los conjuntos de las entradas *CI*, *CD* y *CA* se tomó en cuenta que pueden ser variables, dependiendo de la forma de la cara y la posición de esta respecto a la cámara, por ello al ejecutar el sistema se toma el primer cuadro del conductor y se obtiene valores iniciales con los que se definen los conjuntos difusos que se adaptarán en cada ejecución del sistema; para esto se considera que el conductor está en una posición inicial óptima (centrada al camino). Las entradas de *EAR* Y *MAR* cuentan con la característica que, al detectar un parpadeo,

decae rápidamente a un valor aproximado a cero y en caso de bostezo aumenta aproximadamente a un valor de 1 por lo cual no requieren ajuste, mientras que la entrada *DisCars* se maneja con intervalos de aproximación definidos distancia cerca, media y lejana. Las MFs definidas para las diferentes entradas pueden ser apreciados en la Figura 4.12.

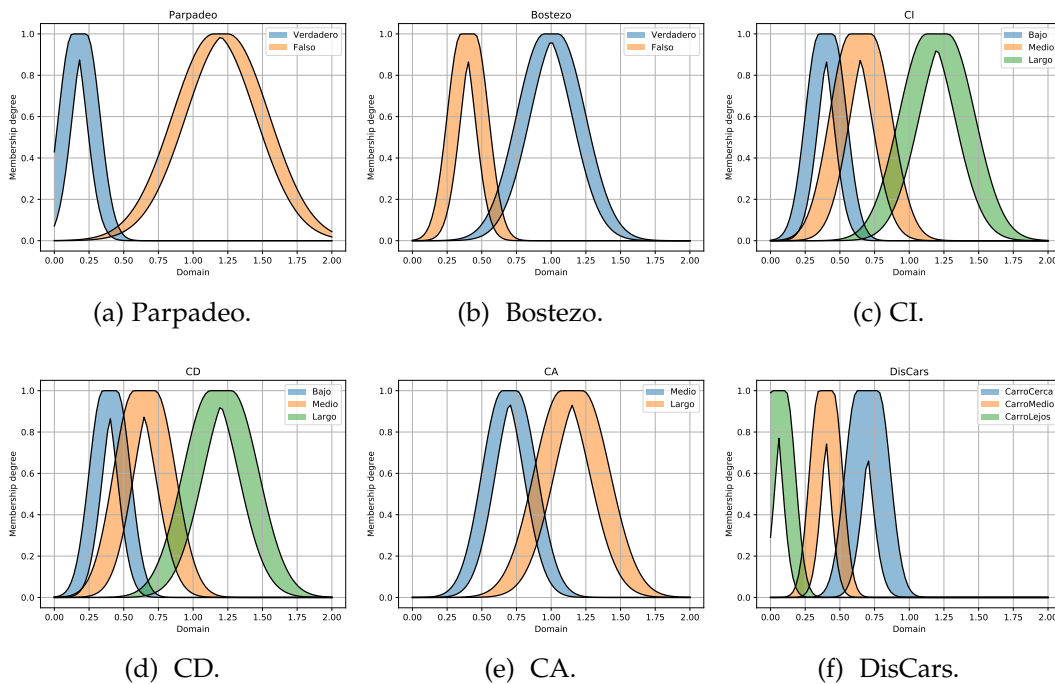


Figura 4.12: Conjuntos difusos definidos para las diferentes entradas.

Para la salida nivel de riesgo se definieron 4 MFs, entre los que se encuentran el nivel bajo, medio, alto y somnoliento, siendo este último el de mayor valor e impacto del IT2FLS, las MFs pueden observarse en la Figura 4.13.

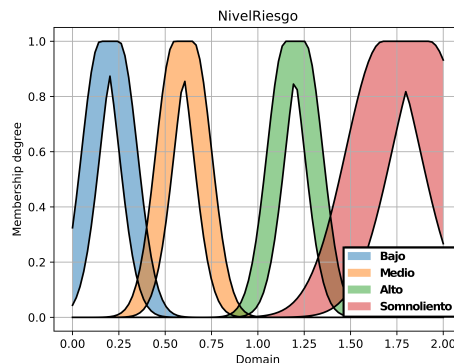


Figura 4.13: Conjuntos difusos definidos para la salida.

Base de Conocimiento

En esta parte residen las reglas canónicas que permiten la transferencia de experiencia al sistema, existen las mismas dos estructuras canónicas (Mamdani y TSK) para los sistemas difusos tipo-1 y tipo-2. La diferencia entre tipo-1 y tipo-2 es asociada con la naturaleza de las MF, que no es de tanta relevancia cuando se crean las reglas, ya que sólo la diferencia es que se tienen conjuntos difusos tipo-2 involucrados. Se definieron 13 reglas de tipo Mamdani para este trabajo y son las que se muestran a continuación:

R^1 : SI EAR ES $ParpadeoV$ Y $tempoE$ ES $TempoV$ ENTONCES y ES $NivelS$

R^2 : SI MAR ES $BostezoV$ Y $tempoM$ ES $TempoV$ ENTONCES y ES $NivelS$

R^3 : SI CD ES $PCDA$ Y MAR ES $BostezoV$ ENTONCES y ES $NivelA$

R^4 : SI CI ES $PCIA$ Y MAR ES $BostezoV$ ENTONCES y ES $NivelA$

R^5 : SI CI ES $PCIA$ Y $DisCars$ ES $CarC$ ENTONCES y ES $NivelA$

R^6 : SI CD ES $PCDA$ Y $DisCars$ ES $CarC$ ENTONCES y ES $NivelA$

R^7 : SI CI ES $PCIB$ Y CD ES $PCDA$ ENTONCES y ES $NivelA$

R^8 : SI CI ES $PCIA$ Y CD ES $PCDB$ ENTONCES y ES $NivelA$

R^9 : SI CA ES $PCAB$ ENTONCES y ES $NivelA$

R^{10} : SI $DisCars$ ES $CarC$ ENTONCES y ES $NivelA$

R^{11} : SI $DisCars$ ES $CarM$ ENTONCES y ES $NivelM$

R^{12} : SI CD ES $PCDM$ ENTONCES y ES $NivelB$

R^{13} : SI CI ES $PCIM$ ENTONCES y ES $NivelB$

R^1 y R^2 se definieron en sentido de detectar dos de los síntomas más característicos de somnolencia, en este caso un parpadeo y bostezo prolongado. R^3 y R^4 para detectar situaciones en que el conductor esté girado y realizando un comportamiento de bostezo y aproximado a una plática. R^5 y R^6 se definieron para situaciones donde el conductor esté girando y un vehículo frontal esté demasiado cerca. R^7 y R^8 para detectar si la posición de la cabeza está demasiado a la izquierda o la derecha. R^9 para detectar si el conductor tiene la posición de la cabeza hacia abajo. R^{10} y R^{11} para detectar una distancia media y demasiado cerca de un auto frontal. Por último, R^{12} y R^{13} para la detección de un nivel bajo debido a que el conductor tiene la posición de la cabeza centrada al camino. Siendo que se utiliza un operador de implicación Mamdani, el sistema difuso usa las reglas Zadeh y se denomina como un **sistema difuso Mamdani IT2**. Se eligieron 13 reglas con el fin de tener un mejor desempeño, tomando en cuenta que las situaciones que no activan alguna regla se toman como una situación "Segura" por defecto.

Inferencia

En un T1FS, el motor de inferencia transforma la entrada tipo-1 en una salida tipo-1. Múltiples antecedentes en las reglas son conectados por operaciones de *norma-t* (mínimo o producto). Los grados de pertenencia en los conjuntos de entrada son combinados con aquellos en los conjuntos de salida usando la composición *sup – star*. Múltiples reglas pueden ser combinadas usando operaciones de *norma-t* o durante la defuzzificación por suma de pesos. En el caso del tipo-2 el proceso de inferencia es muy similar. El motor de inferencia mapea el conjunto tipo-2 de entrada en uno de salida. Para la inferencia del sistema desarrollado se utilizó el producto *norma-t* (Ecuación 4.6) y la *norma-s* máxima (Ecuación 4.6) , obteniendo la FOU del sistema para ser interpretada en la reducción de tipo.

$$T_{prod}(a,b) = a \cdot b \quad (4.6)$$

$$S_{max}(a,b) = \max(a,b) \quad (4.7)$$

Reducción de Tipo

Para el cómputo de la salida del sistema en la defuzzificación es necesario conocer los puntos de cambio en la huella de incertidumbre mediante y_l y y_r , estos valores se obtienen mediante algoritmos de reducción de tipo aplicados a la FOU de salida, ver ejemplo de Figura 4.14.

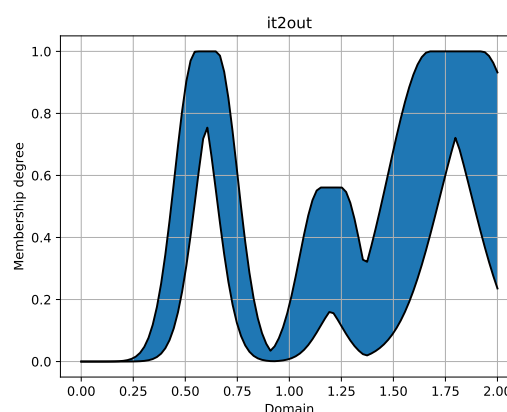


Figura 4.14: Ilustración de FOU de salida del sistema .

El Algoritmo Mejorado Iterativo con condición de Paro (EIASC) propuesto en (Duran *et al.*, 2008) fue el que se consideró después del estudio de la literatura y prueba de otros reductores de tipo. Esta reducción de tipo proporciona una velocidad de procesamiento significativamente más rápida que otros métodos,

como el algoritmo Karnik-Mendel o de Centro de Conjuntos, siendo que se busca una implementación en tiempo real resultó el más idóneo para este trabajo. Su velocidad se debe a que EIASC presenta una inicialización previa, una parte iterativa con condición de paro y una búsqueda en sentido opuesto, el proceso iterativo para encontrar y_l y y_r se puede muestra en la Tabla 4.1.

Tabla 4.1: Algoritmo EIASC.

| Paso | EIASC para $y_l(L) = \min\left(\frac{\sum_{i=1}^n a_i w_i}{\sum_{i=1}^n w_i}\right)$ | EIASC para $y_r(R) = \max\left(\frac{\sum_{i=1}^n b_i w_i}{\sum_{i=1}^n w_i}\right)$ |
|------|---|---|
| 1 | Inicialización $a = \sum_{i=1}^n a_i c_i$ $b = \sum_{i=1}^n c_i$ $L = 0$ | Inicialización $a = \sum_{i=1}^n b_i d_i$ $b = \sum_{i=1}^n d_i$ $R = n$ |
| 2 | Cómputo de: $L = L + 1$ $a = a + a_L(d_L - c_L)$ $b = b + (d_L - c_L)$ $y_l(L) = a/b$ | Cómputo de: $a = a + b_r(d_r - c_r)$ $b = b + (d_r - c_r)$ $y_r(R) = a/b$ $R = R - 1$ |
| 3 | SI $y_l(L) \leq a_{L+1}$, se Detiene, de otra forma regresa al paso 2. | SI $y_r(R) \geq b_r$, se Detiene, de otra forma regresa al paso 2 |

Defuzzificación

Con los valores de la salida del sistema se realiza un promedio de los puntos de cambio y_l y y_r obtenidos en la reducción de tipo tal como se indica en 4.8.

$$y = \frac{y_l + y_r}{2} \quad (4.8)$$

Obteniendo un valor de salida y que es utilizado como resultado del sistema. Un ejemplo de salida para la FOU de la Figura 4.15, con los valores de entrada $EAR=1.19$, $MAR=0.33$, $CI:0.54$, $CD=0.78$, $CA=0.86$, $DisCars=0.16$; se obtiene $y_l=0.54791106$ y $y_r=0.75995477$ con el reductor de tipo EIASC. Por último, mediante el promedio de y_l y y_r resulta en una salida y de 0.65393292, donde predomina el conjunto de *Nivel de riesgo Medio* debido a que el valor de CA representa una pequeña inclinación hacia abajo, y los valores de CI y CD representan un pequeño giro de la posición de la cabeza hacia la derecha.

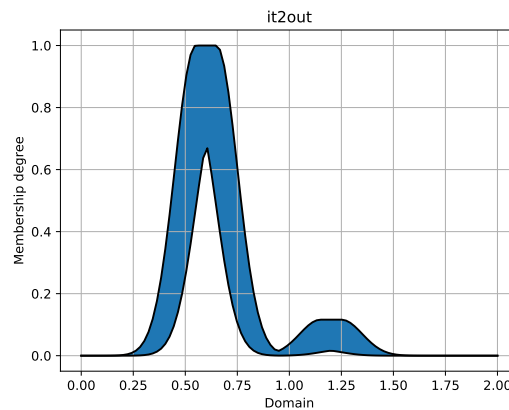


Figura 4.15: Ilustración de FOU de salida del sistema.

Como resultado, se diseñó y desarrolló el IT2FLS para el sistema de asistencia que se adecua para la tarea propuesta. Un diagrama donde se detallan los componentes del IT2FLS construido se muestra en la Figura 4.16.

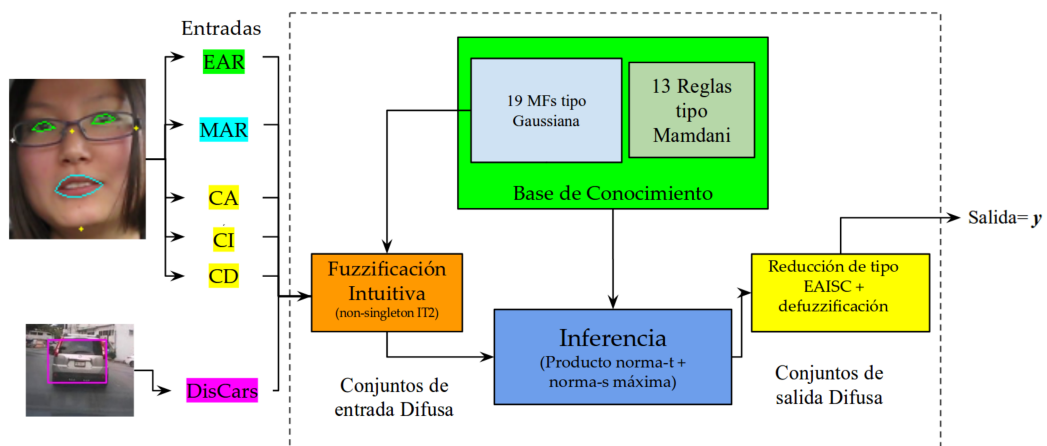


Figura 4.16: Diagrama del IT2FLS para el asistente de conducción propuesto.

4.1.3. Implementación


Para la implementación se utilizó la tarjeta *Jetson TX2* de NVIDIA. La configuración de *software*, *hardware* y de instalación en el vehículo se describen en esta sección.

Jetson TX2

Los sistemas embebidos de la familia *Jetson* de la marca NVIDIA son dispositivos que se enfocan en el desarrollo y despliegue de aplicaciones de Inteligencia Artificial.







Para este trabajo se optó por el uso del modelo TX2, esta versión cuenta con características que se adecuan a los requisitos requeridos para el desarrollo del trabajo y que se describen en la Tabla 4.2.

Tabla 4.2: Características de Jetson TX2.

| Característica | Jetson Tx2 |  |
|----------------|--|---|
| GPU | Pascal 256 Núcleos | |
| CPU | Dual-Core Denver 64-Bit Quad-Core ARM | |
| Memoria | 8GB LPDDR4 | |
| Almacenamiento | 32GB | |
| PCIe | Gen 2-1x4 + 1x1 OR 2x1 + 1x2 | |
| CSI | 12x CSI2 D-PHY | |
| Display | DP 1.2 eDP 1.4 HDMI 2.0 DOS 1x4 DSI | |
| Wifi | Si | |

Después de la adquisición de la plataforma se migró el sistema de adquisición y se desarrolló el controlador IT2FLS. Para el desarrollo del sistema se contó con las siguientes especificaciones de software : Sistema Operativo ubuntu 18.04, Dlib 3.17, OpenCV 3.4.1, PyIT2FLS y Jetson.GPIO. El hardware utilizado para el desarrollo del sistema de asistencia se muestra en la Tabla 4.3.

Tabla 4.3: Especificaciones de elementos de Hardware.

| Nombre | Cantidad | Descripción | Características | Imagen |
|------------------------------------|----------|---|---|---|
| <i>Display</i> <i>Waveshare</i> | 1 | Entrada de información mediante la característica <i>touch</i> y visualización del entorno, mediciones y salidas del asistente. | 7 pulgadas <i>Touch Screen</i> Puerto HDMI 5V/2A |  |
| Hub USB 3.0 UGREEN | 1 | Concentrador USB para los diversos dispositivos del asistente. | 7 puertos 3.0 Cable USB 3.0/1m 5V/2A |  |
| Batería Atomicthree | 1 | Fuente de energía para el asistente y los periféricos. | 13600 mah 68 wh 19V/3.5A |  |
| <i>Buzzer</i> | 1 | Actuador para avisos sonoros al conductor. | 5V/30mA |  |
| Cámara Logitech C170 | 2 | Dispositivos para captura de secuencias de video en tiempo real. | Resolución 640x480 5 MP 30 FPS |  |
| Soporte ATVIO | 2 | Soportes para posicionamiento de las cámaras. | Rotación 360 Ventosa adherible Soporte Adaptable |  |

Para la implementación del sistema de asistencia en el vehículo se consideró el diagrama de conexión de la Figura 4.17.

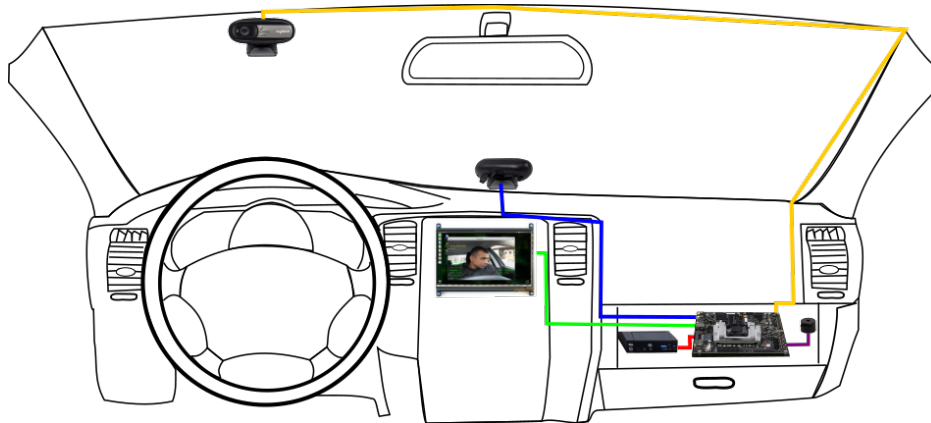


Figura 4.17: Diagrama de conexión del sistema.

Para el posicionamiento de las cámaras de monitoreo del conductor y el camino se utilizó un par de soportes para *Smartphones* colocados, como se muestra en las Figuras 4.18a y 4.18b, respectivamente.



(a) Posición de la cámara de conductor. (b) Posición de la cámara del camino.

Figura 4.18: Implementación de las cámaras en el vehículo.

La pantalla se colocó en la parte central del vehículo para consulta del funcionamiento correcto del sistema y calibración de la vista de las cámaras, Figura 4.19a. Por último, los periféricos y complementos como la batería, el HUB, el *Buzzer* y la *Jetson TX2* se colocaron en la guantera con el fin de estar resguardados, tal como se muestra en la Figura 4.19b.

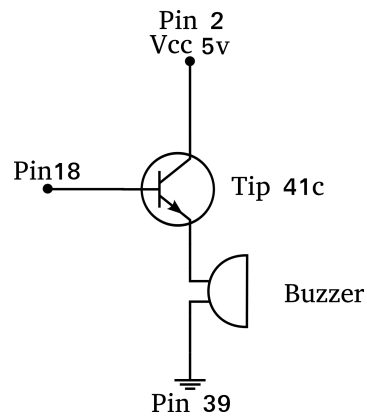


Figura 4.21: Circuito de conexión de *Buzzer*.

4.2. Discusión

Hasta este punto se expuso el diseño y desarrollo del sistema de asistencia propuesto en sus tres módulos principales. Como resultado se obtuvo un prototipo funcional que se evalúa mediante métricas de detección del estado del arte adaptadas a experimentos que se describen en el siguiente capítulo.

Capítulo 5

Pruebas y Resultados

En este capítulo se presentan los experimentos diseñados e implementados para este trabajo, así como las métricas de evaluación y resultados obtenidos.

5.1. Diseño e Implementación de Pruebas

En esta sección se detallan los experimentos y métricas implementadas al sistema de asistencia propuesto.

5.1.1. Métricas

Para la evaluación se utilizaron métricas y conceptos enfocados en detección que más destacan en el estado del arte (*Velez and Otaegui, 2016*). Los conceptos utilizados clasifican las detecciones en relación a lo predicho con lo que es realmente y se definen como se muestra a continuación:

- *VP*: Positivos clasificados correctamente como positivos, una detección correcta.
- *VN*: Negativos clasificados correctamente como negativos.
- *FN*: Positivos clasificados incorrectamente como negativos, un *Ground-Truth* no detectado.
- *FP*: Negativos clasificados incorrectamente como positivos, una detección errónea.

Estos conceptos de detección se utilizaron para evaluar el sistema mediante las métricas de Precisión, *Recall* y *F1-Score*. La métrica de Precisión representa

la capacidad de identificar los objetos relevantes y se define como el número de verdaderos positivos dividido por la suma de verdaderos positivos y falsos positivos:

$$Precisión = \frac{VP}{VP + FP} \quad (5.1)$$

La segunda métrica es la de *Recall* que representa la capacidad de encontrar todos los casos relevantes (todos los cuadros delimitadores del *Ground-Truth*). Se define como el número de verdaderos positivos dividido por la suma de verdaderos positivos y falsos negativos:

$$Recall = \frac{VP}{VP + FN} \quad (5.2)$$

Se plantea el uso de la métrica *F1-Score*, que es la media armónica de Precisión y del Recall, donde entre más se acerca el valor a 1 es el mejor valor (perfecta Precisión y *Recall*) y peor si se acerca a 0. Para interpretar los valores obtenidos por las métricas propuestas, en 5.3 se muestra la composición de esta media.

$$F1 - Score = \frac{Recall * Precisión}{Recall + Precisión} \cdot 2 \quad (5.3)$$

Estos conceptos y métricas se adaptaron para los diferentes experimentos que se presentan en la siguiente subsección.

5.1.2. Experimentos

Los experimentos realizados para evaluar el sistema de asistencia se dividieron en tres tareas: detección de vehículos, detección de inatención y detección de nivel de riesgo.

Detección de Vehículos

En este experimento se mide la eficacia de detección de vehículos en la parte frontal del camino, se optó por el uso de la métrica *Intersection Over Union (IOU)* para medir la precisión de la detección. *IOU* se define como el área de la intersección dividida por el área de la unión de un *bounding box* predicho (B_p) y un *ground-truth box* (B_{gt}):

$$IOU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (5.4)$$

La detección se considera correcta o *VP* cuando *IOU* supera un umbral, en este caso ≥ 0.7 , que coincide con B_{gt} . Se considera una detección incorrecta o *FP* en caso contrario $IOU < 0.7$ y un *FN* cuando un B_{gt} no es detectado. *VN* serían todos los cuadros delimitadores posibles que no se detectaron correctamente (tantos cuadros posibles dentro de una imagen). Una representación de *IOU* puede observarse en la Figura 5.1.



Figura 5.1: Representación gráfica de *IOU*

Para la comparación de este experimento se utilizaron detectores probados durante el desarrollo de este trabajo, entre ellos se encuentran el clasificador en cascada *Haar* (*Viola and Jones, 2004*) y el detector *MobileNets* de (*Howard et al., 2017*). Se usó la base de datos *TME Motorway* (*Caraffi et al., 2012*) y secuencias de video propias que se etiquetaron manualmente con la herramienta *ImgLab* de la librería *Dlib*. En la Figura 5.3 pueden observarse cuadros con *Ground-Truth* y resultados cualitativos de los detectores. La parte cuantitativa medida con las métricas de *Precisión*, *Recall* y *IOU* se abordan en la sección de resultados.

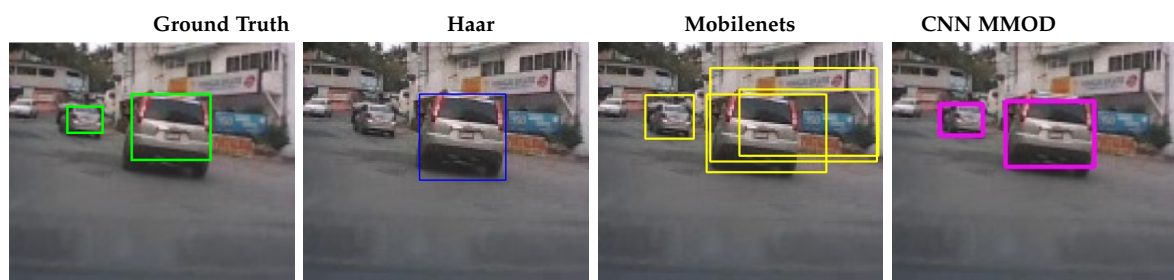


Figura 5.2: Resultados de detectores en secuencias de video no controladas I.

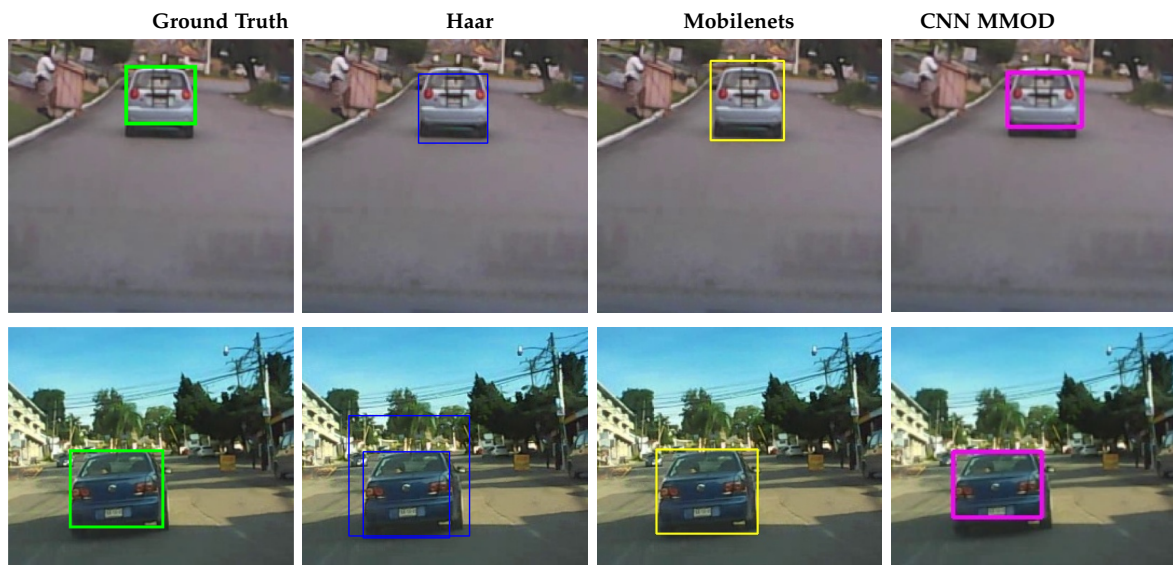


Figura 5.3: Resultados de detectores en secuencias de video no controladas II.

Inatención

Para esta evaluación se utilizaron las bases de datos *UTA Real-Life Drowsiness* (Ghoddosian et al., 2019), *YawDD* (Abtahi et al., 2014) y secuencias de vídeo del sistema propuesto etiquetadas manualmente. Para este caso, se interpretó el nivel de riesgo (siendo nivel bajo, medio, alto y somnoliento) como resultado de la inatención del conductor. Las mediciones se definieron de la siguiente manera: Un *VP* ocurre cuando el sistema detecta correctamente una situación de riesgo de los niveles medio, alto y somnoliento correspondientemente. Un *VN* ocurre cuando el sistema detecta correctamente una situación que no es de riesgo. Un *FP* ocurre cuando una situación que no es de riesgo se detecta incorrectamente como de riesgo o la situación de riesgo no corresponde a su nivel. Se produce un *FN* cuando no se detecta una situación de riesgo.

En la Figura 5.4, pueden observarse detecciones de los diferentes niveles en diversas situaciones no controladas. A las diversas secuencias de video se le aplicaron las métricas de Precisión y *Recall*, el resultado de estas métricas se abarcan en la Sección 5.2.

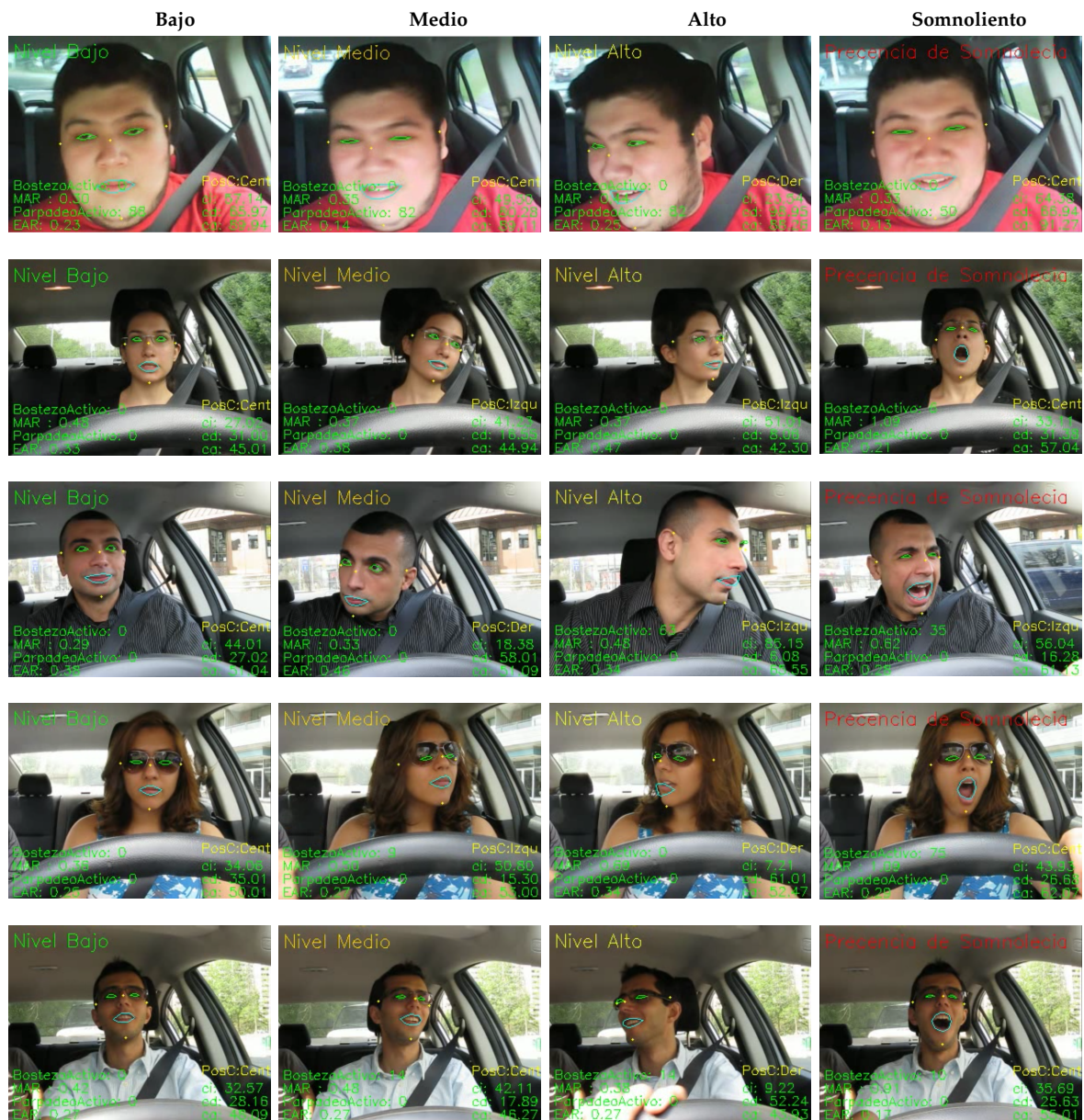


Figura 5.4: Detecciones de diferentes niveles de riesgo .

En diversos trabajos enfocados en la detección de inatención y situaciones de riesgo para diferentes campos, como el de autos inteligentes, se encuentra que hacen uso de diferentes sensores, algoritmos, inferencia y entornos de conducción. Esto conduce a que varios trabajos sean de carácter técnico que se centran en presentar la metodología y el prototipo conseguido, como el caso de (Pattarkine, 2018). Debido a estos motivos realizar comparaciones, en este campo, es algo subjetivo, tal como se hace en *surveys* donde se comparan métricas de detección

para diferentes tipos de tareas que se pueden interpretar de diversas formas y donde se recalca que hacer una comparación justa es uno de los grandes retos de este campo de investigación (Velez and Otaegui, 2016). Conforme a esto, se optó por evaluar sólo con las métricas antes propuestas, este experimento de detección de inatención y el siguiente experimento de detección de inatención y vehículo frontal.

Inatención y Vehículo Frontal

Para este experimento se utilizó la detección de inatención y del vehículo frontal para inferir el nivel de riesgo en los niveles antes explicados. Para este caso, al ser la unión de los componentes del sistema no se encontró una base de datos que contuviera un *Ground-Truth* que satisficiera la evaluación del asistente; por lo cual se utilizaron las secuencias de video de inatención, que fueron complementadas con secuencias del camino que se grabaron simultáneamente, redefiniendo los niveles de riesgo dependiendo de la inatención y la distancia del vehículo frontal más cercano. Ejemplos de la detección de nivel de riesgo se pueden observar en la Figura 5.5, donde se puede apreciar el nivel de riesgo y la salida correspondiente del IT2FLS.



Figura 5.5: Pruebas realizadas de detección de nivel de riesgo del sistema de asistencia.

5.2. Resultados

Como se mencionó anteriormente, la evaluación del sistema se compone de tres experimentos. En esta sección se describen de forma resumida los resultados obtenidos.

5.2.1. Detección de Vehículos

Se realizó el experimento con las dos bases de datos anteriormente mencionadas. Para la evaluación de los detectores (con la base de datos *TME Motorway*) se utilizaron 1176 imágenes de diferentes secuencias de video, donde se encuentran etiquetadas las áreas donde existe la parte trasera de un vehículo con la herramienta *ImgLab*. Las áreas de interés se encuentran marcadas con un cuadro delimitador rojo (como en la Figura 5.6a) y las áreas similares pero que son ignoradas con un cuadro delimitador con una equis, como se muestra en la Figura 5.6b donde son ignorados un camión y una motocicleta, debido a que presentan características con un grado intermedio de similitud que fueron excluidas en el entrenamiento de la red.



(a) Etiquetación correcta en TME.



(b) Etiquetas ignoradas en TME.



(c) Etiquetación correcta en videos propios.



(d) Etiquetas ignoradas en videos propios.

Figura 5.6: Pruebas realizadas de detección de nivel de riesgo del sistema de asistencia.

Los resultados de los detectores *Haar*, *Mobilenets* y CNN MMOD promediados para *TME Motorway*, se encuentran en la Tabla 5.1. Con base a los resultados se puede observar que el detector *Haar* tiene un rendimiento inferior, esto debido a su alta tasa de detección de falsos positivos y el amplio margen de sus cuadros delimitadores. En el caso de *Mobilenets* su rendimiento es mejor, pero en situaciones no controladas puede hacer detecciones erróneas. CNN MMOD demuestra tener mejores resultados en situaciones no controladas y sus cuadros delimitadores de detección tienen una mejor intersección con el *Ground-Truth*.

Tabla 5.1: Resultados para la base de datos *TME Motorway*.

| Detector | <i>Recall</i> Promedio | Precisión Promedio | <i>IOU</i> Promedio |
|-------------------|---------------------------|-----------------------|------------------------|
| <i>Haar</i> | 86.53 | 88.39 | 76.54 |
| <i>Mobilenets</i> | 90.21 | 92.45 | 80.42 |
| CNN MMOD | 97.04 | 95.91 | 88.74 |

Para la base de datos de secuencias de video propio, se utilizó *ImgLab* para etiquetar 675 cuadros diferentes, se usó la misma etiquetación de áreas de interés (Figura 5.6c) y a ser ignoradas (Figura 5.6d). Los resultados son mostrados en la Tabla 5.2, donde igual que con la tabla anterior, se puede apreciar un mejor rendimiento con CNN MMOD, es este caso para imágenes que presentan variaciones de iluminación y que fueron capturadas desde las cámaras del sistema de asistencia desarrollado.

Tabla 5.2: Resultados para la base de datos de secuencias de video propias.

| Detector | <i>Recall</i> Promedio | Precisión Promedio | <i>IOU</i> Promedio |
|-------------------|---------------------------|-----------------------|------------------------|
| <i>Haar</i> | 82.22 | 80.47 | 68.77 |
| <i>Mobilenets</i> | 90.38 | 92.91 | 79.43 |
| CNN MMOD | 95.82 | 96.55 | 82.46 |

La razón de porque los primeros dos detectores tienen un rendimiento inferior se debe a que no son aptos para entornos no controlados. En el caso de *Haar* se debe a su alta detección de falsos positivos debido a las variaciones de iluminación, mientras que para *Mobilenets* el problema reside en que trata de ajustar detecciones a regiones similares a la parte trasera de un automóvil, por la configuración de su entrenamiento. Conforme a los resultados mostrados se justifica el uso del detector CNN MMOD para esta implementación, obteniendo promedios entre las dos bases de datos de *Recall* de 96.43, *Precisión* de 96.23 y *IOU* de 85.6.

5.2.2. Detección de Inatención

Para las pruebas de detección de inatención se utilizaron secuencias de video en las cuales los niveles de riesgo se clasificaron dependiendo de la situación del cuadro. La base de datos *UTA Real-Life Drowsiness* se conforma por videos de personas realizando diferentes actividades, entre las que se encuentran actividades de conducción, que se clasifican en tres clases que son: Alerta, Poco vigilante y Somnoliento. Para la base de datos de *UTA Real-Life Drowsiness* se utilizaron partes de 1 a 2 minutos de ocho videos, de los cuales tres contienen un factor de oclusión. Los resultados de cuadros por segundo (FPS), *Recall*, Precisión y *F1-Score* son mostrados en la tabla 5.3.

Tabla 5.3: Resultados para la base de datos *UTA Real-Life Drowsiness*.

| Base de datos | Archivo | FPS | Oclusión | <i>Recall</i> | Precisión | <i>F1-Score</i> |
|---------------------------------|------------------------|-------|------------------------|---------------|-----------|-----------------|
| <i>UTA Real-Life Drowsiness</i> | Fold1_part1/Video1.wmv | 25.32 | No | 0.924 | 0.955 | 0.937 |
| | Fold1_part1/Video2.wmv | 26.02 | No | 0.935 | 0.915 | 0.924 |
| | Fold1_part1/Video3.wmv | 25.68 | No | 0.926 | 0.947 | 0.941 |
| | Fold1_part1/Video4.wmv | 26.35 | Lentes Prescripción | 0.868 | 0.855 | 0.861 |
| | Fold1_part1/Video5.wmv | 26.17 | No | 0.938 | 0.946 | 0.941 |
| | Fold1_part1/Video6.wmv | 25.55 | Lentes Prescripción | 0.87 | 0.854 | 0.861 |
| | Fold1_part1/Video7.wmv | 25.78 | Lentes Prescripción | 0.891 | 0.864 | 0.877 |
| | Fold1_part1/Video8.wmv | 26.39 | No | 0.933 | 0.943 | 0.937 |
| | Promedio | | | 0.910 | 0.909 | 0.910 |

Los resultados para UTA, se puede apreciar que se tienen valores inferiores en los casos de oclusión por lentes de prescripción, el promedio de procesamiento en FPS es de 25.907, lo cual se considera aceptable para implementación en tiempo real y se obtuvo un *F1-Score* de 0.910 que representa un buen balance entre *Recall* y Precisión.

La base de datos YawDD se conforma por videos de participantes masculinos y femeninos, realizando diversas actividades al volante con condiciones de iluminación reales; además, algunos participantes cuentan con objetos de oclusión. Para esta base de datos se utilizaron partes de 1 a 2 minutos de nueve videos, de los cuales cuatro contienen un factor de oclusión por lentes de prescripción y uno por lentes de sol. El resultado para esta base de datos es mostrado en la Tabla 5.4.

Tabla 5.4: Resultados para base de datos *YawDD*.

| Base de datos | Archivo | FPS | Oclusión | Recall | Precisión | F1-Score |
|---------------|------------------------|-------|------------------------|--------|-----------|----------|
| <i>YawDD</i> | 1-FemaleNoGlasses.avi | 23.65 | No | 0.922 | 0.933 | 0.927 |
| | 4-FemaleNoGlasses.avi | 25.85 | No | 0.934 | 0.957 | 0.945 |
| | 8-FemaleGlasses.avi | 24.52 | Lentes Prescripción | 0.896 | 0.905 | 0.900 |
| | 12-FemaleGlasses.avi | 24.09 | Lentes Prescripción | 0.902 | 0.855 | 0.877 |
| | 11-FemaleGlasses.avi | 26.3 | Lentes Sol | 0.843 | 0.881 | 0.861 |
| | 10-FemaleNoGlasses.avi | 24.77 | No | 0.901 | 0.942 | 0.921 |
| | 12-MaleGlasses.avi | 25.76 | Lentes Prescripción | 0.892 | 0.911 | 0.901 |
| | 13-MaleNoGlasses.avi | 23.68 | No | 0.924 | 0.947 | 0.935 |
| | 15-MaleGlasses.avi | 26.23 | Lentes Prescripción | 0.905 | 0.901 | 0.902 |
| | Promedio | | 0.896 | 0.911 | 0.903 | |

Para *YawDD* se pueden observar, también, resultados inferiores para videos con oclusiones, ya que la oclusión con lentes de sol provoca incertidumbre ya que no se conoce realmente el estado de los ojos de la conductora (Figura 5.4). Para este caso se tiene un *F1-Score* un poco menor (0.903) a comparación al la base de datos anterior con un promedio de cuadros de 24.87.

Por último, con seis secuencias de video propias, se utilizaron videos entre 1 a 3 minutos de duración, siendo que sólo una contiene un factor de oclusión. Los resultados pueden observarse en la Tabla 5.5. Estas secuencias fueron tomadas con el asistente implementado en el vehículo en tiempo real y se obtuvo un procesamiento promedio de FPS de 24.63, teniendo que considerar que se realizaron en simultáneo el procesamiento del asistente y el grabado de dos archivos de video que correspondían a el conductor y camino.

Tabla 5.5: Resultados para secuencias de video propias.

| Base de datos | Archivo | FPS | Oclusión | Recall | Precisión | F1-Score |
|---------------|-----------------|-------|------------------------|--------|-----------|----------|
| Propio | Conductor_1.avi | 23.65 | No | 0.908 | 0.937 | 0.922 |
| | Conductor_2.avi | 25.85 | No | 0.91 | 0.924 | 0.916 |
| | Conductor_3.avi | 25.52 | No | 0.886 | 0.901 | 0.904 |
| | Conductor_4.avi | 23.04 | No | 0.906 | 0.912 | 0.903 |
| | Conductor_5.avi | 24.09 | Lentes Prescripción | 0.859 | 0.885 | 0.884 |
| | Conductor_6.avi | 25.68 | No | 0.915 | 0.931 | 0.899 |
| | Promedio | | 0.897 | 0.915 | 0.905 | |

Para las secuencias de video propias se puede apreciar, también, un resultado inferior para la oclusión por lentes de prescripción; esto se presenta en las tres bases de datos y es debido a que el detector de puntos de referencia intenta hacer una aproximación y en ocasiones por diversos factores, como un reflejo en los cristales del lente, se obtienen valores erróneos o poco aproximados. El resultado obtenido en la implementación real demuestra valores promedio aceptables de *Recall* de 0.89, Precisión de 0.915 y *F1-Score* de 0.905.

5.2.3. Detección de Nivel de Riesgo

Para el experimento de detección de nivel de riesgo general, con la detección de inatención y vehículo frontal, se utilizaron las secuencias de video ya mencionadas. En el experimento de inatención teniendo como agregado la detección de las secuencias de vídeo del camino, como puede observarse en la Figura 5.5. Se re-etiquetaron los niveles de riesgo para cada situación, en este caso con ambos cuadros y se obtuvieron los resultados contenidos en la Tabla 5.6.

Tabla 5.6: Resultados de detección de nivel de riesgo para secuencias de video propias.

| Base de datos | Archivo | FPS | Oclusión | <i>Recall</i> | Precisión | <i>F1-Score</i> |
|---------------|--------------------------------|-------|---------------------|---------------|-----------|-----------------|
| Propio | Conductor_1.avi & Camino_1.avi | 21.65 | No | 0.925 | 0.942 | 0.933 |
| | Conductor_2.avi & Camino_2.avi | 25.85 | No | 0.923 | 0.939 | 0.930 |
| | Conductor_3.avi & Camino_3.avi | 25.52 | No | 0.892 | 0.911 | 0.914 |
| | Conductor_4.avi & Camino_4.avi | 23.04 | No | 0.927 | 0.934 | 0.918 |
| | Conductor_5.avi & Camino_5.avi | 24.09 | Lentes Prescripción | 0.877 | 0.889 | 0.904 |
| | Conductor_6.avi & Camino_6.avi | 25.68 | No | 0.922 | 0.949 | 0.905 |
| | | | Promedio | 0.911 | 0.927 | 0.917 |

En este caso, el procesamiento promedio de FPS (24.63) es el mismo que en el de inatención debido a que se capturaron simultáneamente. En este caso se tiene información proveniente de dos cuadros con condiciones diferentes, siendo este uno de los motivos de usar la detección con puntos de referencia y CNN MMOD que permiten trabajar con ambientes no controlados, obteniendo los valores de entrada antes mencionados (*EAR, MAR, CI, CD, CA* y *DisCars*) con cierta incertidumbre

que son procesados con el IT2FLS para obtener una detección aproximada del nivel de riesgo. Como resultado se obtuvieron los valores de *Recall* de 0.911, Precisión de 0.927 y un *F1-Score* de 0.917 para el sistema de asistencia propuesto.

5.3. Discusión

En este Capítulo se detallaron las métricas de detección utilizadas para la evaluación del asistente a través de tres experimentos. Los experimentos se diseñaron para evaluar la detección de vehículos frontales, inferencia de nivel de riesgo mediante detección de inatención e inferencia de nivel de riesgo mediante la conjunción de detección de vehículos frontales e inatención. Los experimentos se aplicaron a secuencias de video obtenidas del prototipo de asistente y de bases de datos con información relevante para cada experimento. Como resultado se obtuvieron valores superiores a 0.90 en las métricas de *Recall*, *Precisión* y *F1-Score*, que representan un valor aceptable en el campo de investigación y desarrollo de asistentes de conducción.

Capítulo 6

Conclusiones

En este capítulo se realiza un análisis de las conclusiones a las que se llegaron después de la evaluación del sistema propuesto y de estudiar los resultados obtenidos. También se realiza un análisis de trabajos futuros.

6.1. Objetivos y Alcances Logrados

En la Tabla 6.1, se muestran las actividades realizadas en relación a los objetivos y alcances planteados para el desarrollo de la tesis.

Tabla 6.1: Objetivos y alcances realizados.

| Objetivo | Actividad |
|---|--|
| Analizar la lógica difusa de tipo-2 para el desarrollo del sistema asistente de conducción. | Se estudió la teoría de conjuntos difusos tipo-1 y tipo-2 (intervalo y general), además de que se realizó la implementación de controladores tipo 1 y 2. |
| Experimentación con técnicas de Visión por Computadora para obtención e interpretación de secuencias de imagen. | Se estudió e implementaron técnicas de visión para entornos no controlados (clasificadores en cascada <i>Haar</i> , HOG, detección de puntos de referencia facial, <i>MobileNets</i> , <i>Tiny YOLOv3</i> , CNNMMOD.). |
| Analizar los sistemas embebidos en computación heterogénea. | Se estudió y experimentó con la NVIDIA <i>Jetson</i> TK1 y TX2. |

| Realizar un sistema de alertas sonoro para el sistema. | Se creó un circuito con un <i>Buzzer</i> como actuador de salida para el nivel alto y detección de somnolencia. |
|---|---|
| Implementar el sistema desarrollado en el sistema embebido seleccionado. | Se implementó la parte de adquisición de información y el IT2FLS en <i>Jetson TX2</i> . |
| Comparar los resultados obtenidos con métricas del estado del arte. | Se evaluó el sistema con métricas de detección (<i>Recall</i> , <i>Precisión</i> y <i>F1-Score</i>) expuestas en (<i>Velez and Otaegui, 2016</i>). |
| Alcances | Actividad |
| Experimentar con técnicas de Visión por Computadora para la identificación de obstáculos y monitoreo del conductor. | Se usó CNNMMOD para la detección de vehículos frontales y se entrenó un predictor de 22 puntos de referencia para el monitoreo del conductor. |
| Desarrollo del sistema propuesto usando Lógica Difusa Tipo-2. | Se diseño y desarrollo un controlador IT2FLS con 6 entradas, 13 reglas Mamdani, 19 MFs gaussianas tipo-2, reductor de tipo EAISC y una salida <i>y</i> . |
| Realizar pruebas experimentales del sistema en ambientes reales. | Se implementó el prototipo del sistema de asistencia en un vehículo donde se presentaron entornos no controlados (ambientes reales). |
| Evaluar el desempeño del sistema propuesto. | Se realizó la evaluación en tres experimentos diferentes. Adaptado métricas de detección y utilizando secuencias de vídeo de bases de datos y obtenidas por el sistema. |

6.2. Resultados del Trabajo

Productos

Durante el desarrollo de este proyecto se obtuvieron los siguientes productos:

1. Reporte del estado del arte: documento en el que se presenta un resumen de diversas publicaciones recientes que tratan los tópicos del tema de tesis.
2. Diseño, desarrollo e implementación del prototipo de asistente de conducción.

3. Predictor de 22 puntos de referencia faciales.
4. Poster presentado en la Primera Jornada de Ciencia y Tecnología Aplicada del CENIDET (Figura A.2).
5. Poster presentado en la Escuela de Inteligencia Artificial y Robótica 2018 de la UTEZ y congreso ICMEAE 2018 (Figura A.1 y A.3).
6. Artículo para la Segunda Jornada de Ciencia y Tecnología Aplicada del CENIDET (Figura A.5).
7. Co-autor de artículo publicado en CONIELECOMP, UDLAP 2019 (Figura A.4).
8. Artículo para la Escuela de Inteligencia Artificial y Robótica 2019 de la UTEZ (Figura A.7).

6.2.1. Aportaciones

Como aportaciones al tema de los vehículos autónomos y al marco teórico se obtuvieron:

1. Controlador IT2FLS para la detección de nivel de riesgo en tarea de conducción.
2. Prototipo completo del sistema de asistencia embebido.
3. Predictor de 22 puntos de referencia para localización de regiones faciales y detección de parpadeo, bostezo y posición de cabeza.

6.2.2. Conclusiones

En la terminación de este proyecto de maestría, se llegan a las siguientes conclusiones:

1. En este trabajo se diseñó, desarrolló e implementó un sistema embebido para asistencia de conducción basado en visión y Lógica Difusa Tipo-2 para la detección de situaciones de riesgo y alerta de estas.
2. En la parte de adquisición de información se trabajó con técnicas de visión que permitieran la detección de eventos y regiones de interés en entornos no controlados. Para el monitoreo del conductor se utilizó relaciones de aspecto sobre detección de rasgos faciales mediante puntos de referencia, entrenando

un predictor de 22 puntos con un error medio de 8.40, una ventaja de esta técnica es la posibilidad de detección de la región ocular a través de lentes de prescripción y la tolerancia a otros factores de oclusión . Para la detección de vehículos en la parte frontal del camino se optó por la utilización de una CNN MMOD la cual se comparó con otros detectores (*Haar, Mobilenets*) probados durante el desarrollo del sistema obteniendo valor promedio de 96.43 de *Recall*, 96.23 de precisión y *IOU* de 85.6 en la base de datos *TME Motorway* y secuencias de video capturadas con el asistente desarrollado.

3. En la parte de inferencia se diseñó y creó un IT2FLS con 6 entradas, 13 reglas Mamdani, 19 MFs gaussianas tipo-2, reductor de tipo EAISC y una salida y que representa los niveles de riesgo bajo, riesgo medio, riesgo alto y somnolencia.
4. El asistente se construyó con una *Jetson TX2* para el procesamiento, dos cámaras *Logitech c170* con soporte para captura de información, batería *Atomichthree* como fuente de energía, HUB usb 3.0 como concentrador de dispositivos, *Display Waveshare touch screen* para introducción/visualización de información y *Buzzer* para emisión de alertas sonoras. Se implementó el sistema de asistencia en un vehículo, se realizaron pruebas del mismo y se capturó simultáneamente secuencias de vídeo del conductor y camino sin procesar para el etiquetado manual del nivel de riesgo.
5. En la experimentación se encontró con la limitante del que al comparar lo desarrollado con otros trabajos del campo de lo vehículos inteligentes no es posible hacer una comparación completamente, ya que no existe un procedimiento estándar para recopilar y etiquetar los conjuntos de datos de video utilizados para calcular métricas y la longitud difieren significativamente de un trabajo a otro siendo que la mayoría es de carácter técnico y en diferentes *surveys* se recalca el desafío que aún no se ha resuelto: la estandarización de la preparación y validación de los conjuntos de datos de evaluación. Debido a esto se considera aceptable un valor mayor a 0.9 en métricas de *Recall*, Precisión y *F1-Score* por lo cual fueron consideradas para este trabajo.
6. Conforme a lo anterior, se realizó la comparación con datasets con *Ground-truth* y se realizó la etiquetación manual de secuencias de video propios. Se diseñaron tres experimentos para medir los resultados del sistema.

El primero para la detección de vehículos frontales cuyo resultado se mencionó en el punto 2. El segundo para la detección de Inatención en donde se utilizaron los datasets UTA *Real-Life Drowsiness* (*Recall* = 0.910, *Precisión* = 0.909 y *F1-Score* = 0.9105), YawDD (*Recall* = 0.896, *precisión* = 0.911 y *F1-Score* = 0.903) y conjunto de secuencias de video propias (*Recall* = 0.897, *Precisión* = 0.915 y *F1-Score* = 0.905). En el tercer experimento tuvo valores de *Recall* = 0.911, *Precisión* = 0.927, *F1-Score* = 0.917, con una tasa de FPS promedio de 24.63 que indica un procesamiento en tiempo real.

6.2.3. Trabajo futuro

Como trabajo futuro se contempla la implementación en paralelo del sistema propuesto para un mejor desempeño e incorporación de más funciones, así como la refinación con algoritmos y técnicas más robustas. En la parte de adquisición de información, experimentar y entrenar otros modelos como Tiny Yolo para la detección de obstáculos de otras clases (peatones, autobuses, motocicletas, etc) *Redmon et al. (2016)*. También se considera la implementación de otras tareas propias de los asistentes de conducción, tales como detección de carril, colisiones, señales de tráfico y obstáculos en puntos ciegos. En la parte difusa experimentar con métodos de reducción de reglas y uso de otros reductores de tipo como: método Yeh, KM eficiente, IASCO, Límites inciertos, etc.

Bibliografía

- Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., and Hariri, B. (2014). Yawdd: A yawning detection dataset. In *Proceedings of the 5th ACM Multimedia Systems Conference*, pages 24–28. ACM.
- Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2930–2940.
- Borrmann, J. M., Haxel, F., Viehl, A., Bringmann, O., and Rosenstiel, W. (2015). Safe and efficient runtime resource management in heterogeneous systems for automated driving. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 353–360. IEEE.
- Caraffi, C., Vojir, T., Trefny, J., Sochman, J., and Matas, J. (2012). A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera. In *ITS Conference*, pages 975–982.
- Chiang, H.-H., Chen, Y.-L., Wu, B.-F., and Lee, T.-T. (2014). Embedded driver-assistance system using multiple sensors for safe overtaking maneuver. *IEEE Systems Journal*, 8(3):681–698.
- Chien, J.-C., Lee, J.-D., and Liu, L.-C. (2015). A fuzzy rules-based driver assistance system. *Mathematical Problems in Engineering*, 2015.
- Costea, A. D., Vesa, A. V., and Nedevschi, S. (2015). Fast pedestrian detection for mobile devices. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 2364–2369. IEEE.
- Dai, J.-M., Wu, L.-T., Lin, H.-Y., and Tai, W.-L. (2016). A driving assistance system with vision based vehicle detection techniques. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific*, pages 1–9. IEEE.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.
- Duran, K., Bernal, H., and Melgarejo, M. (2008). Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set. In *NAFIPS 2008-2008 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–5. IEEE.

- Gaikwad, V. and Lokhande, S. (2015). Lane departure identification for advanced driver assistance. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):910–918.
- Ghoddosian, R., Galib, M., and Athitsos, V. (2019). A realistic dataset and baseline temporal model for early drowsiness detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0.
- Gorzałczany, M. B. (1987). A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy sets and systems*, 21(1):1–17.
- Gruyer, D., Belaroussi, R., Li, X., Lusetti, B., Revilloud, M., and Glaser, S. (2015). Persee: A central sensors fusion electronic control unit for the development of perception-based adas. In *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*, pages 250–254. IEEE.
- Gu, Q., Yang, J., Zhai, Y., and Kong, L. (2015). Vision-based multi-scaled vehicle detection and distance relevant mix tracking for driver assistance system. *Optical Review*, 22(2):197–209.
- Gómez, J. A. (2014). Fusión de datos de sensores mediante Lógica Difusa, aplicación en robótica móvil. Master's thesis, CENIDET, México.
- Haghray, A. A. and Ghaemi, S. (2019). Pyit2fls: A new python toolkit for interval type 2 fuzzy logic systems.
- Hamid, U. Z. A., Zakuan, F. R. A., Zulkepli, K. A., Azmi, M. Z., Zamzuri, H., Rahman, M. A. A., and Zakaria, M. A. (2017). Autonomous emergency braking system with potential field risk assessment for frontal collision mitigation. In *Systems, Process and Control (ICSPC), 2017 IEEE Conference on*, pages 71–76. IEEE.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Huang, J., Li, R., An, J., Ntalasha, D., Yang, F., and Li, K. (2017). Energy-efficient resource utilization for heterogeneous embedded computing systems. *IEEE Transactions on Computers*, 66(9):1518–1531.
- Inchaurbe Chaterina, J. (2014). Detección de personas basada en hog y svm.
- Jerry, M. M. (2019). *UNCERTAIN RULE-BASED FUZZY SYSTEMS: Introduction and New Directions*. SPRINGER.
- JetsonHacks (2018). Nvidia jetson tx2 j21 header pinout.
- Karnik, N. N. and Mendel, J. M. (2001). Operations on type-2 fuzzy sets. *Fuzzy sets and systems*, 122(2):327–348.
- Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874.

- King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758.
- King, D. E. (2015). Max-margin object detection. *arXiv preprint arXiv:1502.00046*.
- Klette, R. and CITR, A. (2015). Vision-based driver assistance systems.
- Le, V., Brandt, J., Lin, Z., Bourdev, L., and Huang, T. S. (2012). Interactive facial feature localization. In *European conference on computer vision*, pages 679–692. Springer.
- Liang, Q. and Mendel, J. M. (2000). Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy systems*, 8(5):535–550.
- López, D. D. (2019). Extracción de Superpíxeles y Fusión Local Mediante un Algoritmo de Agrupamiento Difuso Intuicionista. Master's thesis, CENIDET, México.
- Manoharan, R. and Chandrakala, S. (2015). Android opencv based effective driver fatigue and distraction monitoring system. In *Computing and Communications Technologies (ICCCCT), 2015 International Conference on*, pages 262–266. IEEE.
- Mendel, J. M. (2001). *Uncertain rule-based fuzzy logic systems: introduction and new directions*. Prentice Hall PTR Upper Saddle River.
- Menzel, T., Bagschik, G., and Maurer, M. (2018). Scenarios for development, test and validation of automated vehicles. *arXiv preprint arXiv:1801.08598*.
- Nieto, M., Otaegui, O., Velez, G., Ortega, J. D., and Cortes, A. (2014). On creating vision-based advanced driver assistance systems. *IET intelligent transport systems*, 9(1):59–66.
- Nieto, M., Vélez, G., Otaegui, O., Gaines, S., and Van Cutsem, G. (2016). Optimising computer vision based adas: vehicle detection case study. *IET Intelligent Transport Systems*, 10(3):157–164.
- Nvidia, S. W. (2019). Nvidia/jetson-gpio.
- Ozaki, N., Uchiyama, M., Tanabe, Y., Miyazaki, S., Sawada, T., Tamai, T., and Banno, M. (2015). Implementation and evaluation of image recognition algorithm for an intelligent vehicle using heterogeneous multi-core soc. In *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, pages 410–415. IEEE.
- Palazzi, A., Borghi, G., Abati, D., Calderara, S., and Cucchiara, R. (2017). Learning to map vehicles into bird's eye view. In *International Conference on Image Analysis and Processing*, pages 233–243. Springer.
- Pattarkine, P. (2018). Driver drowsiness alert system with effective feature extraction. *IJREST*, 5(4):26–31.
- Petrovai, A., Danescu, R., and Nedevschi, S. (2015). A stereovision based approach for detecting and tracking lane and forward obstacles on mobile devices. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 634–641. IEEE.

- Ponce-Cruz, P., Molina, A., and MacCleery, B. (2016). *Fuzzy Logic Type 1 and Type 2 Based on LabVIEW™ FPGA*. Springer.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 397–403.
- Satzoda, R. K., Lee, S., Lu, F., and Trivedi, M. M. (2015). Snap-das: A vision-based driver assistance system on a snapdragon tm embedded platform. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 660–665. IEEE.
- Schwiegelshohn, F., Gierke, L., and Hübner, M. (2015). Fpga based traffic sign detection for automotive camera systems. In *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2015 10th International Symposium on*, pages 1–6. IEEE.
- Siordia, O. G. S. (2009). Modelado del Comportamiento de Conducción de Vehículos de Transporte y/o Carga Aplicando Redes Neuronales Artificiales y Visión Artificial. Master's thesis, CENIDET, México.
- Son, S. and Baek, Y. (2015). Design and implementation of real-time vehicular camera for driver assistance and traffic congestion estimation. *Sensors*, 15(8):20204–20231.
- Soukupova, T. and Cech, J. (2016). Eye blink detection using facial landmarks. In *21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia*.
- Varga, R., Vesa, A. V., Jeong, P., and Nedeveschi, S. (2014). Real-time pedestrian detection in urban scenarios. In *Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference on*, pages 113–118. IEEE.
- Velez, G. and Otaegui, O. (2016). Embedding vision-based advanced driver assistance systems: a survey. *IET Intelligent Transport Systems*, 11(3):103–112.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.
- Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.
- Yang, S., Luo, P., Loy, C. C., and Tang, X. (2016). Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yao, B., Hagrais, H., Al Ghazzawi, D., and Alhaddad, M. J. (2012). An interval type-2 fuzzy logic system for human silhouette extraction in dynamic environments. In *International Conference on Autonomous and Intelligent Systems*, pages 126–134. Springer.

Yao, B., Hagnas, H., Lepley, J. J., Peall, R., and Butler, M. (2016). An evolutionary optimization based interval type-2 fuzzy classification system for human behaviour recognition and summarisation. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 004706–004711. IEEE.

Zadeh, L. A. (1987). *Fuzzy sets and applications*. Number 511.32 F8.

Anexo A

Documentos correspondientes a participaciones



Figura A.1: Reconocimiento como ponente en la Escuela de Inteligencia Artificial y Robótica 2018, UTEZ.



Figura A.2: Reconocimiento como participante de poster en Primera Jornada de Ciencia y Tecnología Aplicada, CENIDET 2018.



Figura A.3: Reconocimiento como concursante en ICMEAE, 2018.

UDLAP[®]

Universidad de las Américas Puebla

by the Department of Computing,
Electronics & Mechatronics hereby award this certificate to

Arturo Rendón Castro

for the oral presentation of the technical article titled
**Improvement of MRI images through
heterogeneous interpolation techniques**
at the 29th International Conference on Electronics,
Communications and Computers, CONIELECOMP 2019



Gerardo Ayala San Martín, Ph.D.
Academic Director



Rocío Salazar Varas, Ph.D.
Conference Chair

February 27th, 2019, Puebla, Mexico.

 Scanned with
CamScanner

Figura A.4: Reconocimiento como co-autor en artículo publicado en CONIELECOMP, UDLAP 2019.



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

**EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO**

OTORGAN LA PRESENTE

CONSTANCIA

A

**Ángel Arturo Rendón Castro, Jesús Antonio Luna
Álvarez, Dante Mújica Vargas**

POR SU PARTICIPACIÓN CON EL ARTÍCULO: **"FUZZY CONTROL OF
STEWART PLATFORM WITH ARTIFICIAL VISION"**, PRESENTADO
EN LA SEGUNDA JORNADA DE CIENCIA Y TECNOLOGÍA
APLICADA, QUE SE LLEVO A CABO LOS DÍAS 4 Y 5 DE ABRIL DE
2019, EN LA CIUDAD DE CUERNAVACA, MORELOS, MÉXICO.

CUERNAVACA, MOR., A 5 DE ABRIL DE 2019.



S. E. P.
CENTRO NACIONAL DE INVESTIGACION
Y DESARROLLO TECNOLÓGICO

DR. VÍCTOR HUGO OLIVARES PEREGRINO
DIRECTOR



Figura A.5: Constancia de autor de artículo publicado en la Segunda Jornada de Ciencia y Tecnología Aplicada, CENIDET 2018.



**EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL INSTITUTO TECNOLÓGICO DE IGUALA**

OTORGA EL PRESENTE

RECONOCIMIENTO

A

ÁNGEL ARTURO RENDÓN CASTRO

POR SU DESTACADA PARTICIPACIÓN COMO PONENTE DE LA CONFERENCIA DENOMINADA "SISTEMA EMBEBIDO PARA ASISTENCIA DE CONDUCCIÓN BASADO EN LÓGICA DIFUSA TIPO-2" EN EL MARCO DEL 1er COLOQUIO "ARTIFICIAL INTELLIGENCE", REALIZADO EN EL INSTITUTO TECNOLÓGICO DE IGUALA.

IGUALA, GRO., A 9 DE MAYO DE 2019.

"TECNOLOGIA COMO SINÓNIMO DE INDEPENDENCIA"
M.D.I.S. ARELY BARCENAS NAVA
DIRECTORA



Figura A.6: Reconocimiento como ponente en el 1er coloquio "Artificial Intelligence", Tecnológico de Iguala 2019.

Sistema Embebido para Detección de Inatención en Tarea de Conducción

Ángel Arturo Rendón Castro¹, Dante Mújica Vargas¹, Andrea Magadán Salazar¹ y Andrés Blanco Ortega²

¹Depto. de Ciencias Computacionales, ²Depto. Ingeniería en Mecánica

Tecnológico Nacional de México / CENIDET

Cuernavaca, México

{angel.rendon18ce, dantemv, magadan, andres.blanco}@cenidet.edu.mx

Abstract

Un factor común de accidentes viales se debe a la inatención en la tarea de conducción (Somnolencia, distracción, etc.). Por lo tanto, áreas como el de Sistemas de Transporte Inteligentes se encuentra en continuo desarrollo para brindar mayor seguridad. Un ejemplo son los sistemas de asistencia que se enfocan en mejorar la seguridad de los ocupantes mediante la fusión de información de sensores que reconocen el entorno, procesando con métodos y algoritmos que detectan situaciones de riesgo que son atendidas con la activación de actuadores y/o recomendaciones al conductor. En este artículo se propone un sistema de asistencia que detecta el nivel inatención del conductor y muestra una serie de alertas. El sistema obtiene información mediante visión por computadora y realiza la inferencia con lógica difusa, a su vez el sistema está implementado en la plataforma embebida NVIDIA Jetson TX2. Los experimentos en tiempo real demuestran que el sistema propuesto es altamente eficiente para encontrar la somnolencia y alertar al conductor obteniendo una tasa de detección > 0.90 y de precisión > 0.88 .

Keywords— Fuzzy logic, Computer vision, embedded systems, Jetson tx2, inattention

1. Introducción

La inatención en la tarea de conducción puede ser generada por varios factores, tales como el estado de somnolencia del conductor y el enfoque de atención del conductor en otras actividades, factores que pueden conllevar desde una pérdida material hasta un accidente mortal. Con el objetivo de tratar de minimizar el error humano en las actividades de conducción debido a la distracción del conductor, actualmente se implementan sistemas de asistencia de conducción para brindar herramientas de detección de obstáculos y monitoreo del conductor de forma óptima y en tiempo real, de tal forma que el conductor pueda ser alertado de posibles situaciones de riesgo que representen un peligro para él y los pasajeros del vehículo.

Existen diferentes tipos de asistentes de conducción, entre los que destacan los basados en información interna del vehículo, signos vitales y externos. Los primeros se basan en información recolectada de mecanismos del vehículo (velocidad, aceleración, frecuencia de frenado, etc.) y tienen cierto grado de control sobre algunos actuadores del vehículo. Usualmente están dotados de un nivel alto e intermedio de autonomía, pero su desventaja radica en el costo elevado de implementación y investigación protegida por patentes. Los basados en signos vitales (Electrocardiogramas, electroencefalogramas, presión arterial, frecuencia cardíaca, etc.) se enfocan en monitorear el estado de salud del conductor. Un ejemplo de este tipo se puede encontrar en los trabajos de (Kartsch, Benatti, Schiavone, Rossi, & Benini, 2018), (Li & Chung, 2015), (L. Yang, Ma, Zhang, Guan, & Jiang, 2018), donde se hace uso de dispositivos en forma de diadema que toman electroencefalogramas que registran la actividad bioeléctrica cerebral en condiciones basales de reposo o somnolencia, aunque con una tasa aceptable de detección suelen ser invasivos y restan confort a la tarea de conducción. Por último los basados en parámetros externos, estos utilizan información producto de una fusión de datos obtenidos de diversos sensores. Suelen ser muy flexibles en cuanto diseño e implementación, realizados para diversos propósitos como el de monitorear obstáculos en el camino, como en los trabajos de (Godha, 2017), (Yoo, Son, Ham, & Sohn, 2016), que utilizan métodos basados en visión para la alerta de objetos en el camino.

En este artículo se propone un sistema de asistencia para la detección de inatención en la tarea de conducción, implementado en la Tarjeta NVIDIA Jetson TX2 con un sistema de adquisición basado en visión por computadora y un controlador difuso. Este sistema está orientado a monitorear y asistir al conductor mediante una serie de alarmas que permitan notificar de una posible situación de riesgo causada por la inatención. El resto del artículo se compone de las siguientes secciones: Sección II, Marco Teórico. Descripción del sistema en sección III. Resultados en sección IV y en V se encuentran las conclusiones y trabajo futuro.

2. Marco Teórico

2.1. Detección facial de puntos de referencia

Algoritmo introducido en (Kazemi & Sullivan, 2014), donde por medio de un conjunto de árboles de regresión se realiza la alineación de 68 puntos de referencia en cuestión de milisegundos.

Figura A.7: Artículo enviado a la Escuela de Inteligencia Artificial y Robótica, UTEZ 2019.