



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**  
**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**



**TÍTULO DE LA TESIS:**

**ALGORITMO DE VISIÓN ARTIFICIAL PARA  
DETECCIÓN DE REDUCTORES DE VELOCIDAD EN  
SUPERFICIES VIALES**

**TESIS PRESENTADA PARA OBTENER EL GRADO DE:**

**DOCTORA EN INGENIERÍA DEL LENGUAJE Y DEL  
CONOCIMIENTO**

**PRESENTA:**

Ana Luisa Ballinas Hernández

analuisa.ballinas@correo.buap.mx

**DIRECTOR DE TESIS:**

Dr. Ivan Olmos Pineda

iolmos@cs.buap.mx

**Puebla, Puebla; enero 2022**

Resumen. Uno de los desafíos actuales para la conducción autónoma es la detección temprana de irregularidades en las calles para evitar daños o accidentes vehiculares; en particular, la detección de reductores de velocidad es una tarea crucial para una navegación segura. En países en vías de desarrollo es muy común encontrar reductores de velocidad sin señalamiento lo que dificulta su detección. Las técnicas existentes no han logrado resolver esta tarea al 100% ya que algunos de ellos no funcionan en entornos a escala real o presentan errores altos. En este trabajo se propone una metodología para la detección de reductores de velocidad tanto señalizados como sin señalamiento. Para los reductores de velocidad señalizados se entrena un modelo aplicando máquinas de vectores de soporte a un conjunto de imágenes 2D, donde se extraen histogramas de patrones binarios locales para reconocer los patrones de señalamiento a partir de un conjunto de imágenes y se obtiene una exactitud del 89%. Para la detección de reductores sin señalamiento se aplica visión estéreo para reconstruir escenas 3D de calles que son convertidas en mallas triangulares aplicando triangulación de Delaunay. Se entrena un modelo aplicando máquinas de soporte vectorial donde se extraen características basadas en valores propios a partir de las mallas de superficies y se obtiene una exactitud del 78%. A pesar de que los resultados presentan un porcentaje de error alto con respecto a los resultados reportados en la literatura, se consideran distintas condiciones como: entornos a escala real y diferentes materiales de las calles, por lo cual mejora las técnicas existentes.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento de la investigación	2
1.1.1. Definición del problema	2
1.1.2. Objetivos de la investigación	3
1.1.3. Justificación de la investigación	4
1.1.4. Preguntas de Investigación	5
1.2. Alcances y Limitaciones	5
1.3. Aportaciones de la investigación	5
1.4. Organización de la tesis	6
<b>2. Marco Teórico</b>	<b>9</b>
2.1. Procesamiento digital de imágenes	9
2.1.1. Filtro bilateral	9
2.1.2. Filtros de Gabor	10
2.1.3. Ecuilización del histograma	10
2.1.4. Umbralización	11
2.1.5. Operadores morfológicos	11
2.2. Visión estéreo	11
2.2.1. Calibración de cámaras	12
2.2.2. Rectificación	13
2.2.3. Mapa de disparidad	14
2.2.4. Nubes de puntos	15
2.2.5. Reconstrucción de superficies 3D	16
2.2.6. Procesamiento de mallas triangulares	16
2.3. Proceso de extracción de conocimiento KDD	17
2.3.1. Extracción de características	19
2.3.2. Algoritmos del aprendizaje supervisado	19

2.3.3. Aprendizaje profundo	21
2.4. Síntesis	23
<b>3. Estado del Arte</b>	<b>25</b>
3.1. Taxonomía de técnicas para detección de RV	25
3.1.1. Monitoreo por vibraciones usando sensores de smartphone	25
3.1.2. Técnicas basadas en imágenes	27
3.1.3. Sensores LIDAR	28
3.1.4. Visión estéreo	29
3.1.5. Aprendizaje profundo	30
3.1.6. Técnicas híbridas	30
3.2. Comparación de resultados de algunos trabajos relacionados	31
3.3. Segmentación de calles	34
3.4. Síntesis	35
<b>4. Metodología</b>	<b>37</b>
4.1. Captura de imágenes estéreo	37
4.2. Segmentación de calles con FCN	38
4.3. Preprocesamiento de imágenes	38
4.4. Detección de RV señalizados	39
4.5. Generación de mapas de disparidad	39
4.6. Generación de nubes de puntos	40
4.7. Creación de mallas de superficies	41
4.8. Procesamiento de mallas 3D	41
4.8.1. Extracción de Características	42
4.8.2. Detección de RV sin señalamientos	42
4.8.3. Evaluación del modelo	42
4.9. Algoritmo	43
4.10. Síntesis	46
<b>5. Resultados Obtenidos</b>	<b>49</b>
5.1. Captura de imágenes estéreo	49
5.2. Segmentación de calles con FCN	51
5.3. Preprocesamiento de imágenes	54
5.4. Detección de RV señalizados	57
5.5. Generación de mapas de disparidad	58
5.6. Generación de nubes de puntos	59

5.7. Creación de mallas de superficies 3D	60
5.8. Procesamiento de mallas 3D	60
5.9. Extracción de características	62
5.10. Detección de RV sin señalamientos	62
5.11. Discusión	63
5.12. Síntesis	68
<b>6. Conclusiones</b>	<b>69</b>
<b>Referencias</b>	<b>70</b>



# Índice de figuras

1.1. Reductor de velocidad sin señalamientos bien marcados	3
2.1. Proceso para la construcción de imágenes estéreo	12
2.2. Geometría epipolar del proceso de análisis de correspondencias	14
2.3. Proceso de triangulación de la visión estéreo	15
2.4. Funcionamiento del algoritmo de muestreo de disco de Poisson	16
2.5. Fases del proceso KDD	17
2.6. Funcionamiento de una red neuronal artificial	20
2.7. Modelo de una máquina de vectores de soporte	22
2.8. Funcionamiento de la FCN para la segmentación semántica	23
3.1. Imagen de reductor de velocidad y su segmentación	27
3.2. Tasa de detección de RV reportada por algunos autores	32
4.1. Metodología propuesta para la detección de reductores de velocidad	37
5.1. Sistema de visión estéreo	49
5.2. Imágenes de vialidades capturadas con el sistema de visión estéreo	51
5.3. Base de datos de vialidades capturadas con el sistema de visión	52
5.4. Eliminación de región de no interés en imágenes estéreo	53
5.5. Comparación entre la segmentación verdadera y la predicha	54
5.6. Imágenes de vialidades con y sin reductores de velocidad	55
5.7. Resultados del procesamiento de imágenes	56
5.8. Resultado de la detección de RV señalizados	57
5.9. Curva ROC del modelo de clasificación de RV señalizados	58
5.10. Mapas de disparidad generados	59
5.11. Reconstrucción 3D de nubes de puntos	60
5.12. Mallas triangulares de superficies 3D	60
5.13. Decimación de mallas 3D conservando su topología	61



5.14. Suavizado laplaciano de mallas de superficies 3D. . . . .	61
5.15. Histogramas de características de mallas 3D. . . . .	62
5.16. Resultado de la clasificación de RV sin señalamiento. . . . .	63
5.17. Curva ROC del modelo de clasificación de RV sin señalamientos. . . . .	64
5.18. Tiempos de ejecución promedio. . . . .	64

# Índice de cuadros

5.1. Métricas de evaluación del modelo RV con señalamientos . . . . .	58
5.2. Rangos de valores de las nubes de puntos en metros. . . . .	60
5.3. Selección de características basadas en valores propios para mallas 3D . . . . .	62
5.4. Métricas de evaluación del modelo RV sin señalamientos . . . . .	63
5.5. Tiempo de detención mínimo de RV para distintas velocidades . . . . .	66
5.6. Comparación de la detección de RV señalizados con otros enfoques . . . . .	67
5.7. Comparación de la detección de RV sin señalamientos con otros enfoques . . . . .	68



# Capítulo 1

## Introducción

Debido al auge de las nuevas tecnologías ha sido posible construir autos autónomos capaces de navegar y realizar algunas tareas sin la intervención de un conductor humano. El desarrollo de estas tecnologías ha ido en aumento y ha sido un tema de amplio interés por la comunidad científica ya que estos autos serán, en un futuro próximo, el principal medio de transporte terrestre. En países como: Alemania, España, Japón, Estados Unidos, Singapur, Suecia, China, entre otros, han empezado a circular prototipos de autos autónomos, sin embargo, no han logrado ser totalmente autónomos debido a restricciones en el entorno. Algunos países se encuentran preparando la infraestructura y legislación necesaria para que estos autos circulen debido a que mundialmente no existe una regulación estándar.

En las últimas décadas la tendencia de la industria automotriz se ha centrado en el desarrollo de autos autónomos. La mayoría de los nuevos modelos de autos que se venden en el mercado incluyen ya funciones autónomas de conducción, un ejemplo es la función que permite que el auto se estacione solo. Al no existir conductor, el auto deberá aplicar algoritmos de la Inteligencia Artificial para la toma de decisiones en las tareas que realice. En los últimos años, empresas como Tesla Motors, Google, Audi, Volkswagen, General Motors, Volvo, Nissan, Mercedes Benz, BMW, Ford, Toyota, Renault, entre otras, han ido incorporando tecnologías para la conducción autónoma e incluso han construido prototipos de autos totalmente autónomos.

Un antecedente importante han sido las competiciones de autos autónomos que se han llevado a cabo como: el *DARPA Grand Challenge* (del 2004 al año 2006) y el *DARPA Urban Challenge* (a partir del 2007 a la fecha). En estas competencias los autos deben partir de una ubicación origen hasta llegar a un destino en algún punto sin un conductor humano, recorriendo el tráfico de la ciudad y otros ambientes extremos [Buehler et al.](#)

(2009). En general, los autos participantes han presentado buenos resultados en la navegación a pesar de la complejidad de los escenarios y de los inconvenientes que deben resolver. La solución a estos inconvenientes ha logrado avances importantes en el desarrollo de autos autónomos.

## 1.1. Planteamiento de la investigación

Las principales tareas que realizan los autos autónomos para su navegación son: percepción del entorno, localización, planeación de rutas y control del movimiento electromecánico. La percepción del ambiente es una tarea crucial para el diseño de autos autónomos además de ser el foco de interés de esta investigación. Esta tarea se puede realizar mediante el uso de diversos sensores; en este trabajo se emplean videocámaras para la adquisición de datos. Diversos trabajos han estudiado la identificación de obstáculos, señales de tránsito e irregularidades en las superficies de rodamiento. Con los datos adquiridos del ambiente, los autos han aplicado técnicas de visión por computadora para resolver conflictos de tráfico por sí mismos, sin embargo, algunas tareas específicas no han sido completamente resueltas con las técnicas existentes.

### 1.1.1. Definición del problema

Frecuentemente, las vialidades de países en vías de desarrollo presentan reductores de velocidad que no siguen adecuadamente la norma de tránsito establecida para señalizaciones horizontales debido a que las marcas no son claramente visibles por el desgaste del tiempo y por la falta de mantenimiento (ver Figura 1.1). Esta situación afecta la seguridad y estabilidad de la conducción autónoma debido a que si no se detectan a tiempo pueden causar daños importantes en los vehículos y en los pasajeros.

Las técnicas actuales para el análisis de superficies en vialidades no han abordado a fondo la detección de reductores de velocidad cuando la señalización vial no está bien marcada. Los trabajos existentes han aplicado técnicas del procesamiento digital de imágenes para identificar los patrones de señalamiento, sin embargo, cuando estos obstáculos no tienen un señalamiento bien definido, el error en la detección es muy alto. Por lo tanto, en este trabajo se aplican técnicas de visión estéreo y aprendizaje automático para identificar automáticamente reductores de velocidad tanto señalizados como sin señalamiento en el piso.



Figura 1.1: Reductor de velocidad sin señalamientos bien marcados.

### **1.1.2. Objetivos de la investigación**

El objetivo general y los objetivos específicos se presentan a continuación.

#### **Objetivo general**

El objetivo general es diseñar un algoritmo que detecte reductores de velocidad con y sin señalamientos en superficies de rodamiento vial mediante la aplicación de técnicas de visión estéreo y de aprendizaje automático para la toma de decisiones en la conducción autónoma.

#### **Objetivos específicos**

Los objetivos específicos del presente trabajo se listan a continuación:

1. Construir una base de datos de imágenes estéreo para reconstruir superficies de vialidades a partir de escenas tridimensionales.
2. Aplicar algoritmos de segmentación de calles para determinar la región de interés en las imágenes mediante modelos de aprendizaje profundo.
3. Optimizar la reconstrucción de superficies generada por nubes de puntos 3D mediante la aplicación de métodos de análisis de mallas de superficies triangulares.
4. Implementar algoritmos de extracción de características en imágenes estéreo que permitan inferir la presencia de reductores de velocidad por análisis de superficies 3D.

5. Generar modelos de clasificación de imágenes con reductores de velocidad tanto señalizados como no señalizados mediante la aplicación de algoritmos de aprendizaje automático.

### **1.1.3. Justificación de la investigación**

Los prototipos de autos autónomos existentes han logrado circular en ambientes controlados bajo ciertas restricciones donde a través de sus sensores detectan las señalizaciones existentes en las vialidades. La circulación de autos autónomos en países en vías de desarrollo no ha sido posible debido a que los escenarios no tienen las condiciones adecuadas y se presenta un alto grado de incertidumbre. En las zonas de tránsito urbanas de estos países existen factores que afectan la circulación y hacen que la conducción tenga errores debido a los diversos obstáculos que se encuentran, tales como: semáforos, cruceros, señales de tránsito, peatones, otros vehículos, ciclistas, entre otros, más aún, se presentan irregularidades y elevaciones en las superficies de rodamiento vial. La detección de obstáculos por elevación en superficies es un aspecto relevante para que la conducción autónoma sea segura y estable.

En México existe la Norma NOM-034-SCT2-2011 que establece los lineamientos de señalización horizontal (sobre el piso) y vertical (sobre postes) de carreteras y vialidades urbanas para reductores de velocidad (Norma Oficial Mexicana NOM-034-SCT2-2011, 2011). Los reductores de velocidad se consideran como irregularidades en la superficie de rodamiento que obligan a los vehículos a reducir su velocidad en zonas de conflicto. En esta norma se establece que los reductores de velocidad deben pintarse con franjas diagonales de colores negro y amarillo alternadamente. Existen diversos trabajos de procesamiento digital de imágenes que analizan el patrón de estas señalizaciones para la detección automática de reductores de velocidad. Sin embargo, en países en vías de desarrollo no es posible aplicar directamente esta técnica debido a que a los reductores de velocidad no se les da el adecuado mantenimiento y es muy común encontrar reductores de velocidad sin una señalización vial clara, por lo que se dificulta su detección.

En este trabajo se considera el uso de videocámaras debido a que son sensores digitales de bajo costo, con respecto a otros sensores usados para los autos y a que si se aplican las técnicas adecuadas pueden ser lo suficientemente precisas. Además, se aplican técnicas de visión artificial para analizar superficies de rodamiento debido a que se desean detectar reductores de velocidad por la elevación en las superficies de rodamiento y no por su señalización vial.

---

#### **1.1.4. Preguntas de Investigación**

A partir de la definición del problema se plantean las siguientes preguntas de investigación:

- ¿Cuáles técnicas de visión artificial pueden servir para determinar la presencia de elevaciones en las superficies de rodamiento en vialidades?
- ¿Cuáles elementos de la visión por computadora son necesarios para diseñar un algoritmo que pueda detectar reductores de velocidad con y sin señalamientos?

### **1.2. Alcances y Limitaciones**

En este trabajo se diseña un algoritmo que detecta reductores de velocidad con o sin señalamientos marcados con dimensiones grandes debido a que son los que más daño pueden ocasionar aplicando técnicas de análisis de elevaciones en las superficies. Las entradas del algoritmo son imágenes estéreo de calles del estado de Puebla, en zonas urbanas y semiurbanas, que son capturadas usando un sistema de visión estéreo. La salida del algoritmo es la clasificación de imágenes que indique si en la imagen se detecta o no se detecta algún reductor de velocidad.

Una restricción es que la profundidad abarcada por la visión estéreo es de corto alcance debido a que la reconstrucción de escenas 3D no es muy exacta en distancias largas. Además, la captura de imágenes estéreo se realiza bajo condiciones controladas del ambiente considerando un clima adecuado con alta luz solar para tener una buena iluminación y exposición. Para que el algoritmo funcione adecuadamente se consideran imágenes sin sombras considerables, ocasionadas por construcciones o vegetación, y preferentemente se consideran imágenes de calles sin obstáculos que impidan la detección de los reductores de velocidad. Por lo tanto, la precisión del algoritmo se ve afectada por estos factores y por la distancia a la que se capturan los RV. El algoritmo no se prueba en tiempo real debido al alto costo computacional ocasionado por la reconstrucción estéreo de las superficies. Además, el algoritmo únicamente será capaz de detectar RV en las superficies de rodamiento pero no podrá identificar a detalle el tipo de reductor detectado.

### **1.3. Aportaciones de la investigación**

Las principales aportaciones de esta investigación son las siguientes:



- Construcción de un algoritmo de aprendizaje automático para la detección de reductores de velocidad (con forma ovalada y trapezoidal) por elevación en las superficies sin la condición de que las señalizaciones en las vialidades sean claras.
- Implementación de métodos para el análisis de mallas de superficies tridimensionales que ayuden a detectar elevaciones significativas más fácilmente y a discriminar aquellas elevaciones poco relevantes para la detección de reductores de velocidad.

## 1.4. Organización de la tesis

En esta tesis se diseña un algoritmo para detectar reductores de velocidad tanto señalizados como no señalizados en superficies de rodamiento vial aplicando técnicas de visión artificial. La tesis se encuentra organizada de la siguiente forma:

En el Capítulo 2 se describen los fundamentos teóricos relacionados con el problema a resolver. Los principales temas son: conducción autónoma, visión estéreo, métodos de segmentación de áreas de interés en imágenes 2D y 3D, métodos de extracción automática de características en imágenes 2D y 3D, análisis y estimación de superficies 3D, así como el proceso de descubrimiento de conocimiento.

En el Capítulo 3 se realiza el análisis del estado del arte. Se realiza una clasificación de trabajos relacionados con la detección de reductores de velocidad y análisis de elevaciones en superficies de rodamiento en vialidades. Las principales técnicas son: monitoreo de superficies por aceleraciones, procesamiento digital de imágenes, sensores LIDAR, visión estéreo, técnicas híbridas que combinan las anteriores y aprendizaje profundo. Además, se presenta una gráfica con la tasa de detección de reductores de velocidad reportada en algunos trabajos.

En el Capítulo 4 se describe la propuesta metodológica para dar solución a la detección automática de RV señalizados y no señalizados mediante visión estéreo y aprendizaje automático supervisado. Las fases son las siguientes: captura de imágenes estéreo, preprocesamiento de imágenes, detección de RV por patrón de señalamiento, segmentación de calles, generación de mapas de disparidad, generación de nubes de puntos, creación de mallas de superficies 3D, procesamiento de mallas, extracción de características 3D, construcción de un modelo de clasificación supervisada para detectar RV por elevación en superficies y evaluación de los modelos.

En el Capítulo 5 se presentan los experimentos y resultados más relevantes obtenidos de acuerdo con la metodología propuesta. Además, se muestran gráficas comparativas de

---

los resultados del algoritmo propuesto con otros algoritmos para evaluar su desempeño. Finalmente, en el Capítulo 6 se presentan las conclusiones de esta investigación, así como los posibles trabajos a futuro que se derivan de esta tesis.



# Capítulo 2

## Marco Teórico

En este capítulo se presentan los conceptos relacionados con el procesamiento digital de imágenes aplicado en este trabajo, las fases del proceso de visión estéreo, la generación de nubes de puntos 3D y la reconstrucción de mallas de superficies 3D; además, se describen algunos algoritmos del aprendizaje automático empleados para la detección de RV.

### 2.1. Procesamiento digital de imágenes

Para el procesamiento digital de imágenes se aplican varias técnicas que ayudan a mejorar la calidad de las imágenes digitales y facilitar el reconocimiento de patrones que se pueden obtener de ellas. Algunas técnicas de procesamiento de imágenes empleadas para el desarrollo de este trabajo se presentan a continuación.

#### 2.1.1. Filtro bilateral

El filtro bilateral es un filtro no lineal que se encarga de conservar los bordes al reducir ruido y suavizar imágenes. El valor de la intensidad en cada pixel de la imagen es reemplazado por una media ponderada de la intensidad de pixeles vecinos. El filtro bilateral  $\hat{I}$  de la imagen se define de acuerdo con la Ec. [2.1](#).

$$\hat{I}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|), \quad (2.1)$$

donde  $W_p$  es el peso que se asigna mediante la cercanía espacial de píxeles y la diferencia de intensidad. Este término es calculado de acuerdo con la Ec. 2.2.

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|)g_s(\|x_i - x\|) \quad (2.2)$$

$I$  es la imagen original,  $x$  son las coordenadas del píxel a filtrar,  $\Omega$  es el centro de la ventana  $x$ ,  $f_r$  es el núcleo gama para suavizar las diferencias en las intensidades y  $g_s$  es el núcleo espacial para suavizar las diferencias de coordenadas (éstas dos últimas pueden ser una función gaussiana).

### 2.1.2. Filtros de Gabor

Son filtros lineales pasa banda que son una función sinusoidal multiplicada por una función gaussiana. Estas funciones están localizadas tanto en el dominio espacial como en el dominio frecuencial con tratamiento del rango de frecuencia espacial radial y orientaciones múltiples Mery (2015). Las funciones de Gabor están dadas por la Ecuación 2.1.2.

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right) \quad (2.3)$$

where  $\sigma_x$  y  $\sigma_y$  denota la envolvente gaussiana a lo largo de los ejes  $x$  e  $y$ , y  $u_0$  define la frecuencia radial de la función de Gabor.

### 2.1.3. Ecuación del histograma

Una técnica muy utilizada para realizar un ajuste de los niveles de contraste es la ecualización de histogramas Nixon & Aguado (2012). La técnica modifica la estructura de una imagen de tal manera que se generan histogramas aproximados a una distribución uniforme. Este método utiliza distintos esquemas de distribución de intensidades de color para transformarse en una imagen con una distribución uniforme de los valores de gris. Existen algunos métodos para ecualizar el histograma de una imagen pero uno de los que causan menor distorsión en la imagen es el histograma adaptativo con restricción de contraste (CLAHE). El método CLAHE divide la imagen en múltiples subregiones y clasifica el histograma de cada región por separado para obtener los nuevos valores mediante la interpolación de cada píxel.

---

### 2.1.4. Umbralización

El umbral de una imagen es un valor que separa los valores de píxeles en dos clases para generar una imagen en blanco y negro. La función de un umbral  $f_{umbral}(a)$  asigna todos los píxeles a uno de los valores de intensidad fijo que puede ser 0 o 1. Una de las técnicas más exactas para la binarización de imágenes es la umbralización Otsu que obtiene el valor umbral de forma que la dispersión dentro de cada segmento sea lo más pequeña posible pero que la dispersión sea lo más alta posible entre distintos segmentos, calculando el cociente entre ambas varianzas

### 2.1.5. Operadores morfológicos

Las operaciones morfológicas son comúnmente empleadas para el tratamiento de estructuras geométricas en imágenes. Las operaciones básicas son: erosión y dilatación [Burger & Burge \(2010\)](#). En una imagen binaria el elemento estructurante  $H$  contiene solo valores de 0 y 1, es decir,  $H(i, j) \in \{0, 1\}$ . La erosión binaria de una imagen binaria  $I$  por el elemento estructurante  $H$  está definida por la Ecuación [2.4](#).

$$I \ominus H = \{p \in \mathbb{Z}^2 | H_p \subset I\} \quad (2.4)$$

donde  $H_p$  es un subconjunto de  $I$  en la posición  $p$ .

La dilatación se describe como el crecimiento de píxeles en la imagen alrededor de la circunferencia de cada región para incrementar su dimensión. La dilatación de  $I$  por el elemento estructurante  $H$  se define de acuerdo con la Ecuación [2.5](#).

$$I \oplus H = \{(p + q) | \forall p \in I, q \in H\} \quad (2.5)$$

## 2.2. Visión estéreo

La visión estéreo es una técnica que trata de reconstruir ambientes tridimensionales a partir de imágenes en dos dimensiones [Hamzah & Ibrahim \(2016\)](#). Un sistema de visión estéreo utiliza dos o más cámaras colocadas vertical u horizontalmente separadas a una cierta distancia. De esta forma se capturan imágenes simultáneas que son empleadas para recuperar información de profundidad de la escena. El proceso general se muestra en la Figura [2.1](#) y se detalla a continuación.

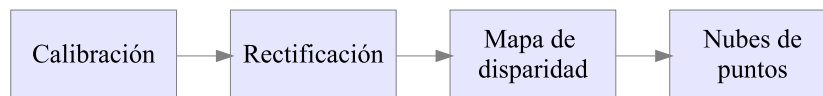


Figura 2.1: Proceso para la construcción de imágenes estéreo.

Adaptado de: Hamzah & Ibrahim (2016).

### 2.2.1. Calibración de cámaras

Debido a las imperfecciones de las cámaras se pueden producir imágenes distorsionadas. La calibración de cada cámara individual consiste en hacer una estimación de los parámetros intrínsecos y extrínsecos de la misma así como de las imperfecciones en las formas de los lentes que provocan desviaciones laterales en los haces de luz de la Escalera et al. (2010). La primera se manifiesta por una forma de efecto de barril y la segunda ocurre debido a que los lentes de la cámara no son perfectamente paralelos a la imagen plana. Estos parámetros se describen a continuación:

- **Parámetros intrínsecos:** modelan la geometría interna de la cámara y las características ópticas del sensor óptico. Los valores a calcular son: las distancias focales horizontal y vertical  $(f_x, f_y)$ , las coordenadas del centro óptico  $(c_x, c_y)$  así como las distorsiones del lente: radial  $\mathbf{K} = (k_1, k_2)$  y longitudinal o tangencial  $\mathbf{P} = (p_1, p_2, p_3)$ . La distorsión radial provoca que las líneas rectas de las imágenes aparezcan curvas, mientras que la distorsión tangencial provoca que la toma de imágenes no esté perfectamente paralela al plano de la imagen.
- **Parámetros extrínsecos:** miden la colocación física de la cámara, es decir, su posición y orientación respecto al sistema de coordenadas establecido para la escena. Estos parámetros dan la relación respecto al sistema de coordenadas del usuario en lugar del sistema de coordenadas de la cámara. Se considera que hay una traslación  $\mathbf{t}$  seguida de una rotación  $\mathbf{R}$  de la cámara.

Los parámetros extrínsecos y la proyección en la imagen  $\mathbf{U}$  de un punto en el espacio  $\mathbf{X}$  se pueden representar de acuerdo con las Ecs. 2.6 y 2.7.

$$\mathbf{U} = \mathbf{P}\mathbf{X} = \mathbf{K}(\mathbf{R}|\mathbf{T})\mathbf{X} \quad (2.6)$$

$$\begin{pmatrix} c_1 \\ c_2 \\ z \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.7)$$

Respecto a las distorsiones de las imágenes se aplican las Ecs. 2.8 y 2.9 de cada eje correspondiente para corregirlas:

$$x_r = x_i + x_i(k_1r^2 + k_2r^4) + 2p_1x_iy_i + p_2(r^2 + 2x_i^2) \quad (2.8)$$

$$y_r = y_i + y_i(k_1r^2 + k_2r^4) + p_1(r^2 + 2y_i^2) + 2p_2x_iy_i \quad (2.9)$$

donde  $(x_r, y_r)$  son las coordenadas reales de los pixeles,  $(x_i, y_i)$  las coordenadas ideales y  $r$  la distancia al centro óptico de la imagen.

La calibración estéreo consiste en encontrar la relación geométrica entre dos cámaras en el espacio. Es decir, corresponde a la estimación de la matriz de rotación  $\mathbf{R}$  y el vector de traslación  $\mathbf{t}$  que relacionan ambas cámaras. La matriz de rotación se calcula de acuerdo con la Ec. 2.10 y el vector de rotación se calcula según la Ec. 2.11

$$\mathbf{R} = \mathbf{R}_0\mathbf{R}_1^t \quad (2.10)$$

$$\mathbf{t} = \mathbf{t}_0 - \mathbf{R}\mathbf{t}_1 \quad (2.11)$$

donde  $\mathbf{R}_0$ ,  $\mathbf{R}_1$ ,  $\mathbf{t}_0$  y  $\mathbf{t}_1$  son la matriz de rotación y vector de rotación correspondientes a cada imagen.

### 2.2.2. Rectificación

Una vez que las cámaras han sido calibradas se pueden corregir las distorsiones en las imágenes capturadas. Sin embargo, puede ser difícil mantener la alineación de las imágenes incluso con equipos de alta precisión. Para corregir estos errores se aplica un proceso de rectificación realizando un emparejamiento de puntos mediante búsqueda de correspondencias. Este proceso se encarga de transformar una pareja de sistemas ópticos en otras dos que sean colineales y donde las líneas epipolares sean paralelas a los ejes de las imágenes. La Figura 2.2 muestra dos epipolos, dos líneas epipolares y un plano epipolar para cada imagen.  $\mathbf{O}_I$  y  $\mathbf{O}_R$  son los centros ópticos de las dos cámaras.  $\mathbf{p}_I$  y  $\mathbf{p}_D$  son las proyecciones del punto  $\mathbf{P}$  dentro de los planos de la imagen.  $\mathbf{EP}_I$  y  $\mathbf{EP}_D$  denotan los epipolos.

Se consideran algunas restricciones epipolares para encontrar la correspondencia entre puntos de una imagen Klette et al. (2001). Un punto  $\mathbf{P}_I$  en la imagen izquierda puede corresponder a un solo punto en la imagen derecha que se encuentra en correspondencia con la línea epipolar en la imagen derecha definida por  $\mathbf{P}_I$ . Otra restricción es que con el uso de la geometría estéreo estándar las líneas epipolares en los dos planos de las imágenes coinciden con el renglón horizontal de las imágenes.



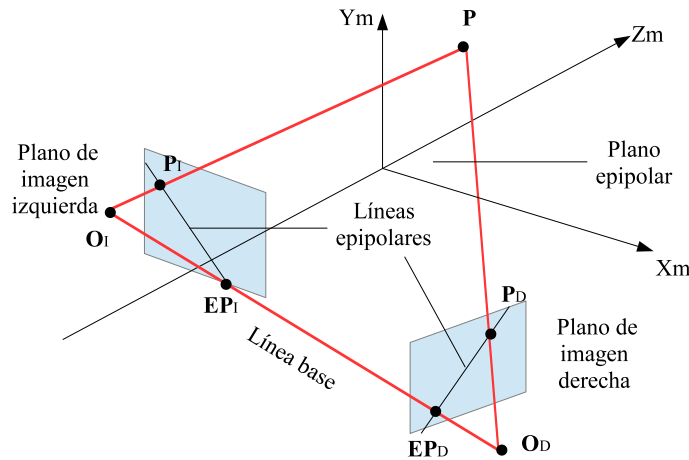


Figura 2.2: Geometría epipolar del proceso de análisis de correspondencias.

Fuente: Klette et al. (2001).

La rectificación se logra cuando las líneas epipolares en una imagen coinciden con las líneas de la otra imagen.

Para encontrar las líneas epipolares y los epipolos será necesario calcular la matriz esencial y la matriz fundamental. La matriz esencial  $E$  contiene información sobre la traslación y rotación que relaciona a ambas cámaras en el espacio físico (Klette et al. (2001)). Se calcula de acuerdo con la ecuación  $E = R[t]_x$  donde  $t$  es el vector de traslación y  $R$  la matriz de rotación. La matriz fundamental  $F$  contiene la misma información que  $E$  pero incorpora los parámetros intrínsecos que permiten relacionar ambas cámaras en coordenadas de píxeles. La matriz fundamental sigue la expresión  $F = (M_1^{-1})^t E M_0^{-1}$  donde  $M_0$  y  $M_1$  son las matrices de cámara derecha e izquierda respectivamente. El cálculo de estas matrices requiere de un sistema correctamente calibrado.

### 2.2.3. Mapa de disparidad

Para calcular la disparidad entre imágenes se consideran dos cámaras de características similares donde cada una de ellas cumple el modelo *pinhole* y donde los ejes ópticos de las cámaras son paralelos. La disparidad es la diferencia de coordenadas horizontales de cada par de puntos  $(x_1, x_2)$  la cual es calculada según la Ec. 2.12.

$$disparidad(x_1, x_2) = x_1 - x_2 = \frac{bf}{z} \quad (2.12)$$

donde  $f$  es la distancia focal y  $b$  es la línea base o distancia entre cámaras, asumiendo

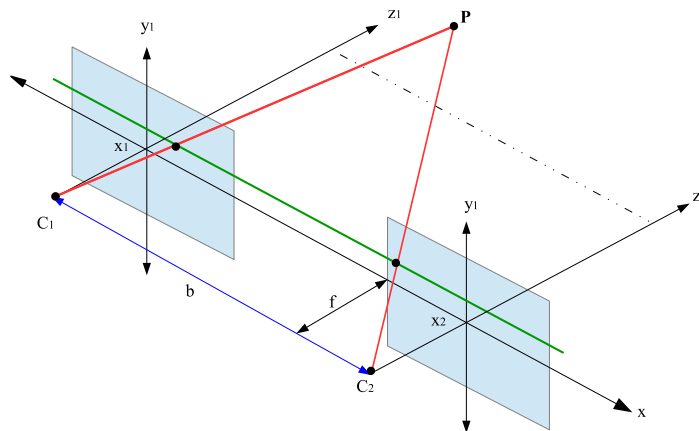


Figura 2.3: Proceso de triangulación de la visión estereoa.

Adaptado de: Klette et al. (2001).

que la coordenada  $y$  es fija, es decir,  $y = y_1 = y_2$ . Los valores de la disparidad de los puntos de las imágenes permiten obtener las coordenadas de los puntos proyectados  $\mathbf{P}$  en el espacio tridimensional donde los parámetros  $f$  y  $b$  se asumen conocidos. Este proceso se muestra en la Figura 2.3.

El cálculo de cada par de puntos del mapa de disparidad genera un mapa de profundidad que permite reconstruir superficies en un espacio 3D que pueden ser vistos desde distintos ángulos. En un mapa de profundidad cada punto  $\mathbf{P}$  en la imagen tiene asociada la distancia a la que se encuentra con respecto a la cámara. La profundidad de un punto en la escena es inversamente proporcional a la diferencia en distancia de los puntos de imagen correspondientes y sus centros de la cámara. A partir de imágenes rectificadas y de un mapa de disparidad se puede generar una nube de puntos 3D.

## 2.2.4. Nubes de puntos

Una nube de puntos es un conjunto de vértices en el espacio 3D sin una estructura regular que relaciona las coordenadas  $(X, Y, Z)$  que representan las posiciones de una escena tridimensional Cyganek & Siebert (2011). Además de las coordenadas geométricas, se pueden almacenar otras cantidades de información como el color en formato RGB, la normal de la superficie, entre otras. En el proceso de visión estereoa comúnmente se generan cantidades grandes puntos 3D para reconstruir escenas, lo cual tiene un alto costo computacional. Aunque las nubes de puntos pueden texturizarse, es muy habitual que en las aplicaciones se conviertan las nubes de puntos en mallas poligonales mediante la aplicación de un proceso de reconstrucción de superficies.

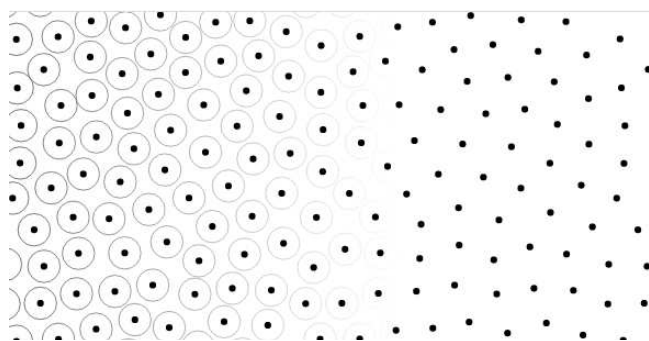


Figura 2.4: Funcionamiento del algoritmo de muestreo de disco de Poisson.

Licencia CC.

### **Simplificación de nubes de puntos**

Existen algoritmos que resuelven el problema de tomar muestras aleatorias sobre la superficie de una malla 3D. Uno de ellos es el algoritmo de muestreo de disco de Poisson y el algoritmo de distribución Monte Carlo [Corsini et al. \(2012\)](#). El algoritmo de disco de Poisson produce puntos compactos pero separados entre sí a una determinada distancia mínima como se muestra en la Figura [2.4](#). Se utilizan muestras candidatas al azar y se va comprobando que esten separadas a una distancia mínima para que el punto se considere como activo y sea parte del conjunto de puntos simplificados.

### **2.2.5. Reconstrucción de superficies 3D**

Las superficies 3D deben ser reconstruidas a partir de un conjunto de nubes de puntos o de medidas de profundidad dispersas [Jain et al. \(1995\)](#). Una malla poligonal es una colección de vértices unidos por aristas, donde, la unión de tres vértices unidos forman un triángulo. Existen varias técnicas para convertir una nube de puntos en una superficie 3D, una de las técnicas más empleadas es la triangulación de Delaunay que forma una malla de triángulos a partir de la nube de puntos [Wöhler \(2012\)](#). Mediante esta técnica se genera una red de triángulos convexa que cumpla la condición de que cada triángulo dentro de una circunferencia circunscrita no debe contener ningún vértice de otro triángulo en el interior.

### **2.2.6. Procesamiento de mallas triangulares**

Actualmente, se han desarrollado diversos algoritmos para analizar mallas tridimensionales lo cual tiene muchas aplicaciones para el reconocimiento de objetos en escenas

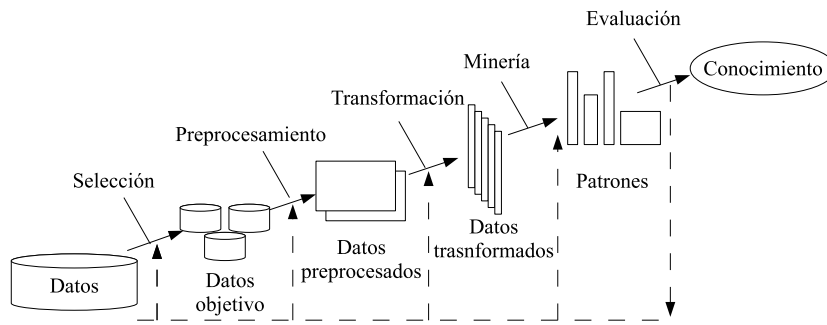


Figura 2.5: Fases del proceso KDD.  
Adaptado de: Hernández Orallo et al. (2004).

3D. Algunas tareas de procesamiento de mallas son: simplificación de nubes de puntos usando decimación, filtro laplaciano, componentes conectados, entre otras; estas técnicas se describen a continuación.

## 2.3. Proceso de extracción de conocimiento KDD

Es el proceso de descubrir conocimiento e información valiosa dentro de conjuntos de datos contenidos en algún repositorio que sirve para hacer conclusiones basadas en los modelos de datos. El descubrimiento de conocimiento en conjuntos de datos es el proceso no trivial de identificación de patrones válidos y potencialmente útiles. El descubrimiento de patrones debe ser validado para datos nuevos con algún grado de certeza. La Figura 2.5 muestra las etapas del proceso KDD las cuales se describen a continuación Hernández Orallo et al. (2004).

### Selección de datos

Consiste en identificar las fuentes de información útil, así como, en recolectar los datos requeridos para obtener conocimiento a partir de ellos. En muchas ocasiones, dependiendo de la aplicación, los datos necesarios para el proceso KDD pertenecen a distintas fuentes. Por lo cual el reto consiste en realizar una búsqueda exhaustiva de cada fuente que aporte datos valiosos al proceso. Los datos de imágenes pueden ser recolectadas de distintas fuentes. Para ambientes de tráfico vehicular, uno de los repositorios públicos más completos es la suite KITTI Geiger et al. (2013). En este repositorio se han creado conjuntos de datos obtenidos con distintas tecnologías como visión estéreo, sensores LIDAR, odometría, flujo óptico y seguimiento de objetos 3D. La

recolección de datos de fuentes propias puede ser obtenida mediante el uso de sensores como: videocámaras, sensores LIDAR, RADAR, sensores odométricos, sensores de ultrasonido, entre otros.

### **Preparación de los datos**

Esta fase se encarga de seleccionar, limpiar y transformar los datos que se van a analizar para mejorar su calidad. La calidad del conocimiento descubierto no solo depende del algoritmo sino también de la calidad de los datos. Para ello, después de haber realizado la recopilación se deben seleccionar y preparar un subconjunto de datos donde se aplican diversas técnicas de procesamiento de imágenes con el objetivo de eliminar la información irrelevante de las mismas y segmentar la región de interés.

### **Minería de datos**

Esta fase se encarga de descubrir patrones en grandes conjuntos de datos. Incluye un ajuste de modelos y la determinación de patrones de los datos observados. Este ajuste es importante para inferir conocimiento. El aprendizaje automático tiene como objetivo diseñar algoritmos que aprendan para realizar determinadas tareas cada vez mejor Müller & Guido (2016). Para ello se construye un modelo basado en los datos recopilados, el cual es una descripción de los patrones y relaciones entre los datos que pueden usarse para generar predicciones. Además, se seleccionan los algoritmos de aprendizaje automático y se valida cada modelo. Las tareas principales de la minería de datos son: clasificación, agrupamiento o *clustering* y reglas de asociación.

La clasificación es una tarea en la que cada elemento pertenece a una clase indicado mediante el valor de un atributo. El resto de los atributos de la instancia se utilizan para predecir la clase. Para ello se construye un modelo conciso de la distribución de etiquetas de clase para inferir a partir de un conjunto de datos de entrenamiento la clase a la que pertenecen los datos de prueba.

### **Evaluación y validación**

En esta fase se seleccionan y validan los modelos mediante la aplicación de métricas de evaluación de los resultados obtenidos con el objetivo de medir la calidad de los patrones descubiertos. Los patrones descubiertos deben tener tres propiedades, por ello deben ser: precisos, comprensibles y útiles. El modelo predictivo deberá ser entrenado con una porción de los datos (datos de entrenamiento) y luego validarlo con el resto de los datos

---

(datos de prueba). Esta separación es necesaria para validar la precisión del modelo de forma independiente ya que de otra forma el modelo se considera sobreestimado.

El método de evaluación más usado es la **validación cruzada con n pliegues (n-folds cross validation)** donde los datos son divididos aleatoriamente en n grupos. Un grupo se utiliza para el conjunto de prueba y con los otros n-1 restantes se construye un modelo para predecir el resultado. Las medidas de evaluación más comunes para los modelos de clasificación binaria son las matrices de confusión. De estas se extienden medidas estadísticas como: precisión, exactitud, recuerdo, sensibilidad y la medida F [Hernández Orallo et al. \(2004\)](#). Otra medida de evaluación muy útil son las curvas ROC (*Receiver Operating Characteristic*) que representan gráficamente la sensibilidad contra especificidad que resultan de los puntos de corte en el rango de los resultados.

### 2.3.1. Extracción de características

Consiste en obtener información que está presente en las distintas regiones de interés de las imágenes para generar un vector características que es usado para generar un modelo predictivo. Existen una diversidad de técnicas para extraer características de geometría, color, textura, descriptores estadísticos, basadas en valores propios, entre otras.

#### Histogramas de patrones binarios locales (LBP)

El histograma de patrones binarios locales (LBP) es el vector de características utilizado para entrenar un modelo predictivo donde, según su valor, se etiqueta la clase a la que pertenece cada imagen. La idea es extraer información de textura del histograma de ocurrencia de patrones binarios locales calculados a partir de la relación entre el valor de intensidad de cada pixel con sus ocho vecinos. Las características de LBP son las frecuencias de cada uno de los contenedores del histograma. El LBP se calcula en tres pasos: (i) codificación, (ii) mapeo e (iii) histograma [Mery \(2015\)](#).

### 2.3.2. Algoritmos del aprendizaje supervisado

Existen dos tipos de aprendizaje automático: supervisado y no supervisado. El primero consiste en que dado un conjunto de datos de entrenamiento previamente etiquetado en clases se encuentran mecanismos que nos permitan clasificar nuevos objetos de clases desconocidas. Algunos métodos del aprendizaje automático supervisado son: árboles de decisión, vecinos más cercanos, redes neuronales, máquinas de vectores de

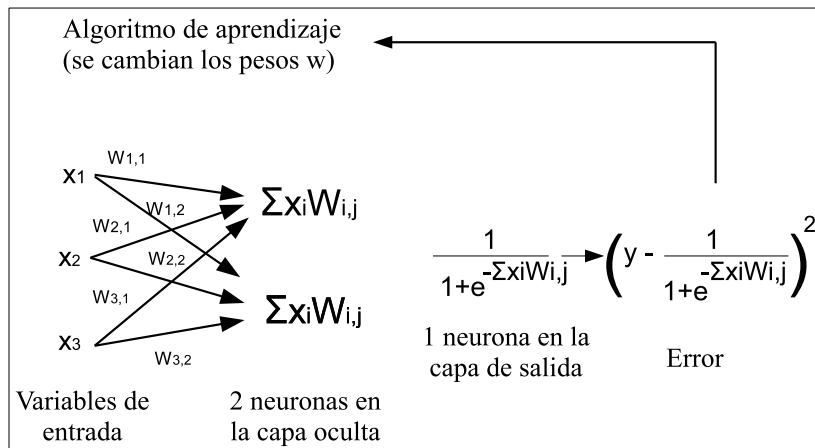


Figura 2.6: Funcionamiento de una red neuronal artificial.

Fuente: Jain et al. (1995).

soporte, entre otros. En contraste, el aprendizaje no supervisado consiste en que dada una muestra de objetos, los cuáles no han sido clasificados previamente, se descubren las clases en las que se agrupan dichos objetos según su similitud. Algunos métodos del aprendizaje no supervisado son: agrupamiento por K-medias, mapas auto-organizados (SOM), estimación de distancia, algoritmo EM (Expectation Maximization), etc.

La tarea de construcción de modelos predictivos requiere que la etapa de entrenamiento y validación estén bien definidas para asegurar que las predicciones serán precisas. Esta tarea es iterativa y deberá repetirse las veces que sea necesario con el objetivo de encontrar los mejores resultados. En este trabajo se utiliza únicamente el aprendizaje automático supervisado por lo que a continuación se explican tres de las principales técnicas.

### Redes neuronales artificiales

Es un paradigma computacional que permite modelar problemas complejos en los que puede haber interacciones no lineales entre variables. La idea es simular el funcionamiento del cerebro con un conjunto grande de neuronas interconectadas. La unidad básica es la neurona y se pretende tener una neurona artificial (Patterson & Gibson (2017)). Posteriormente se pretende tener un conjunto de neuronas artificiales conectadas como una red neuronal. La red neuronal se puede representar como un grafo dirigido con varios nodos y arcos. Cada uno de estos elementos funciona independientemente de los demás usando datos locales, de entrada y salida a cada nodo, para dirigir su procesamiento (ver Figura 2.6).

---

La organización más popular consta de una capa de entrada en la que cada nodo es una variable independiente a examinar, algunos nodos internos organizados en varias capas ocultas y una capa de salida con los nodos de salida (con los valores de las variables objetivo). Cada nodo de la capa de entrada se conecta a cada nodo de la capa oculta. Estos últimos nodos pueden estar conectados a nodos de otra capa oculta o a la capa de salida. Cada arco se etiqueta por un peso de conexión  $w$  y en cada nodo existe una función de activación que indica el efecto del nodo correspondiente sobre los datos que entran. Una red neuronal ya entrenada se puede usar introduciendo los valores de los atributos de una instancia en los nodos de entrada y los nodos de salida determinan la predicción para dicha instancia. Entrenar la red neuronal consiste en encontrar los valores adecuados para los pesos  $w_i$  de cada neurona tal que la salida final sea la adecuada. Los tipos de redes neuronales más usados son: perceptrón (booleano), perceptrón (continuo), red de una sola capa de perceptrones, perceptrón multicapa, entre otras variantes.

### **Máquinas de Vectores de Soporte (SVM)**

Es un clasificador lineal que usan separadores lineales o hiperplanos en espacios de características de alta dimensionalidad, introducidos por funciones núcleo o *kernel*, con sesgos inducidos. Con esta técnica se busca un discriminante lineal que maximice la distancia a los ejemplos fronterizos de las diferentes clases [Cristianini et al. \(2000\)](#).

Una máquina de vectores de soporte es un modelo que representa los puntos de entrenamiento en el espacio, separando las clases a dos espacios los más amplios posibles mediante un hiperplano de separación definido como el vector entre los dos puntos, de las dos clases, más cercanos al **vectores de soporte** (ver Figura [2.7](#)). Se debe entrenar una SVM para construir el modelo predictivo. Cuando los nuevos datos (de prueba) se corresponden con el modelo pueden ser clasificados en una u otra clase. El reto de estas técnicas consiste en que en los problemas no lineales el discriminante lineal también se puede encontrar porque se utilizan núcleos para convertir el problema en un problema de mayor dimensionalidad. Una buena separación entre las clases genera una clasificación de datos correcta.

### **2.3.3. Aprendizaje profundo**

Una red convolucional (CNN) es un algoritmo de aprendizaje profundo capaz de utilizar una imagen de entrada asignando pesos a objetos en la imagen para poder diferenciarlos en la salida. Las CNN son muy empleada en el análisis de imágenes



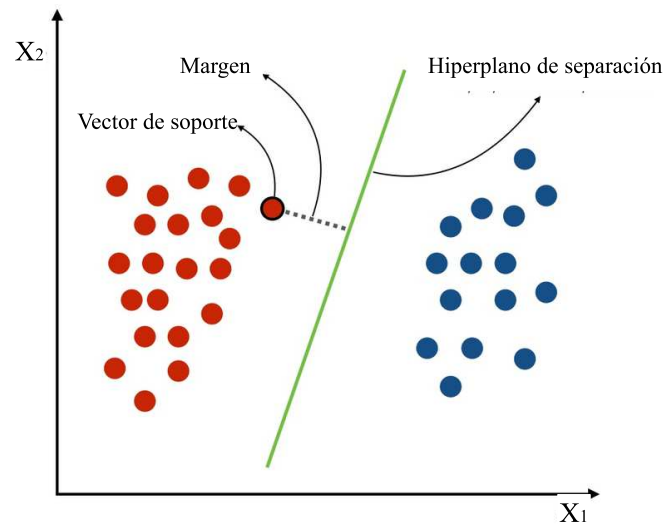


Figura 2.7: Modelo de una máquina de vectores de soporte.

Fuente: [Hernández Orallo et al. \(2004\)](#)

visuales para el reconocimiento y detección de objetos, así como para la clasificación de imágenes. Este algoritmo requiere un preprocesamiento menor que los algoritmos clásicos de clasificación debido a que los CNN tienen la capacidad de aprender filtros y características. La arquitectura de una CNN está inspirada en la organización de la corteza visual y es análoga a las neuronas en el cerebro humano.

La implementación de una CNN debe tener atributos como la entrada y núcleos convolucionales. La entrada es un tensor de la forma (número de imágenes) x (ancho de imagen) x (altura de la imagen) x (profundidad de la imagen). Los núcleos convolucionales tienen un ancho y alto (los cuáles son hiperparámetros) donde la profundidad debe ser la misma de la imagen. Las capas convolucionan la entrada y pasan su resultado a la siguiente capa. Las capas de agrupación local o global permiten optimizar los cálculos subyacentes al reducir las dimensiones de los datos combinando las salidas de los grupos de neuronas en una capa con una sola neurona. Además, este tipo de capas pueden calcular el valor máximo y el promedio de cada grupo de neuronas en la capa anterior. Finalmente, existe un tipo especial de capas llamadas completamente conectadas (FCN) que conectan neuronas en una capa con cada neurona en otra capa. Estas son utilizadas para realizar segmentación de imágenes.

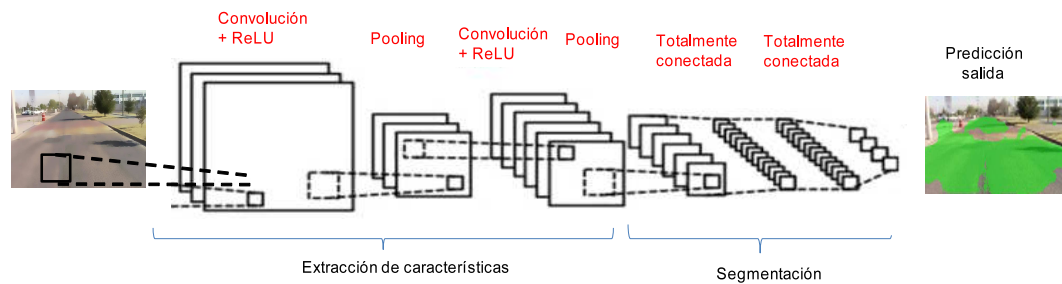


Figura 2.8: Funcionamiento de la FCN para la segmentación semántica.

Fuente: Long et al. (2015)

### Red convolucional totalmente conectada (FCN)

Una FCN es un caso particular de una CNN ya que su diseño consta de una capa de entrada y una de salida además de múltiples capas ocultas. Las capas ocultas típicamente consisten de una secuencia de capas convolucionales con ciertos pesos. Para ello se emplean grupos de píxeles cercanos de la imagen de entrada y se realizan productos escalares contra una matriz pequeña llamada kernel. El kernel tiene un tamaño determinado el cual recorre todas las neuronas de entrada y genera una nueva matriz de salida correspondiente a una nueva capa de neuronas ocultas (Mohit et al., 2018).

Se utilizan funciones de activación como la capa ReLU (Rectified Linear Unit), Sigmoid, tanh, entre otras. Posteriormente, se aplican convoluciones adicionales tales como capas de agrupación o *pooling*, capas completamente conectadas y capas de normalización, las cuáles son denominadas capas ocultas debido a que sus entradas y salidas están enmascaradas por la función de activación y la convolución final. La Figura 2.8 muestra una arquitectura típica de una CNN con múltiples capas.

## 2.4. Síntesis

A partir de esta sección se puede concluir que se pueden aplicar técnicas de visión por computadora y el proceso KDD para la detección de reguladores de velocidad por elevación en las superficies de calles. En este trabajo se aplican algoritmos de aprendizaje supervisado para clasificar la existencia o no de reductores de velocidad en imágenes 3D generadas a partir de conjuntos de nubes de puntos. Las nubes de puntos son convertidas en superficies de mallas tridimensionales para analizar las elevaciones en las calles causadas por RV. En el siguiente capítulo se presentan los trabajos más

significativos que abordan la detección de reductores de velocidad y que se relacionan con los fundamentos teóricos vistos en este capítulo.

# Capítulo 3

## Estado del Arte

En esta sección se describen los trabajos más relevantes sobre la detección de reductores de velocidad y se realiza una clasificación de las técnicas existentes en la literatura, además, se presenta una gráfica comparativa de los resultados que se han obtenido.

### 3.1. Taxonomía de técnicas para detección de RV

Las técnicas que se han empleado para la detección de reductores de velocidad son: monitoreo de superficies usando sensores de *smartphones*, técnicas basadas en imágenes, sensores LIDAR, visión estéreo, aprendizaje profundo y técnicas híbridas Choong et al. (2020); a continuación se describe cada técnica.

#### 3.1.1. Monitoreo por vibraciones usando sensores de smartphone

Se han desarrollado prototipos de dispositivos móviles que monitorean irregularidades en las superficies de vialidades mediante el uso de sensores de teléfonos inteligentes como: acelerómetros, sensores de ultrasonido, de posición y de movimiento para detectar reductores de velocidad y para almacenar sus coordenadas geográficas Afrin et al. (2015); Al-Shargabi et al. (2020); Astarita et al. (2012); Chugh et al. (2014); Daraghmi & Daadoo (2016); Edwan et al. (2019); Harikrishnan & Gopi (2017); Jain et al. (2012); Lanjewar et al. (2015); Madli et al. (2015); Salau et al. (2019); Singh et al. (2017); Sujitha et al. (2016). Estos dispositivos capturan las vibraciones mediante el análisis de señales de aceleración de la superficie. El Sistema de Posicionamiento Global (GPS) se emplea para conocer en tiempo real la ubicación de los reductores de velocidad

en las vialidades [Daraghmi & Daadoo \(2016\)](#); [Singh et al. \(2018\)](#). Los obstáculos en las superficies viales son identificados mediante técnicas de procesamiento de señales de aceleración para reconocer anomalías y clasificar superficies con reductores de velocidad o cruces peatonales [Salau et al. \(2019\)](#); [Souza et al. \(2017\)](#). Se ha aplicado un sistema de reconocimiento de patrones de las lecturas del acelerómetro, así como un clasificador automático usando máquinas de vector de soporte para reconocer estas anomalías [Perttunen et al. \(2011\)](#).

Estos dispositivos utilizan una aplicación de teléfonos inteligentes que envía alarmas de forma temprana a los conductores sobre la existencia de reductores de velocidad para que a una distancia apropiada los vehículos puedan reducir su velocidad y evitar daños [Fekry et al. \(2013\)](#); [Kavitha & Ravikumar \(2021\)](#); [Mukherjee & Majhi \(2016\)](#); [Vimalkumar et al. \(2017\)](#). Además, se ha desarrollado un algoritmo de extracción de datos obtenidos por sensores de teléfonos inteligentes que aplica la minería de datos para encontrar anomalías en las superficies mediante el reconocimiento de patrones [Silva et al. \(2017\)](#). Debido a la popularidad de los teléfonos inteligentes es posible que los usuarios ayuden a monitorear la calidad de la vialidad durante el periodo de conducción [Souza \(2018\)](#). Se ha utilizado la técnica de *Crowdsourcing* para recuperar la ubicación de reductores de velocidad que es almacenada en la nube de acuerdo a la información obtenida de los sensores de teléfonos inteligentes y de los usuarios conectados a una red [Alam et al. \(2020\)](#); [Chen et al. \(2019\)](#); [Singh et al. \(2017\)](#). Sin embargo, la exactitud depende de la comunidad conectada al sistema y podría incluso ser información errónea. La ventaja de los dispositivos que monitorean las superficies por vibraciones es que tienen un diseño muy sencillo y al usar los sensores internos de teléfonos inteligentes resulta un sistema de bajo costo al alcance de todos los usuarios. El uso de GPS en la navegación de autos es de gran ayuda para dar señales preventivas a los conductores debido a que presentan una precisión aceptable. Las aplicaciones de alertas son útiles cuando los reductores de velocidad tienen una ubicación fija. No obstante, para eventos no planificados, por ejemplo, cuando se colocan nuevos RV sin actualizar su ubicación en la nube, no es posible detectarlos.

El GPS y sensores de teléfonos inteligentes presentan las siguientes desventajas: el uso constante del GPS sobrecarga la red en la que se conectan los dispositivos móviles, los errores de precisión son altos debido a que son vulnerables al error ocasionado por del GPS, el retardo es grande cuando el sistema es probado en tiempo real, algunos sensores presentan confiabilidad baja, además, son sensibles a las condiciones del ambiente real. A pesar de las diversas desventajas, estas tecnologías podrían complementarse con otras técnicas para mejorar la precisión en la detección de



Figura 3.1: Imagen de reductor de velocidad y su segmentación.

Fuente: Devapriya et al. (2016)

reductores de velocidad.

### 3.1.2. Técnicas basadas en imágenes

Las técnicas basadas en imágenes realizan procesamiento digital de las imágenes y algunas incluso utilizan clasificadores automáticos para la detección de reductores de velocidad Babu et al. (2021); Chen et al. (2014); Choi et al. (2012); Danti et al. (2013); Tedeschi & Benedetto (2017); Wang et al. (2017). Se han utilizado técnicas como análisis de patrones de color y segmentación de áreas de interés a imágenes digitales (ver un ejemplo en la Figura 3.1). También se han aplicado técnicas básicas de procesamiento de imágenes como el método de umbralización Otsu, métodos de operación morfológica para analizar varios patrones de señalamiento de los RV cuando están bien señalizados y cuando el patrón presenta ruido Babu et al. (2020); Manikandan & Bharathi (2018). Se ha estimado el punto de fuga en las imágenes para determinar la región de interés y se ha calculado la distancia a los reductores de velocidad para reducir la velocidad de los vehículos Kiran & Murali (2014). Además, se ha aplicado un análisis de señales de las líneas del patrón periódico para detectar RV Darwiche & El-Hajj-Chehade (2019). Se ha aplicado un clasificador *Haar Cascade*.

La ventaja principal de los algoritmos de segmentación es que son fáciles de implementar y que utilizan cámaras de bajo costo para la adquisición de imágenes. Sin embargo, la precisión de la detección de reductores de velocidad es baja debido a que está sujeta a la visualización del señalamiento vial y en caso de no estar bien marcado es difícil detectarlos con estas técnicas. Además, cuando las señalizaciones de los reductores de velocidad son muy variadas es difícil aplicar la misma técnica de visión a todos los tipos de patrones.

Asimismo, se han utilizado estas técnicas para la detección de diversas señales de tránsito verticales que incluyen señalamientos de reductores de velocidad Khan et al. (2011); Lillo-Castellano et al. (2015); Soheilian et al. (2013); Wang et al.

(2014); Zhu et al. (2016). Se han aplicado algoritmos de segmentación y métodos de aprendizaje automático para el reconocimiento de señales preventivas de reductores de velocidad Jung et al. (2016); Maldonado-Bascón et al. (2007).

Las limitantes de estas técnicas son las siguientes: presentan mal funcionamiento cuando hay poca iluminación en el ambiente, está sujeta a condiciones climatológicas óptimas, además, la precisión del reconocimiento de los señalamientos disminuye por la escala, rotación y oclusión de las imágenes de señales de tráfico Prieto & Allen (2009). Otra desventaja de este tipo de detectores es que las señales no indican una ubicación precisa de los reductores de velocidad, lo que ocasiona confusión en la conducción.

### 3.1.3. Sensores LIDAR

Otra técnica utilizada para detectar objetos y obstáculos en las superficies viales es el uso de sensores LIDAR. A través de estos sensores es posible obtener datos precisos del contenido de una escena mediante la obtención de datos llamados *point clouds* Hernández & Marcotegui (2009); Serna & Marcotegui (2014). Se han analizado superficies para detectar guarniciones y la geometría de las carreteras Hernández & Marcotegui (2009); Vosselman & Zhou (2009), obteniendo detalles de su morfología y de las irregularidades en las superficies viales Cavalli et al. (2008). Para ello se diseña un algoritmo paralelo que aplica un filtrado de datos de la superficie y de los objetos así como la segmentación de datos usando operadores morfológicos.

También se ha diseñado un algoritmo que segmenta y clasifica objetos urbanos aplicando mapas de elevación digital Serna & Marcotegui (2014). Para la clasificación se aplican máquinas de vector de soporte de acuerdo con los rasgos geométricos, de color y contextuales de los objetos. Para la detección de obstáculos se han hecho predicciones de los cambios en las condiciones de la superficie que consideran la velocidad del auto en tiempo real y la distancia al reductor de velocidad a partir de los datos recolectados por los sensores LIDAR Lee et al. (2018); Yun et al. (2019). Además, se ha diseñado un algoritmo para extraer señales de tránsito usando conocimiento semántico, datos del sensor LIDAR, operadores morfológicos y segmentación Guan et al. (2014); Melo et al. (2018); Yang et al. (2012).

Las ventajas de estos sensores son: el alto nivel de exactitud en la detección de datos debido a que los LIDAR son sensores de largo alcance que pueden coleccionar datos en escenarios altamente complejos y poco visibles por los factores ambientales. Las desventajas son: el elevado costo de estos sensores que oscila entre \$320,000 y \$1,800,540 MX, el alto costo computacional debido a grandes cantidades de datos

---

que son procesados y que son difíciles de interpretar y que no siguen protocolos internacionales [LIDAR \(2018\)](#).

### 3.1.4. Visión estéreo

Se ha propuesto un método de detección de RV mediante el uso de sensores Kinect de Microsoft basado en la reconstrucción 3D de la escena, donde además se calcula la altura del RV y se estima la distancia al mismo [Lion et al. \(2018\)](#). La principal desventaja es que el sensor Kinect es de corto alcance y su rango de visión es corto, lo que dificulta la detección de RV en escenarios de escala real a distancias largas, además, el algoritmo solo funciona cuando los señalamientos están bien pintados.

Se ha implementado un método de estimación del perfil de la superficie de vialidades mediante visión estéreo aplicado a la detección de reductores de velocidad. Se estima el espacio libre, basado en histograma de disparidad, y se calculan mapas de elevación digital usando pixeles en el espacio libre para evitar interferencias de obstáculos [Lee & Yoon \(2018\)](#). La principal limitante es la generación de ruido ocasionada por la estimación de los ángulos entre la cámara y la superficie, además de la interferencia de obstáculos.

Se ha realizado una estimación de superficies en base a la pendiente de los terrenos y a la presencia de obstáculos mediante visión estéreo y procesamiento de superficies usando *B-Splines* [Broggi et al. \(2013\)](#). De esta forma se obtiene un mapa de elevación de la superficie y se detectan obstáculos como autos o peatones de acuerdo a la altura en las regiones del mapa. Se ha construido un algoritmo para representación de escenas mediante la técnica de *stixel-world* [Badino et al. \(2009\)](#). Las superficies son representadas por barras rectangulares adyacentes con cierta anchura y altura para detectar obstáculos. Se ha diseñado un algoritmo para detección de guarniciones mediante ajustes de *splines* cúbicos y mapas de elevación digital [Oniga & Nedevschi \(2011\)](#). Además, se ha diseñado un algoritmo para la extracción de señales de tránsito como cruces tipo cebra y marcas de carriles [Sebsadji et al. \(2010\)](#).

Los algoritmos basados en visión estéreo funcionan tanto para ambientes rurales como urbanos en escenarios reales. La principal ventaja de estos algoritmos es que utilizan sensores de bajo costo para la captura de imágenes. No obstante, las desventajas son: el alto costo computacional requerido para analizar datos densos en las imágenes y el corto alcance debido a la descalibración de las cámaras.



### 3.1.5. Aprendizaje profundo

Se propone un método que detecta e informa al conductor sobre la presencia de RV con o sin señalamientos en calles de la India en tiempo real. Se aplican técnicas de aprendizaje profundo donde se entrena un modelo de red neuronal convolucional (CNN) para clasificar y reconocer los RV en las imágenes [Shah & Deshmukh \(2019\)](#). Además, se aplica un algoritmo de segmentación semántica para la detección de reductores de velocidad en vialidades [Arunpriyan et al. \(2019\)](#). Se utiliza una red neuronal convolucional basada en la interpretación de cada pixel en las imágenes. Se ha desarrollado un prototipo embebido que ayuda a los autos a reducir su velocidad cuando se detecta un RV a una distancia que también se estima donde se entrena un modelo de aprendizaje profundo y técnicas de visión artificial [Dewangan & Sahu \(2020\)](#). La desventaja de este método es que el algoritmo funciona para ambientes a escala miniatura donde los RV están correctamente señalizados pero no funciona en ambientes a escala real.

La limitante principal de esta técnica es que utiliza modelos estándar previamente construidos que sirven para el tipo de imágenes consideradas y no para cualquier conjunto de imágenes con diversos elementos de tráfico. Además, se deben etiquetar los reductores de velocidad de todo el conjunto de imágenes de entrenamiento para poder entrenar al sistema, lo cual requiere mucho trabajo manual.

### 3.1.6. Técnicas híbridas

Existen trabajos que combinan algunas técnicas de las mencionadas anteriormente con la finalidad de mejorar el desempeño individual de cada una. Se han combinado técnicas de procesamiento de imágenes con aprendizaje automático para detectar RV. Se ha desarrollado un algoritmo de reconocimiento de caracteres (OCR) que aplica técnicas de filtrado de imágenes, segmentación de áreas de interés, extracción de características a un patrón binario local y se ha entrenado un modelo de clasificación usando máquinas de vector de soporte para reconocer RV [Irrehbude et al. \(2019\)](#).

Algunos trabajos combinan el uso de GPS y sensores de teléfonos inteligentes con algoritmos de aprendizaje automático para la detección de RV y en general para detectar anomalías en las superficies viales [Dey et al. \(2019\)](#). Se ha aplicado el algoritmo de clasificación de bosques aleatorios, de minería de datos, para analizar la información obtenida de los sensores [Joon et al. \(2019\)](#). La limitante de estas técnicas es que están sujetas a la sobrecarga de la red y los retardos en el uso de sensores de teléfonos

---

inteligentes.

La investigación realizada por Fernández presenta un algoritmo que combina el uso de sensores LIDAR y el procesamiento digital de imágenes para detectar reductores de velocidad como un tipo de obstáculos representados por cruces tipo cebra, propios de España y de algunos otros países de habla hispana Fernández et al. (2012). Los resultados de este trabajo revelan que cuando el auto viaja a una velocidad estándar la detección de reductores de velocidad se realiza a un 94%. Sin embargo, cuando hay muchos vehículos con sensores LIDAR, sus señales podrían superponerse y confundirse.

Se ha construido un sistema de visión para autos usando sensores LIDAR y procesamiento digital de imágenes que detecta reductores de velocidad Succang et al. (2017). Se utilizan sensores LIDAR para detectar los reductores de velocidad y se ha encontrado que cuando el RV se encuentra muy cerca del punto de visión la precisión es muy alta. Se diseña un sistema de visión estéreo para detectar reductores de velocidad usando mapas de disparidad, detección de bordes para identificar calles y procesamiento de imágenes por la morfología del patrón de señalización de los reductores de velocidad Bahena et al. (2010).

## 3.2. Comparación de resultados de algunos trabajos relacionados

La Figura 3.2 muestra una gráfica de la tasa de detección de reductores de velocidad reportadas en algunos trabajos relacionados. Como se puede observar, la mayoría de los trabajos consideran que los reductores de velocidad tienen señalizaciones viales bien marcadas. En general, presentan buenos resultados, pero no consideran las mismas condiciones de interés de este trabajo.

El trabajo de Sebsadji, que aplica la técnica de visión estéreo, presenta una tasa de detección del 77% la cual tiene un porcentaje relativamente bajo con respecto a la media Sebsadji et al. (2010). Los trabajos que usan sensores LIDAR en la detección presentan una tasa del 82% y 83% lo cual es un valor aceptable. El trabajo que combina el uso de sensores LIDAR y técnicas de procesamiento de imágenes para detectar RV presenta una tasa de detección del 94% Fernández et al. (2012). Aunque este valor es alto, el uso de sensores LIDAR tiene la principal limitante de que existe alto traslape entre señales de varios de estos sensores que interactúan simultáneamente. Los trabajos que aplican la técnica de monitoreo de superficies por vibraciones presentan una tasa de detección

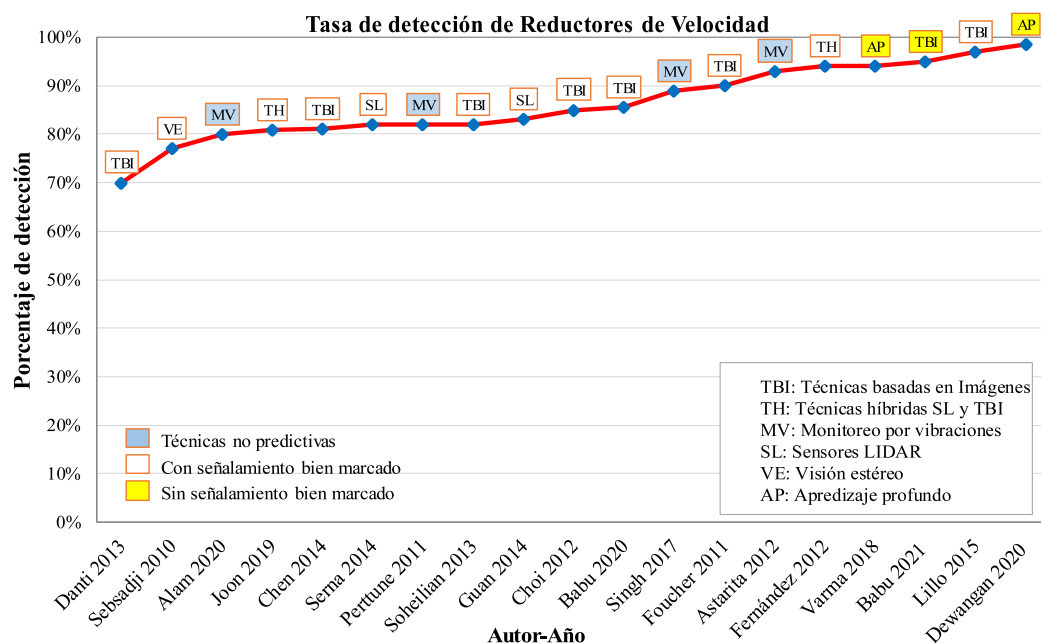


Figura 3.2: Tasa de detección de RV reportada por algunos autores.

Fuente: Elaboración propia.

del 80 %, 93 %, 82 % y 89 % Alam et al. (2020); Astarita et al. (2012); Perttunen et al. (2011); Singh et al. (2017). Aunque estas técnicas presentan buenos resultados no son modelos predictivos para la detección de reductores de velocidad y no corresponden a las mismas condiciones consideradas en este trabajo. El trabajo que combina el uso de GPS y minería de datos presenta una tasa de detección de RV del 80.9 %, lo cual es aceptable, pero está sujeto a los errores de precisión que presentan los GPS, a los retardos y sobrecarga de la red Joon et al. (2019).

Los trabajos que aplican técnicas basadas en imágenes presentan resultados muy variados que dependen de las condiciones bajo las cuales son capturadas las imágenes. Estos resultados presentan, respectivamente, una tasa de detección de reductores de velocidad del 86 %, 95 %, 81 %, 85 %, 70 %, 90 %, 97 % y 82 % Babu et al. (2020) Babu et al. (2021); Chen et al. (2014); Choi et al. (2012); Danti et al. (2013); Foucher et al. (2011); Lillo-Castellano et al. (2015); Soheilian et al. (2013). Los trabajos que presentan una tasa alta de detección (del 90 %, 95 % y 97 %) analizan imágenes capturadas bajo condiciones altamente controladas. Las limitantes que tiene esta técnica son: los algoritmos de detección no funcionan bien cuando hay poca iluminación, la correcta detección está sujeta a condiciones climatológicas ideales, se debe considerar que las imágenes se encuentren a una cierta escala y rotación, sin deformaciones ni

---

sombras ya que estos factores disminuyen la precisión de la detección.

Cabe mencionar que la mayoría de estos trabajos detectan reductores de velocidad cuando su señalamiento vial está bien marcado y existen pocos métodos de detección cuando su señal no está bien marcada. Uno de los trabajos relacionados con esta investigación es llevado a cabo por Babu [Babu et al. \(2021\)](#) el cual aplica técnicas básicas de procesamiento de imágenes donde se detectan líneas de los límites de RV mediante la transformada de Hough. Está basado en la detección de RV por color y solo funciona para calles con material de asfalto. Los resultados obtenidos presentan una tasa de detección del 95 % pero tienen la limitante de que no funcionan para calles con material distinto al asfalto, además, las irregularidades en las calles podrían causar confusiones en la detección. La metodología propuesta es independiente del material de la calle y funciona para distintos materiales.

Otro trabajo relacionado con las condiciones consideradas en esta investigación es el realizado por Varma [Varma et al. \(2018\)](#) el cual aplica el aprendizaje profundo para la detección de reductores de velocidad. En ese trabajo se presenta una tasa de detección de aproximadamente el 94 %, el cual está por arriba de la media como se observa en la Figura [3.2](#). Sin embargo, a pesar de mostrar buenos resultados no se pueden obtener detalles de los reductores de velocidad. La metodología propuesta en este trabajo permite identificar el tipo de reductores de velocidad, así como algunos otros detalles sobre su forma y tamaño.

El trabajo que presenta mejores resultados usa un modelo de red neuronal convolucional para la detección de RV con una exactitud del 98.54 % y a pesar de que considera tanto RV marcados como sin señalamiento, el ambiente donde fue probado no funciona en escala real [Dewangan & Sahu \(2020\)](#). Este modelo funciona en un ambiente simulado en escala miniatura y solo en un tipo de material de la calle, por lo tanto, en ambientes a escala real no funciona el modelo. La ventaja del presente trabajo es que la metodología propuesta funciona para ambientes a escala real y con varios tipos de material de las calles.

Existen varias técnicas para detectar RV tanto con señalamiento como sin señalamiento, sin embargo, las técnicas existentes no han logrado resolver por completo esta tarea ya que fallan ante ciertas condiciones del ambiente. A diferencia de los trabajos existentes, en el presente trabajo se consideran ambientes a escala real y varios tipos de materiales de las calles. Para ello se utilizan videocámaras que son más exactas que los sensores de un smartphone y son de bajo costo en comparación con los sensores LIDAR. Para la detección de RV bien señalizados se entrena un modelo donde se extraen los patrones generados por el señalamiento, mientras que para la detección de RV sin señalamientos

se reconstruye la escena 3D para analizar las elevaciones en las superficies de las calles. Por ello, este trabajo presenta grandes ventajas con respecto a los trabajos relacionados.

### 3.3. Segmentación de calles

La segmentación de vialidades es una tarea fundamental para la navegación de autos autónomos y para la detección de objetos de tráfico vehicular. Algunas técnicas empleadas para esta tarea son: técnicas basadas en el análisis de color y texturas, basadas en morfología y aprendizaje automático.

La segmentación de vialidades basada en análisis de color y textura se enfoca, principalmente, en la detección de líneas de señalamiento de los límites de las calles [Sotelo et al. \(2004\)](#), donde se consideran variaciones de iluminación y sombras. La principal limitante de estas técnicas es que fallan cuando los señalamientos no están bien marcados siendo una situación muy común en países en vías de desarrollo. Otras técnicas basadas en color aplican un algoritmo de segmentación basado en el crecimiento de regiones [Alvarez et al. \(2008\)](#). Algunas técnicas basadas en el análisis de texturas se enfocan a la determinación del punto de fuga de la vialidad aplicando un análisis de la distribución de texturas, filtros wavelet y Gabor e información 3d del contenido de las imágenes y [Rasmussen et al. \(2004\)](#), [Soquet et al. \(2007\)](#), [Kong et al. \(2009\)](#) y [Alvarez et al. \(2010\)](#). Estas técnicas presentan errores altos cuando la forma de las vialidades son curvas debido a que es más difícil aproximar la región de interés.

Para la segmentación mediante técnicas basadas en la morfología de imágenes utilizan principalmente el algoritmo de transformación *Watershed* que aplica una transformación matemática sobre el gradiente de una imagen en escala de grises [Beucher & Bilodeau \(1994\)](#), [Beucher \(1992\)](#), [Beucher et al. \(1990\)](#). Este algoritmo considera a la imagen como una superficie topológica donde las intensidades altas indican picos y colinas y las intensidades bajas indican valles. El principal objetivo es extraer marcadores correspondientes a los puntos principales de la vialidad de los límites de la calle. La principal limitante del algoritmo *Watershed* es que tiende a hacer una sobre segmentación de áreas que son falsos positivos además de que es altamente sensible a sombras sobre la región de interés.

Los algoritmos de segmentación de vialidades basados en aprendizaje automático utilizan redes neuronales que es entrenada para etiquetar regiones producidas por la segmentación de distintos objetos en las imágenes [Campbell et al. \(1997\)](#). Otras técnicas más avanzadas realizan segmentación semántica para identificar los objetos

---

dentro de las imágenes de ambientes de tránsito vehicular. Para la clasificación de objetos se aplica el aprendizaje profundo mediante redes neuronales convolucionales (CNN) donde cada uno de los píxeles de una región con características similares (como color, intensidad y textura) pertenecen a una determinada clase [Teichmann et al. \(2018\)](#), [Wang et al. \(2018\)](#). Para ello se genera una arquitectura de red neuronal profunda que es entrenada con una base de datos de imágenes etiquetadas, posteriormente se genera un modelo de predicción que es probado por conjuntos de imágenes de ciertos repositorios. El aprendizaje profundo también es aplicado en la detección de carriles mediante datos obtenidos de sensores LIDAR [Caltagirone et al. \(2017\)](#). A partir de nubes de puntos sin estructura se generan imágenes con vista superior para determinar la elevación con respecto al piso de los objetos. Se utiliza una red convolucional totalmente conectada para hacer segmentación semántica con resultados altamente precisos. La principal limitante de esta técnica es que al coexistir varios sensores LIDAR las señales tiene mucha interferencia que puede causar confusión en la adquisición de los datos. Las CNN son poderosos modelos visuales que producen jerarquías de características, sin embargo, esta técnica falla cuando el tamaño de las imágenes de entrada es arbitrario. En este trabajo se aplica un modelo preentrenado de una FCN para la segmentación semántica de vialidades en países en vías de desarrollo. La segmentación semántica es usada actualmente en diversas aplicaciones de reconocimiento debido a que los resultados obtenidos son muy aceptables. Además, si el modelo ya se encuentra entrenado, el tiempo de ejecución para predecir nuevas imágenes es relativamente pequeño con respecto a otras técnicas.

### 3.4. Síntesis

La detección de reductores de velocidad es un tema de interés para la conducción autónoma y se ha estudiado desde distintos enfoques; en esta sección se han analizado las ventajas y limitantes de cada uno. A pesar de que se han obtenido resultados con precisión alta en la detección de RV cuando están bien señalizados, las técnicas existentes presentan errores altos cuando los señalamientos no están marcados. Este trabajo se enfoca en la detección de RV señalizados y no señalizados usando visión estereo, técnicas de procesamiento de imágenes y aprendizaje automático para ambientes a escala real y considerando varios tipos de material de las calles. En el siguiente capítulo se presenta la metodología propuesta para la detección de RV tanto señalizados como sin señalamientos.



# Capítulo 4

## Metodología

En este capítulo se presenta la propuesta metodológica para la detección de reductores de velocidad tanto señalizados como sin señalamiento. La Figura 4.1 muestra las fases metodológicas para dar solución al problema de investigación y a continuación se describe cada una.

### 4.1. Captura de imágenes estéreo

En esta fase se captura un conjunto de imágenes estéreo de calles con y sin reguladores de velocidad usando un sistema de visión estéreo compuesto por dos cámaras con las mismas características (distancia focal, tamaño de la lente y resolución). Las cámaras están alineadas a la misma altura, separadas a una cierta distancia y de esa forma se capturan pares de imágenes simultáneas de la misma escena. Se

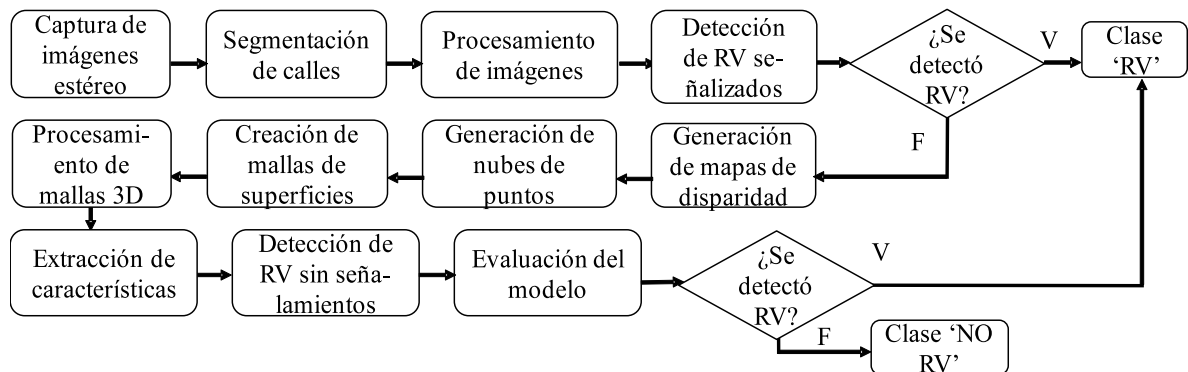


Figura 4.1: Metodología propuesta para la detección de reductores de velocidad.



consideran condiciones controladas del ambiente como: buena iluminación y clima, escenas preferentemente sin sombras y sin oclusiones de peatones, autos u otros obstáculos en la calle. Los RV considerados son aquellos con dimensiones grandes, debido a que son los que más daños pueden ocasionar, con o sin señalamientos viales marcados.

## **4.2. Segmentación de calles con FCN**

Inicialmente se aplica un filtro bilateral al conjunto de imágenes estéreo para eliminar el ruido preservando los bordes de la imagen. Enseguida se aplica un modelo preentrenado para generar la segmentación semántica de calles correspondiente a la región de interés. La segmentación semántica es una técnica del aprendizaje profundo que permite interpretar el contenido de las imágenes mediante la asociación de una etiqueta a cada pixel. De esta forma es posible predecir la clase a la que pertenece cada pixel, para este caso se hace una segmentación binaria donde los pixeles se clasifican en dos clases: calle y no calle. Se utiliza la arquitectura de red VGG de 16 capas correspondiente a una red totalmente convolucional (FCN) previamente entrenada. Una FCN es un caso particular de una CNN donde el tamaño de los datos es arbitrario y producen una salida de datos del mismo tamaño con una inferencia eficiente. Se define la arquitectura de la Figura 2.8 que combina información semántica de una capa profunda y una gruesa con información de una capa fina y poco profunda para generar segmentaciones precisas y detalladas del contenido de las imágenes.

## **4.3. Preprocesamiento de imágenes**

En fase se aplican técnicas de procesamiento digital de imágenes para mejorar su calidad, preparándolas para el proceso de clasificación automática. Las técnicas aplicadas ayudan a mejorar el contraste de las imágenes, a eliminar el ruido y son listadas a continuación:

- Redimensionar imágenes
- Convertir a escala de grises
- Ecuilización CLAHE
- Desenfoque gaussiano

---

Para la detección de RV por patrón de señalamiento, una vez preprocesadas las imágenes, se segmenta el patrón de señalamiento y de esta forma se recorta la región de interés. Las técnicas aplicadas para separar los patrones de señalamiento del fondo de la calle se listan enseguida:

- Umbralización Otsu
- Operadores morfológicos apertura-cerradura
- Componentes conectados
- Filtros de Gabor

La salida de esta fase son imágenes binarias donde el color blanco corresponde al patrón de señalamiento y el fondo negro corresponde a la calle, de esta forma el conjunto de imágenes está preparada para la siguiente fase.

#### **4.4. Detección de RV señalizados**

El conjunto de imágenes preprocesadas y segmentadas es empleado para construir un modelo de clasificación para detectar RV aplicando el algoritmo de máquinas de vectores de soporte para el entrenamiento. En caso de que se detecte un RV en la imagen el algoritmo termina indicando que la clase es RV. En otro caso, cuando el RV no está bien marcado su detección se falla y se continua con la siguiente fase para detectar RV por elevaciones en las superficies mediante visión estéreo. Para medir el desempeño del modelo se calculan algunas métricas de evaluación y se grafican curvas ROC.

#### **4.5. Generación de mapas de disparidad**

A partir de las imágenes de calles segmentadas, se generan mapas de disparidad aplicando el proceso de visión estéreo de la Figura 2.1 y cada fase se describe a continuación:

- Calibración: se estiman los parámetros intrínsecos y extrínsecos de cada cámara individual. Para ello se aplica el método de calibración de Zhang basado en capturar al menos 10 imágenes de un tablero de ajedrez asimétrico en distintas posiciones y orientaciones de la Escalera et al. (2010). Se detectan las esquinas de los cuadrados de la plantilla y se calculan los parámetros que son usados para

corregir las distorsiones en cada cámara. Después de la calibración individual de cada cámara se realiza la calibración estéreo para encontrar la relación geométrica entre ambas cámaras y de esta forma aproximar su posición y orientación con respecto a las coordenadas del mundo real.

- **Rectificación:** en esta fase se busca alinear cada par de imágenes de forma que dado un punto  $p_1$  en el plano de visión de la primera cámara se debe hacer corresponder con el mismo punto de la segunda cámara. Es decir, se aplica que se buscan correspondencias entre los puntos de ambas imágenes de acuerdo a las líneas epipolares de la Figura 2.2. Para esto, se aplica una transformación rígida (rotación y traslación) del sistema de coordenadas de la cámara izquierda a la derecha, ajustando los ángulos y distancias entre las cámaras para conseguir que las imágenes generadas sean coplanares.
- **Mapa de disparidad:** se estiman las disparidades, diferencia entre las coordenadas horizontales de los puntos  $p_L$  y  $p_R$  de la imagen izquierda y la imagen derecha respectivamente, que construyen una nueva imagen en escala de grises denominada mapa de disparidad. Se buscan las parejas de puntos correspondientes que son proyección del mismo punto en el espacio. Se asume que las imágenes están rectificadas, que la escena es estática y que se tienen dos cámaras con características similares. Los ejes ópticos son paralelos y ambas cámaras tienen la misma distancia focal con centros ópticos separados a una distancia.

## 4.6. Generación de nubes de puntos

El mapa de disparidades obtenido se usa para calcular la profundidad y obtener distancias a los puntos observados. Esto se hace mediante una reproyección de coordenadas geométricas que da lugar a una nube de puntos de la imagen donde se almacenan sus coordenadas  $(X, Y, Z)$ . Estas coordenadas corresponden a una estimación de las coordenadas del mundo real en las unidades en que sean configuradas. La visualización de nubes de puntos muestra la reconstrucción 3D de la escena real correspondiente a la región de interés de las calles.

---

## 4.7. Creación de mallas de superficies

Las nubes de puntos obtenidas en la fase anterior son utilizadas para generar mallas de superficies de las calles. Se aplica el algoritmo de triangulación de Delaunay para convertir nubes de puntos en mallas triangulares donde cada punto seleccionado es el vértice de un triángulo que genera la malla de superficie de las calles sin colores asignados Wöhler (2012).

## 4.8. Procesamiento de mallas 3D

Las mallas son procesadas para su mejora aplicando técnicas de análisis de mallas de superficies triangulares. Se aplican técnicas para suavizar las mallas logrando que sean más uniformes eliminando el ruido. Además, se aplica una técnica que simplifica los vértices de las mallas para reducir el costo computacional en su procesamiento preservando su forma. Las técnicas aplicadas para procesar mallas de superficies de calles son las siguientes:

- Suavizado de mallas: se aplica un suavizado laplaciano donde para cada vértice se elige una nueva posición de acuerdo con la información local de sus vecinos. La Eq. 4.1 calcula el laplaciano que alisa la malla 3D.

$$\bar{x}_i = \frac{1}{N} \sum_{j=1}^N \bar{x}_j \quad (4.1)$$

donde N=número de vértices adyacentes al nodo i,  $\bar{x}_j$  =posición del j-ésimo vértice adyacente y  $\bar{x}_i$  =nueva posición del nodo i.

- Decimación: Debido al difícil manejo y alto costo computacional para procesar estos datos, se realiza una reducción de los triángulos de mallas 3D. El objetivo de la reducción es disminuir el tamaño de las mallas preservando su forma y topología y se aplican dos métodos: diezmar y pro diezmar. El algoritmo para diezmar las mallas clasifica cada vértice usando la medida de distancia plano-punto; si se cumple con un determinado criterio de diezmo, se elimina el vértice y se triangula el agujero resultante. El algoritmo de pro diezmar genera mallas progresivas para mejorar el algoritmo de diezmar básico donde se utiliza una cola de prioridad para colocar los vértices clasificados. Además, la malla es dividida en submallas donde cada submalla es procesada de forma recursiva.

Esta fase recibe como entrada las mallas de superficies de calles obtenidas en la fase anterior y genera como salida mallas de superficies procesadas y simplificadas que serán usadas para la detección de RV sin señalamientos pintados.

#### **4.8.1. Extracción de Características**

Diversas características pueden ser extraídas a partir de nubes de puntos y mallas de superficies 3D las cuáles son aquellas; en este trabajo se consideran las características basadas en valores propios y características geométricas consideradas para la detección de RV por elevaciones en las superficies obtenidas en el proceso de visión estéreo.

A partir de esta información se calcula un vector características obtenida por el análisis de superficies 3D de las imágenes segmentadas.

#### **4.8.2. Detección de RV sin señalamientos**

Para el diseño del algoritmo se utiliza el vector características, basado en valores propios y características geométricas, que permita detectar reductores de velocidad por elevaciones en las superficies de calles. Para la preparación del corpus se destina un porcentaje del total de mallas de superficies 3D de calles para entrenamiento y otro porcentaje para pruebas. El conjunto de entrenamiento es etiquetado de acuerdo a la clase a la que pertenecen (R-reductor y NR-no reductor) a partir de su vector características. Se aplican los algoritmos de bosques aleatorios y el clasificador de aumento de gradiente para el entrenamiento de modelos de clasificación de mallas de superficies que detecten RV sin señalamientos. El clasificador es probado con otro conjunto de mallas 3D donde al aplicar el modelo predictivo, obtenido por en el entrenamiento, se infiera la clase a la que pertenece (RV o NRV).

#### **4.8.3. Evaluación del modelo**

El clasificador es evaluado para medir su desempeño y seleccionar aquellos que presenten mejores resultados en la detección mediante el cálculo de métricas de evaluación. Este proceso se itera varias veces hasta obtener el desempeño más aceptable. Se realiza un análisis de resultados mediante matrices de confusión para revisar el rendimiento del algoritmo de detección de RV. Esta técnica permite estimar la precisión, exactitud, *recall*, entre otras métricas para evaluar cada algoritmo. Para ello se realizan varios experimentos con datos de superficies de RV para distintas alturas,

---

longitudes, distancias y geometrías con el objetivo de estimar la precisión del algoritmo y verificar la aceptabilidad del mismo.

Para la síntesis de la información y para verificar la validez de una hipótesis se utiliza el método estadístico de curvas ROC. Se evalúan experimentos mediante el cálculo de sensibilidad (valor predictivo positivo) y especificidad (valor predictivo negativo) para interpretar los resultados del algoritmo. Este método sirve para analizar la exactitud diagnóstica del algoritmo de detección de reguladores de velocidad.

## 4.9. Algoritmo

El Algoritmo 1 describe el proceso de detección de RV que recibe como entrada un conjunto de imágenes estéreo de vialidades  $E = \{(L_1, R_1), \dots, (L_n, R_n)\}$  y la salida es la clase que indica si en cada par de imágenes se detectan o no se detectan RV  $C = \{C_1, \dots, C_n\}$ . La primer tarea del algoritmo consiste en obtener los parámetros de la visión estéreo obtenidos del proceso de calibración de cámaras de acuerdo a los

parámetros intrínsecos y extrínsecos presentados en la sección [2.2.1](#)

---

**Algoritmo 1:** ClasificadorReductoresVelocidad.

---

```

Input: conjunto  $E = \{(L_1, R_1), \dots, (L_n, R_n)\}$ 
Output: conjunto  $C = \{C_1, \dots, C_n\}$ 
1  parametrosEstereo  $\leftarrow$  calibracionEstereo()
2  for cada par de imágenes  $E_i = (L_i, R_i)$  en  $E$  do
3       $P_i \leftarrow$  preprocesamientoImagenes( $L_i$ )
4       $detRV_i \leftarrow$  clasificacionRVPatron( $P_i$ )
5      if  $detRV_i = verdadero$  then
6           $C_i \leftarrow$  'RV'
7      else
8           $S_i = (LSE_i, RSE_i) \leftarrow$  segVialidades( $E_i$ )
9           $malla_i \leftarrow$  generarMallas3D( $S_i$ , parametrosEstereo)
10          $(predRV_i, M_i) \leftarrow$  clasificacionRVElevaciones( $malla_i$ )
11         if  $predRV_i = verdadero$  then
12              $C_i \leftarrow$  'RV'
13         else
14              $C_i \leftarrow$  'NO RV'
15         end
16     end
17 end
18
19 Def segVialidades( $E_i$ ):
20      $E_i \leftarrow$  filtroBilateral( $E_i$ )
21      $S_i \leftarrow$  probarModeloFCNPreentrenado()
22      $mascara_i \leftarrow$  generarMascaraROI( $S_i$ )
23      $ROI_i \leftarrow$  eliminarRegionNoInteres( $mascara_i$ )
24     return  $ROI_i$ 
25
26 Def generarMallas3D( $S_i$ , parametrosEstereo):
27      $(LR_i, RR_i) \leftarrow$  rectificacionEstereo( $LS_i, RS_i$ , parametrosEstereo)
28      $MD_i \leftarrow$  generarMDSGBM( $LR_i, RR_i$ )
29      $MDP_i \leftarrow$  filtroWLS( $MD_i$ )
30      $NP_i \leftarrow$  generarNubesPuntos( $MDP_i$ )
31      $NPS_i \leftarrow$  simplificarNubesPuntos( $NP_i$ )
32      $malla_i \leftarrow$  convertirPuntosAMallas( $NPS_i$ )
33      $mallaP_i \leftarrow$  procesamientoMallas( $malla_i$ )
34     return  $mallaP_i$ 

```

---

---



---

```

36 Def clasificacionRVElevaciones(mallai):
37    $VCE_i = \{Car_1, \dots, Car_m\} \leftarrow \text{extraccionCaracteristicas}(malla_i)$ 
38    $predRV_i \leftarrow \text{probarMCRVEntrenado}(VCE_i)$ 
39    $M_i = \{recuerdo_i, precision_i, exactitud_i\} \leftarrow \text{evaluarMCRVE}()$ 
40   return ( $predRV_i, M_i$ )

```

---

El algoritmo recorre el conjunto de cada par de imágenes estéreo  $E_i = (L_i, R_i)$  realizando el siguiente proceso:

- Segmentación de vialidades ( $\text{segVialidades}(E_i)$ ): este módulo recibe como entrada un par de imágenes estéreo  $E_i$  y como salida se obtiene un par de imágenes segmentadas con la región de interés  $ROI_i$  correspondiente a la calle. Para ello se aplica un filtro bilateral para suavizar las imágenes sin eliminar los bordes ( $\text{filtroBilateral}(E_i)$ ). Se realizan pruebas a partir de un modelo de FCN preentrenado de segmentación semántica de calles generando la imagen segmentada ( $\text{probarModeloFCNPreentrenado}()$ ). Posteriormente, se genera una máscara aplicando operadores lógicos sobre cada par de imágenes de acuerdo con la ROI obtenida por la segmentación ( $\text{generarMascaraROI}(S_i)$ ). Finalmente, se elimina la región de no interés, como edificios, árboles, autos, etc., y se devuelve el área ROI correspondiente a la calle ( $ROI_i$ ).
- Generar mallas de superficies 3D ( $\text{generarMallas3D}(S_i, \text{parametrosEstereo})$ ): este módulo se encarga de generar mallas 3D a partir de cada par de imágenes estéreo segmentadas con la ROI ( $S_i$ ). Para ello, se rectifican las imágenes estéreo, izquierda ( $LS_i$ ) y derecha ( $RS_i$ ), a partir de los parámetros de calibración estéreo ( $\text{rectificacionEstereo}(LS_i, RS_i, \text{parametrosEstereo})$ ). Enseguida, se genera un mapa de disparidad aplicando el algoritmo SGBM ( $\text{generarMDSGBM}(LR_i, RR_i)$ ) el cual es una imagen en escala de grises donde la tonalidad del canal indica la profundidad de los píxeles. Ahora, se aplica un filtro WLS de mínimos cuadrados ponderados para suavizar el mapa de disparidad y hacerlo más uniforme ( $\text{filtroWLS}(MD_i)$ ). Posteriormente, se generan nubes de puntos 3D a partir del mapa de disparidad procesado ( $\text{generarNubesPuntos}(MDP_i)$ ). Se aplica un algoritmo de muestreo para simplificar las nubes de puntos que reducen el costo computacional del algoritmo ( $\text{simplificarNubesPuntos}(NPS_i)$ ). Las nubes de puntos se convierten en mallas triangulares aplicando un algoritmo de reconstrucción de superficies de Poisson ( $\text{convertirPuntosAMallas}(NPS_i)$ ).



- Clasificación supervisada de reductores de velocidad ( $clasificacionRV(malla_i)$ ): en este módulo se extraen las siguientes características sobre la malla 3D ( $extraccionCaracteristicas(malla_i)$ ): pendiente de la superficie, altura, curvaturas, forma alpha, volumen de densidad de puntos locales, planaridad, linealidad y eigen entropía. Se genera un vector características que se utiliza para probar un modelo de clasificación previamente entrenado ( $probarMCRVEntrenado$ ) y lograr una predicción de imágenes con y sin reductores de velocidad. Se evalúa el modelo y se calculan algunas métricas de validación (recuerdo, precisión y exactitud). Se genera como salida la predicción del modelo y las métricas de evaluación ( $predRV_i, M_i$ ).
- Detección de reductor de velocidad: . En caso de que la predicción de RV sea verdadera se asigna, para cada par de imágenes, a la clase  $C_i$  el valor 'RV' ya que se ha detectado reductor de velocidad; en caso contrario la clase es 'NO RV'.
- Imprimir resultados: se muestran los resultados de la clase a la que corresponde cada par de imágenes estéreo indicando si hay o no hay RV. Además, se imprimen algunas métricas de evaluación obtenidas al probar el modelo de clasificación supervisado para la detección de RV.

El algoritmo presentado es una propuesta de solución al problema de detección de reductores de velocidad tanto para el caso en que estén señalizados como para los casos cuando no tengan señalamiento. Como se puede observar, en casos cuando el patrón de señalamiento está bien marcado, se aplica un modelo de clasificación, pero si el RV no tiene señalamiento, se aplican técnicas de visión estéreo para reconstruir las superficies 3D y hacer la detección por elevaciones en las superficies que sea independiente del tipo de material de la calle y funciona a escala real.

## 4.10. Síntesis

En este capítulo se presenta la metodología propuesta para detectar reductores de velocidad con y sin señalamientos mediante técnicas de procesamiento de imágenes, visión estéreo y aprendizaje automático. Las etapas que se consideran son: captura de imágenes estéreo, segmentación de calles con FCN, preprocesamiento de imágenes, detección de RV por patrón, generación de mapas de disparidad, generación de nubes de puntos, creación de mallas de superficies, procesamiento de mallas 3D, extracción de características, detección de RV en superficies 3D y evaluación del modelo. Esta

---

metodología considera distintas condiciones de las calles que las técnicas existentes no han logrado resolver al 100 % para la detección de RV con y sin señalamientos.



# Capítulo 5

## Resultados Obtenidos

En este capítulo se presentan los resultados obtenidos de cada fase de la propuesta metodológica presentada en la Sección 4. Los principales resultados de este trabajo son: la construcción de un conjunto de imágenes estéreo de calles con y sin RV y la implementación de un algoritmo para la detección de RV con y sin señalamientos en las calles.

### 5.1. Captura de imágenes estéreo

Para la construcción del sistema de visión estéreo se considera un par de cámaras con las siguientes características: videocámaras Canon Vixia HR R72, sensor de imagen CMOS Full HD a 3.28 megapíxeles, procesador de imagen DIGIC DV 4, captura de vídeo en resolución de 1,920 x 1,080 y 57x (32.5-1853mm) de zoom. La base del sistema permite que las cámaras se desplacen para poder seleccionar la línea base (distancia entre cámaras) manteniendo el área de intersección de la escena entre las dos cámaras. La Figura 5.1 muestra el sistema de visión estéreo que se ha construido.

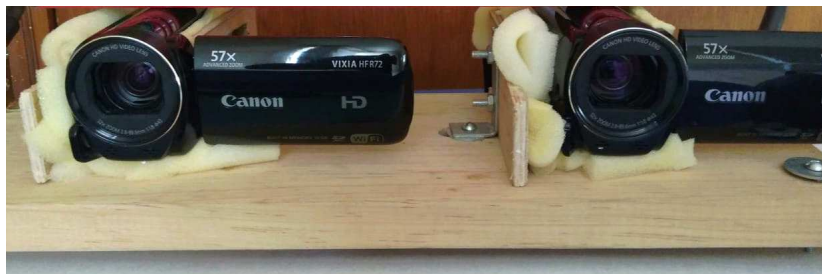


Figura 5.1: Sistema de visión estéreo.

El sistema sirve para capturar simultáneamente imágenes estéreo de vialidades con o sin reductores de velocidad que es montado en la parte exterior delantera de un auto. Para capturar las imágenes estéreo se toman en cuenta las siguientes condiciones del ambiente:

- Iluminación: se consideran imágenes claras con buena exposición, es decir, sin subexposición ocasionada por imágenes oscuras y sin sobreexposición ocasionada por un exceso de luz en la imagen.
- Clima: se considera un nivel de iluminación entre 32,000 lux y 100,000 lux que corresponde a la luz solar en un día promedio mínimo y máximo respectivamente. No se consideran imágenes con clima nublado, sin neblina ni lluvia, tampoco imágenes de noche.
- Otras especificaciones: preferentemente sin sombras causadas por árboles y otros obstáculos.

Las características de las vialidades son las siguientes:

- Tipo de piso: se consideran vialidades con material de asfalto, concreto o adoquín con pocas irregularidades en el piso. Además, se capturan únicamente vialidades cuya superficie no esté altamente deteriorada.
- Con o sin señalizaciones horizontales: este tipo de señalizaciones están basadas en pintura sobre el piso para resaltar las marcas.
- Límites: se consideran calles delimitadas con o sin líneas por acotamientos o por marcas en la guarnición de aceras.
- Sin oclusiones de objetos: se analizan calles preferentemente sin la presencia de autos, peatones, ciclistas y otros objetos que generen oclusión.
- Tipo de vialidad: urbanas y semiurbanas que pueden contener árboles y algunas construcciones en la calle.

Las características de los reductores de velocidad se describen a continuación:

- Formas: los reductores de velocidad son relieves que sobresalen del pavimento en todo el ancho de la vialidad. Se consideran reductores con formas ovaladas o con superficies planas (estos son de tipo trapezoidal y son usados para cruce peatonal).



(a) Vialidad sin reductor de velocidad (b) Vialidad con reductor de velocidad

Figura 5.2: Imágenes de vialidades capturadas con el sistema de visión estéreo.

- Dimensiones: con consideran reductores de velocidad con dimensiones grandes debido a que son los más comunes en México y a que son los que provocan mayor variación de velocidad en la conducción.
- Color: se consideran reductores de velocidad con y sin marcas de señalamiento que puede tener distintas tonalidades debido al material del piso.

La Figura [5.16](#) muestra dos imágenes de vialidades con y sin reductores de velocidad capturadas. Los experimentos han sido ejecutados en una computadora con las siguientes características: procesador AMD A10-8700P Radeon R6 1.8 GHz, 8 GB de RAM y sistema operativo Windows.

## 5.2. Segmentación de calles con FCN

Cada par de imágenes estéreo capturadas con el sistema de visión son preprocesadas mediante la aplicación un filtro bilateral para eliminar el ruido y suavizar la imagen logrando que los colores sean más uniformes conservando los bordes de los objetos. Para ello se utilizan los siguientes parámetros:

- Diámetro de la vecindad de pixeles  $d = 50$
- Valor de sigma en el espacio de color  $\sigma_{Color} = 15$
- Valor de sigma en el espacio de coordenadas  $\sigma_{Space} = 18$

Se recolectan los datos necesarios para probar el modelo FCN preentrenado desde un repositorio público ([https://github.com/JunshengFusemantic\\_segmentation](https://github.com/JunshengFusemantic_segmentation)). El conjunto de cada par de imágenes estéreo capturadas y preprocesadas son utilizadas como nuevas imágenes para probar el modelo de segmentación semántica (ver una muestra



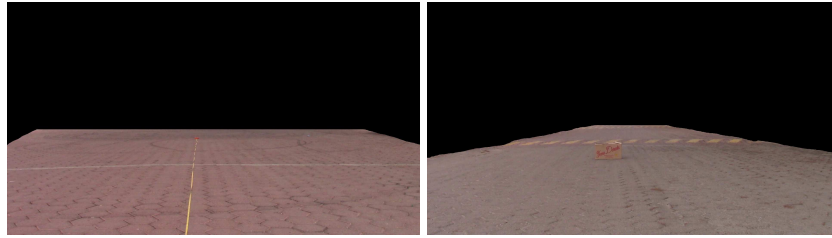
Figura 5.3: Base de datos de vialidades capturadas con el sistema de visión.

de imágenes en Figura 5.3). Cada imagen tiene un tamaño original de 1920 x 1080 píxeles y se han capturado un total de 4,197 segundos de video que equivalen a 104,925 fotogramas.

La segmentación semántica colorea de color verde la calle correspondiente a la región de interés ROI. Las imágenes son redimensionadas a un tamaño de 576 x 160 píxeles para reducir el costo computacional. Se utiliza una computadora con procesador Intel Core(TM) i5-6200U CPU 2.40GHz con 8 GB de RAM. El tiempo medio de ejecución para predecir la segmentación de vialidades por imagen es de 19.03 seg. Las pruebas realizadas con imágenes de vialidades presentan distintas tonalidades en la zona de la calle; sin embargo, existen algunos escenarios donde el modelo actual no funciona correctamente y por ello se consideran las siguientes restricciones sobre las imágenes para que la segmentación de calles funcione de forma aceptable:

- Preferentemente las calles no deben presentar sombras ocasionadas por obstáculos.
- No considerar calles con pavimentación altamente deteriorada.
- No utilizar imágenes con áreas sobre expuestas o zonas con alto brillo.
- Evitar imágenes donde el color de la vialidad se confunde con el de las aceras debido a que no existe un límite entre ambos bien definido.

A partir de las imágenes obtenidas de la segmentación de calles, se genera una máscara binaria que al aplicarla sobre las imágenes estéreo originales elimina las



(a) Vialidad sin reductor de velocidad (b) Vialidad con reductor de velocidad

Figura 5.4: Eliminación de región de no interés en imágenes estéreo.

regiones de no interés (como vegetación, construcciones, obstáculos y cielo) y mantiene la ROI correspondiente a la región de la calle (ver Figura 5.4). Para reducir el costo computacional en la construcción de imágenes estéreo, se generan mapas de disparidad solo de la región de interés.

La Figura 5.5 muestra una comparación entre la segmentación semántica verdadera y la predicha a partir de dos imágenes de vialidades. Para evaluar el desempeño del modelo se aplican dos métricas a un conjunto de 300 imágenes de cada par estéreo, donde una predicción exitosa es aquella donde se maximiza la superposición entre las clases estimadas y verdaderas. Dos métricas muy empleadas por su alta eficacia son el índice de Jaccard y el coeficiente de dado. El índice de Jaccard, conocido también como IoU o intersección sobre unión, calcula el área de superposición entre la segmentación predicha y el *ground truth* o verdadera dividida entre el área de la unión la segmentación predicha y el *ground truth* (ver Ecuación 5.1). Por otro lado, el coeficiente de dado (Dice) es dos veces la superposición dividida por el número total de píxeles en ambas imágenes (ver Ecuación 5.2). Ambas métricas varían entre 0 y 1 donde el 0 indica que no hay superposición y el 1 corresponde a una segmentación perfectamente superpuesta.

$$IoU = \frac{Intersección}{Unión} = \frac{TP}{TP + FP + FN} \quad (5.1)$$

$$Dice = \frac{2 * Intersección}{Unión + Intersección} = \frac{2TP}{2TP + FP + FN} \quad (5.2)$$

donde TP son los verdaderos positivos, FP los falsos positivos y FN los falsos negativos. A partir de un conjunto imágenes se calculan ambas métricas y se promedian para obtener los resultados finales. La media de IoU obtenida es de 0.916 mientras que la media de Dice es 0.938. Esto demuestra que los resultados obtenidos con el modelo FCN para la segmentación semántica de calles presentan una confiabilidad alta.



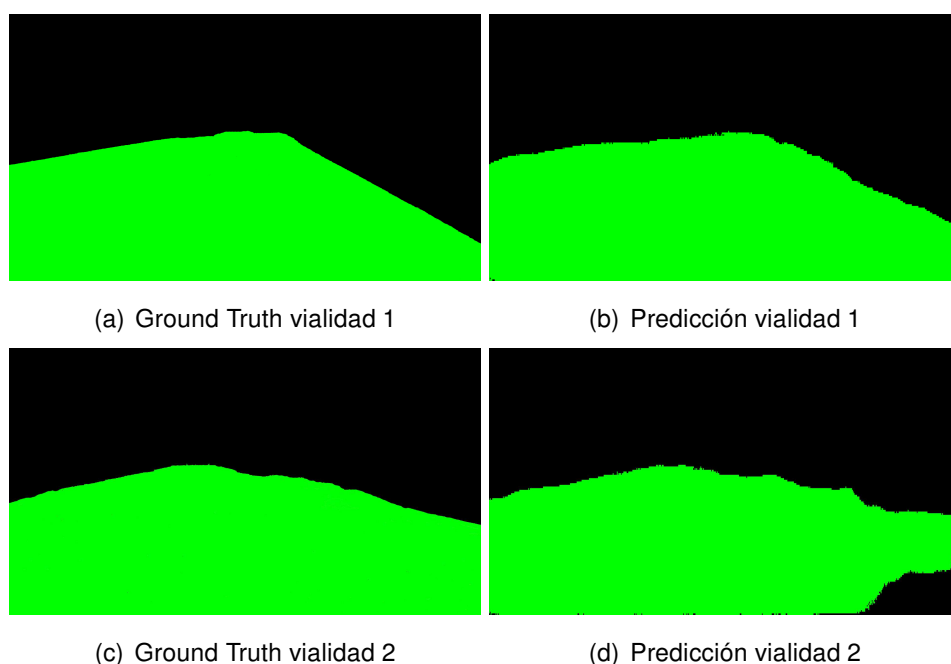


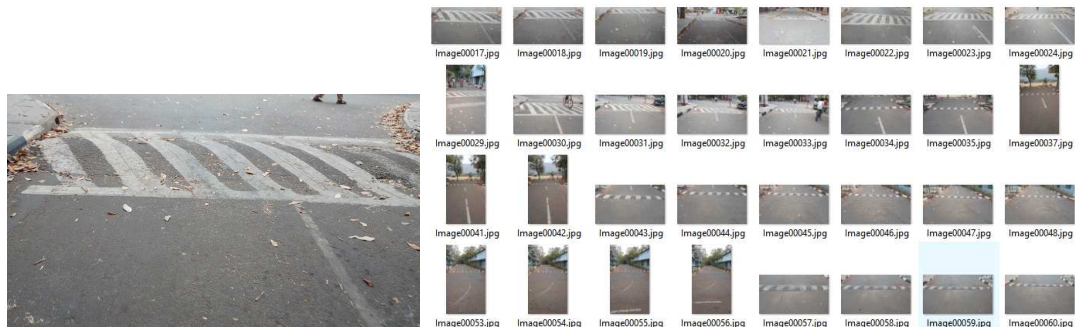
Figura 5.5: Comparación entre la segmentación verdadera y la predicha.

### 5.3. Preprocesamiento de imágenes

En la Figura 5.6 se observa una muestra de un conjunto de imágenes 2D de vialidades con y sin RV señalizados con patrón cruce cebra (obtenido de Varma et al. (2018)). Este conjunto de imágenes es empleado para entrenar un modelo de detección de reductores de velocidad por patrón de señalamiento usando técnicas de procesamiento de imágenes y aprendizaje supervisado.

Debido a que las imágenes de entrada están capturadas bajo diferentes condiciones de iluminación, es necesario aplicar técnicas de procesamiento para uniformizar las imágenes y mejorar su calidad. Las técnicas de preprocesamiento aplicadas se describen a continuación y se indican los valores empleados:

- Redimensionar imágenes a un tamaño de 250 x 100 px
- Convertir a escala de grises: se aplica la ecuación de luminosidad  $L(P_R, P_G, P_B) = 0.21P_R + 0.72P_G + 0.07P_B$ , donde  $P_R, P_G, P_B$  son los componentes  $R, G, B$  del pixel  $P$  in the image  $I(u, v) \in [0, 255]$ .
- Ecuación de histograma adaptativo limitada por contraste CLAHE con un  $kernel = 4x4$ : un histograma  $h$  para una imagen en escala de grises  $I(u, v)$  se



(a) RV con señalamiento bien marcado

(b) Conjunto de imágenes de vialidades

Figura 5.6: Imágenes de vialidades con y sin reductores de velocidad.

calcula mediante la Ecuación 5.3:

$$h(i) = \text{card}\{(u, v) | I(u, v) = i\}. \quad (5.3)$$

donde  $h(i)$  = número de píxeles en  $I$  con el valor de intensidad  $i$ . La ecualización del histograma (HE) se obtiene del histograma acumulativo  $H$  de la siguiente manera:  $H(i) = \sum_{j=0}^i h(j), 0 \leq i < 255$ . La aplicación de CLAHE mejora el contraste de las imágenes dividiendo cada una en bloques de  $4 \times 4$  llamados mosaicos y se iguala el histograma de cada una.

- Desenfoque gaussiano con un kernel =  $5 \times 5$ : empleado para suavizar imágenes donde se aplica la función gaussiana  $G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{u^2+v^2}{2\sigma^2}\right)}$  siendo  $u, v$  las posiciones de los píxeles de la imagen en cada canal  $RGB$  y  $\sigma$  es el parámetro de la desviación estándar gaussiana.

Una vez preprocesadas las imágenes, la ROI correspondiente al señalamiento para la detección de RV por reconocimiento de patrones es segmentada. Las técnicas empleadas son:

- Umbralización Otsu: el valor umbral se calcula de tal manera que la dispersión dentro de cada segmento sea lo más pequeña posible, pero al mismo tiempo que la dispersión sea lo más alta posible entre diferentes segmentos. Según el valor umbral, las imágenes se binarizan, separando las señales de los RV de la calle que corresponde al fondo. La Figura 5.7 a) muestra el resultado de la segmentación por umbralización.
- Operadores morfológicos apertura-cerradura con kernel =  $5 \times 5$ : estos operadores se utilizan para mejorar la forma y el contorno de los señalamientos de RV. Se

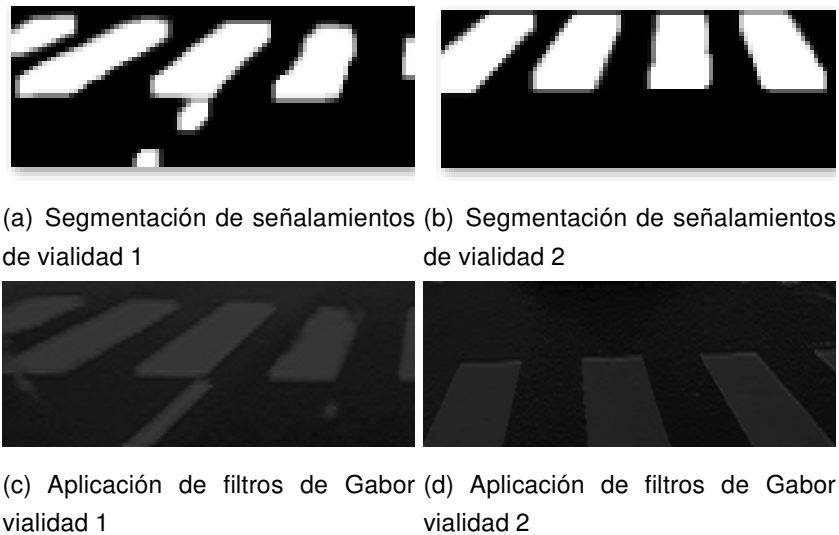
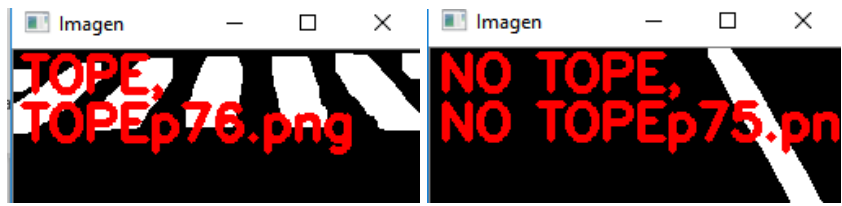


Figura 5.7: Resultados del procesamiento de imágenes.

aplica un operador de apertura que realiza la operación de erosión seguida de dilatación de la siguiente manera:  $A \circ B = (A \ominus B) \oplus B$ , donde  $A$  es una imagen binaria de entrada y  $B$  es la estructura elemento. Enseguida, se aplica un operador de cierre que realiza la dilatación seguida de la operación de erosión de la siguiente manera:  $A \bullet B = (A \oplus B) \ominus B$ . Se utiliza un kernel de  $3 \times 3$  para la aplicación de los operadores de apertura y cierre.

- Componentes conectados: permite encontrar áreas conectadas con características de color y forma similar para ayudar a delimitar la forma de los patrones de señalamiento de RV.
- Filtros de Gabor: filtro lineal cuya respuesta al impulso es una función sinusoidal que se multiplica por una función gaussiana. Se utilizan para resaltar los gradientes correspondientes al contorno del patrón de señalamiento que facilita la detección de RV.

Las imágenes obtenidas son binarizadas de forma que el color blanco indica el patrón de señalamiento del RV y el fondo negro corresponde a la región de la calle (ver ejemplos en Figura 5.7).



(a) Clasificación de imagen con RV (b) Clasificación de imagen sin RV

Figura 5.8: Resultado de la detección de RV señalizados.

## 5.4. Detección de RV señalizados

Para la clasificación de imágenes por reconocimiento de señalamientos se extraen características de un conjunto de 500 imágenes segmentadas con y sin RV señalizados mediante el cálculo de patrones binarios locales (LBP). LBP es un descriptor que calcula la representación local de texturas y se construye comparando cada pixel con sus vecinos dentro de una vecindad generando una codificación binaria que posteriormente se convierte a decimal. El histograma LBP es el vector características empleado para entrenar un modelo predictivo donde de acuerdo con su valor se etiqueta a la clase correspondiente. El conjunto de imágenes es dividido en 80 % para entrenamiento y 20 % para pruebas. Para entrenar al modelo se aplica el algoritmo de máquinas de vectores de soporte (SVM) que es un método robusto y que, a diferencia de otros algoritmos de clasificación, no requiere alto costo computacional. Se utiliza un clasificador lineal binario donde se separan las dos clases en el espacio mediante un hiperplano de separación definido como el vector entre los dos puntos. La Figura 5.8 muestra dos ejemplos de casos exitosos de la clasificación de RV señalizados, a) corresponde a un caso verdadero positivo TP y b) es un caso verdadero negativo TN.

Para medir el desempeño del modelo de detección de reductores de velocidad se calculan las matrices de confusión y algunas métricas de evaluación. El Cuadro 5.4 muestra los resultados numéricos obtenidos por el modelo, donde la precisión obtenida es de 84 %, el valor de *recall* es de 98 % y la exactitud obtenida es de 89 %. Se calcula la curva ROC (característica operativa del receptor) asociada al modelo predictivo que calcula la proporción de verdaderos positivos frente a la razón de los falsos positivos, el área bajo la curva AUC es de 0.91 (ver Figura 5.9). Por lo tanto, se muestra que el modelo presenta resultados aceptables en la detección de reductores de velocidad cuando los señalamientos están bien marcados.

Cuadro 5.1: Métricas de evaluación del modelo RV con señalamientos

Modelo	Total de instancias	Métricas	Valor
SVM	500 imágenes con y sin RV	Precisión	84 %
		Recall	98 %
		Exactitud	89 %

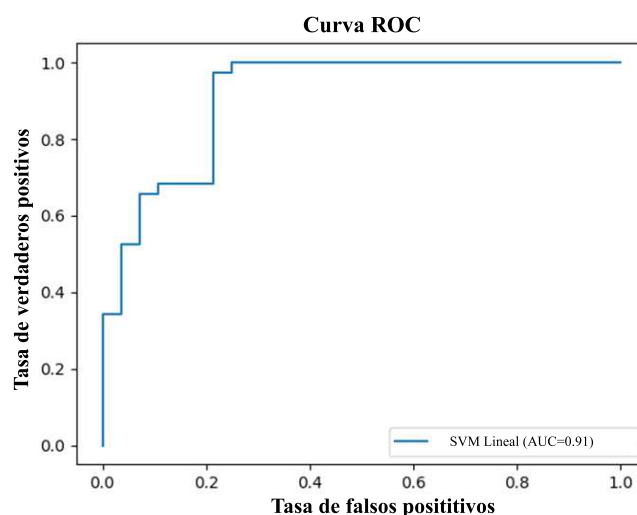


Figura 5.9: Curva ROC del modelo de clasificación de RV señalizados.

## 5.5. Generación de mapas de disparidad

Se aplica el proceso de visión estéreo al conjunto de imágenes capturadas y se obtienen los resultados que se describen a continuación. El primer paso es la calibración individual de cada cámara mediante la obtención de parámetros intrínsecos generados a partir de la captura de imágenes de un patrón asimétrico de ajedrez donde se aplica la corrección de distorsión radial y tangencial. La implementación se realiza usando el lenguaje de programación Python y la biblioteca OpenCV.

Se calculan los parámetros extrínsecos correspondientes a la matriz de rotación y traslación para mapear las coordenadas reales con las coordenadas de los puntos generados por la visión estéreo. Las matrices de calibración estéreo son utilizadas para la rectificar cada par de imágenes con el objetivo de alinearlas entre sí y de esa forma realizar la reconstrucción de imágenes estéreo. Para encontrar la correspondencia entre los pixeles de las imágenes estéreo se utiliza un enfoque de coincidencia por bloques semi-global SGBM.

A partir de las imágenes rectificadas se genera un mapa de disparidad de la calle el

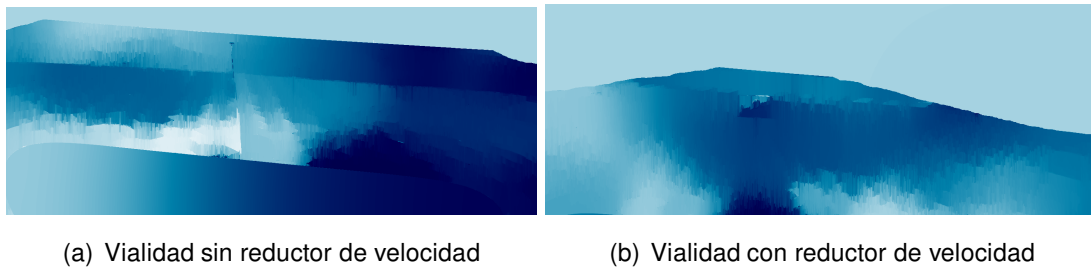


Figura 5.10: Mapas de disparidad generados.

cual es una imagen en escala de grises donde los tonos más claros corresponden a los puntos más cercanos y los más oscuros corresponden a los puntos más lejanos. Estos mapas son procesados mediante un filtro de mínimos cuadrados ponderados WLS para hacerlos más uniformes y para ajustar los puntos faltantes en la escena. WLS es un filtro de suavizado que preserva bordes y sus pesos dependen de los gradientes de la imagen.

Los parámetros utilizados para la generación de mapas de disparidad se describen a continuación:

- Distancia focal entre cámaras: 20cm
- Altura de montaje del sistema de visión: 70-90 cm
- Distancia máxima al reductor de velocidad: 50m

La Figura [5.10](#) muestra los mapas de disparidad generados a partir de la ROI de calles con los parámetros estéreo. Estos mapas permiten percibir la profundidad de las escenas, calcular distancias a diferentes puntos y generar nubes de puntos 3D.

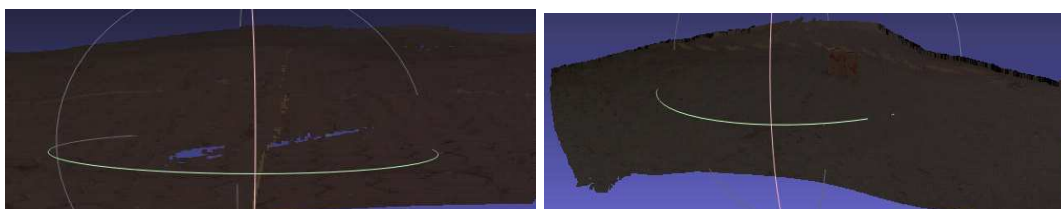
## 5.6. Generación de nubes de puntos

A partir de los mapas de disparidad generados, se aplica un proceso de triangulación para calcular la nube de puntos (X,Y,Z) correspondientes a las coordenadas de las escenas. Los rangos de las nubes de puntos, indicados en unidades de metros, se presentan en el Cuadro [5.2](#).

Las nubes de puntos son almacenadas en formato xyz y visualizadas usando el software libre MeshLab. La Figura [5.11](#) muestra la reconstrucción de calles a partir de la generación de nubes de puntos 3D.

Cuadro 5.2: Rangos de valores de las nubes de puntos en metros.

	X	Y	Z
Min	-19.58	-0.14	1.88
Max	20.00	0.35	75.26



(a) Vialidad sin reductor de velocidad

(b) Vialidad con reductor de velocidad

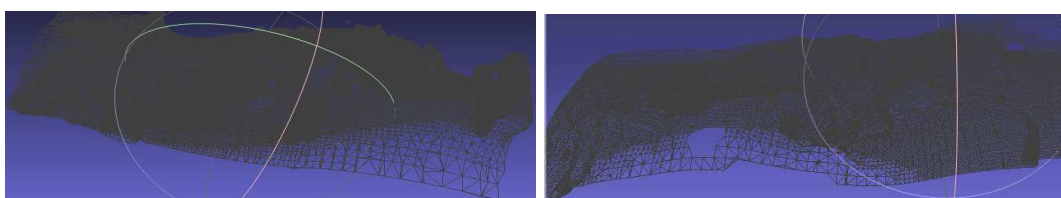
Figura 5.11: Reconstrucción 3D de nubes de puntos.

## 5.7. Creación de mallas de superficies 3D

Para convertir las nubes de puntos en mallas de superficies 3D, donde se generan triángulos a partir de tres vértices unidos por aristas, se utiliza la triangulación de Delaunay. La Figura 5.12 muestra la reconstrucción de mallas triangulares de las superficies de calles segmentadas. Estas mallas son utilizadas para el análisis de elevaciones que permitan determinar la presencia de reductores de velocidad. El tamaño total obtenido en la reconstrucción de calles es aproximadamente de 920,000 puntos.

## 5.8. Procesamiento de mallas 3D

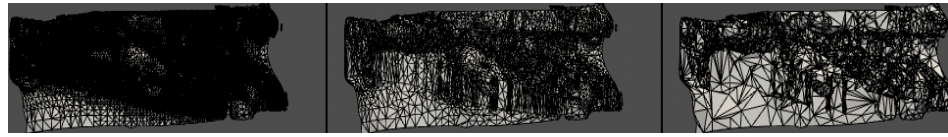
Para reducir el costo de procesamiento de las mallas generadas, se aplican dos métodos para simplificar el total de vértices: diezmar y pro diezmar. En la Figura 5.13 se visualizan



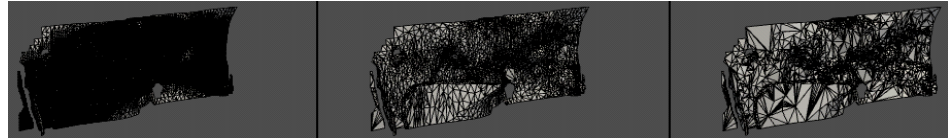
(a) Vialidad sin reductor de velocidad

(b) Vialidad con reductor de velocidad

Figura 5.12: Mallas triangulares de superficies 3D.

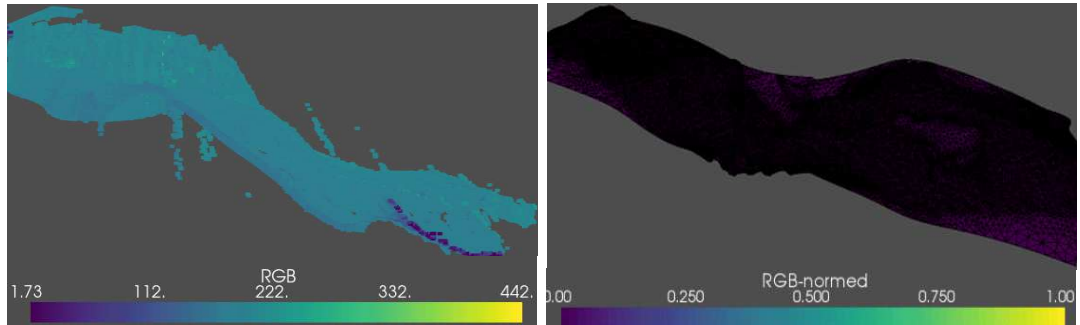


(a) Vialidad sin reductor de velocidad



(b) Vialidad con reductor de velocidad

Figura 5.13: Decimación de mallas 3D conservando su topología.



(a) Vialidad sin reductor de velocidad

(b) Vialidad con reductor de velocidad

Figura 5.14: Suavizado laplaciano de mallas de superficies 3D.

los resultados de la simplificación de mallas usando los dos métodos mencionados ambos con un porcentaje de reducción del 80 %. Cabe destacar que, a pesar de que el tamaño de las mallas se reduce, se mantiene la forma original de la superficie de las calles.

Las mallas tridimensionales se procesan para hacerlas más uniformes y para eliminar valores atípicos. Se aplica un suavizado laplaciano para suavizar mallas poligonales donde se elige una nueva posición para cada vértice de acuerdo con la información local de sus vecinos. La Ecuación 5.4 calcula el laplaciano para suavizar mallas 3D.

$$\bar{x}_i = \frac{1}{N} \sum_{j=1}^N x_j \quad (5.4)$$

donde  $N$  = número de vértices adyacentes al nodo  $i$ ,  $x_j$  es la posición del  $j$ -ésimo vértice adyacente y  $\bar{x}_i$  es la nueva posición del nodo  $i$ . La Figura 5.14 muestra el resultado del suavizado de mallas 3D donde los vértices ruidosos son eliminados de las superficies.



## 5.9. Extracción de características

A partir de los vértices de mallas de superficies 3D se selecciona un conjunto de características basadas en valores propios donde las características seleccionadas, considerando solo las que presentan diferencias significativas entre los valores de las mallas con y sin RV, son: linealidad, planaridad, esfericidad, curvaturas y anisotropía; la expresión de cada una es mostrada en el Cuadro 5.3. La Figura 5.15 muestra algunos ejemplos de los histogramas de características extraídos de mallas de superficies 3D.

Cuadro 5.3: Selección de características basadas en valores propios para mallas 3D

Característica	Expresión
Linealidad	$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planaridad	$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1}$
Esfericidad	$S_\lambda = \frac{\lambda_3}{\lambda_1}$
Curvaturas	$C_\lambda = \frac{\lambda_3}{\sum \lambda}$
Anisotropía	$A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1}$

$\lambda_1, \lambda_2$ : valores propios de puntos 3D,  
 $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$   
 $\lambda_1 + \lambda_2 + \lambda_3 = 1$

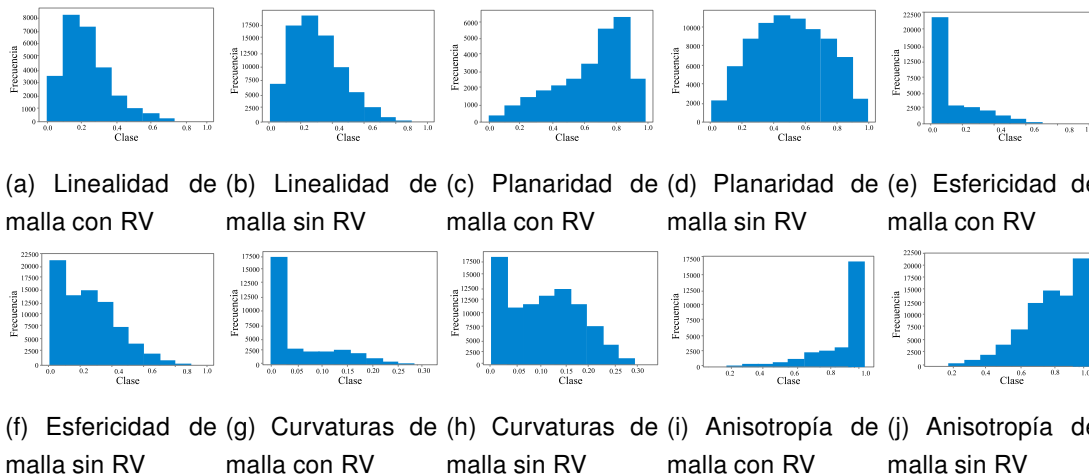
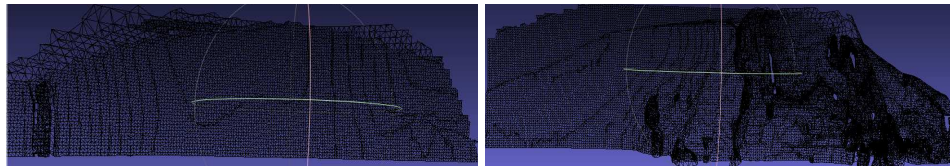


Figura 5.15: Histogramas de características de mallas 3D.

## 5.10. Detección de RV sin señalamientos

Para la detección de RV no señalizados se extraen características de un conjunto de 500 mallas 3D procesadas de superficies de calles con y sin RV. A partir del conjunto total



(a) Malla sin reductor de velocidad

(b) Malla con reductor de velocidad

Figura 5.16: Resultado de la clasificación de RV sin señalamiento.

de imágenes se utiliza el 80 % para entrenamiento y el 20 % para pruebas. Se aplica el algoritmo de máquinas de vectores de soporte para entrenar el modelo de clasificación y se utiliza un clasificador lineal binario donde se separan las dos clases de imágenes (con y sin RV) mediante un hiperplano. La Figura 5.16 muestra dos ejemplos de mallas de superficies donde el resultado de la clasificación de RV no señalizados es exitosa. Para medir el desempeño del modelo de detección de reductores de velocidad sin señalamientos se calculan las matrices de confusión y métricas de evaluación. El Cuadro 5.10 muestra los resultados numéricos obtenidos por el modelo, donde se obtiene el 77 % de precisión, el 82 % de *recall* y el 78 % de exactitud. Además, se calcula la curva ROC asociada al modelo predictivo donde se calcula la proporción de verdaderos positivos frente a la tasa de falsos positivos y el área bajo la curva obtenida es de 0.74 (ver Figura 5.17).

Cuadro 5.4: Métricas de evaluación del modelo RV sin señalamientos

Model	Total de instancias	Métricas	Valor
SVM	500 mallas 3D con y sin RV	Precisión	77 %
		Recall	82 %
		Exactitud	78 %

## 5.11. Discusión

Los tiempos de ejecución promedio de las tareas del Algoritmo 1 para cada par de imágenes estéreo, usando un equipo de cómputo con procesador AMD A10-8700P Radeon R6 1.8 GHz, 8 GB de RAM y sistema operativo Windows, son: segmentación de calles 0.45 seg, preprocesamiento de imágenes 2D 0.01 seg, detección de RV señalizados 0.04 seg, generación de mapas de disparidad 9.84 seg, generación de nubes de puntos 39.18 seg, generación de mallas de superficies 2.98 seg, procesamiento de mallas 3D 4.33 seg, extracción de características en mallas 3D 3.55

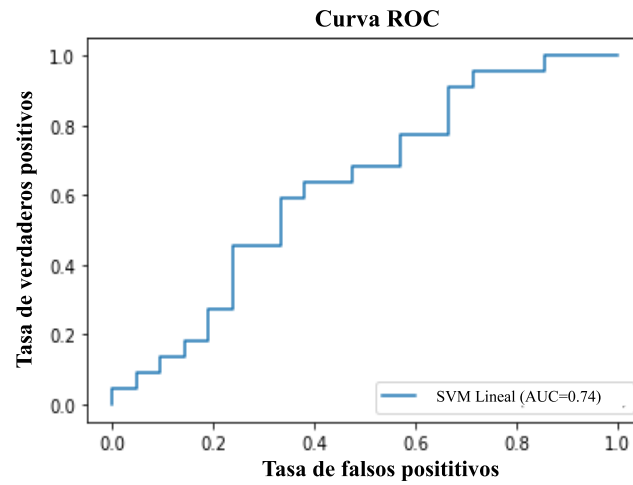


Figura 5.17: Curva ROC del modelo de clasificación de RV sin señalamientos.

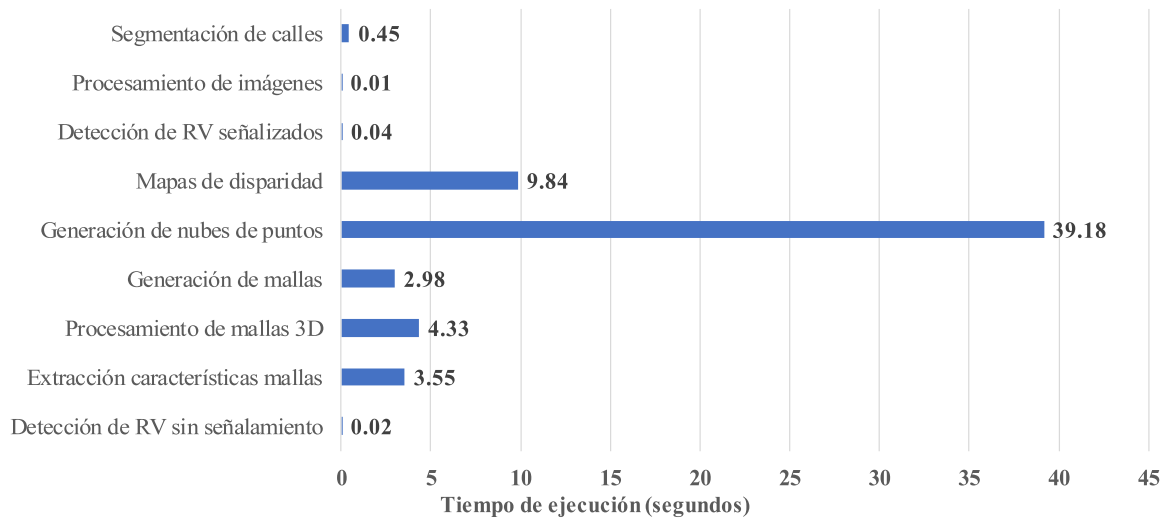


Figura 5.18: Tiempos de ejecución promedio.

---

seg y detección de RV sin señalamiento 0.02 seg, donde el tiempo promedio total es 60.37 segundos (ver Figura 5.18). Este tiempo es grande a pesar de que la tarea de generación de nubes de puntos solo reconstruye la región de interés de la calle y no toda la escena completa.

Los reductores de velocidad se presentan principalmente en vialidades de zonas urbanas para controlar la velocidad de los automóviles, por ello, se consideran las velocidades máximas para escenarios reales del Cuadro 5.11 de acuerdo con el reglamento de tránsito y vialidad del estado de Puebla; estas velocidades se encuentran en unidades de  $km/h$  y son convertidas a unidades de  $m/s$ . En este cuadro se muestra la distancia de detención mínima como la suma de la distancia de reacción más la distancia de frenado. En la última columna se estima el tiempo de detención mínimo que los autos requieren para detectar a tiempo un RV sin afectar su estabilidad. Este tiempo se calcula a partir de la fórmula  $d = vt$ , donde  $d = distancia$ ,  $v = velocidad$  y  $t = tiempo$ . Despejando el tiempo se obtiene  $t = d/v$  de forma que el tiempo mínimo real requerido para la detección de RV es de 2.88 seg para cualquier velocidad, el cual es menor que el tiempo total de ejecución del algoritmo propuesto.

El tiempo total de ejecución del algoritmo está sujeto a la arquitectura del equipo de cómputo empleada. El procesador AMD A10-8700P utilizado para los experimentos es un iGPU (unidad de proceso de gráficos integrada) de gama media con un rendimiento de punto flotante de precisión simple FP32 de 614 GFLOPS, lo cual indica la cantidad de miles de millones de operaciones de coma flotante que un procesador puede realizar por segundo [cpu-monkey \(2022\)](#). Los tiempos de ejecución podrían reducirse sustancialmente al emplear equipos con GPU debido a que los gráficos son optimizados por el uso de tarjetas gráficas dedicadas. El empleo de GPUs como EVGA GeForce RTX 3090 kingpin hybrid gaming, el cual tiene un rendimiento FP32 de 40.30 TFLOPS = 40,300 GFLOPS, podría reducir el tiempo de ejecución hasta en un 65.63% [gpu-monkey \(2022\)](#).

Adicionalmente, si se ejecuta un *kernel* 2D de CUDA en una GPU sobre los mapas de disparidad, donde en la dimensión 1 se almacenan los índices de la nube de puntos y en la dimensión 2 la clase a la que pertenecen, además, si se paralelizan cada una de las coordenadas 3D  $(x, y, z)$  y las normales de mallas triangulares, ya que son independientes entre sí, se puede reducir aún más el tiempo de ejecución. Para ello, se puede hacer uso de la herramienta OpenCL, donde se pueden crear aplicaciones con paralelismo a nivel de datos y tareas la cual es muy empleada para optimizar la generación de nubes de puntos 3D. De esta forma, el algoritmo podría ser eficiente para ejecutarse en tiempo real ya que los reductores de velocidad se podrían detectar a

Cuadro 5.5: Tiempo de detención mínimo de RV para distintas velocidades

Límites de vel.	Tipos de vialidades	Dist. de reacción	Dist. de frenado	Dist. de detención	Tiempo de deten.
30km/h= 8.3m/s	Zonas de tránsito ligero o residenciales	9m	15m	24m	2.88s
40km/h= 11.1m/s	Calles, calzadas y avenidas	12m	20m	32m	2.88s
60km/h= 16.6m/s	Avenidas con carriles divididos con camellón	18m	30m	48m	2.88s
80km/h= 22.2m/s	Carriles centrales de vías de acceso controlado	24m	40m	64m	2.88s

tiempo evitando accidentes o daños en los vehículos.

Los resultados obtenidos muestran que la metodología propuesta es capaz de detectar tanto RV señalizados como RV sin señalamientos. Para reducir el costo computacional se segmenta la región de interés correspondiente a la calle para lo cual se aplica un modelo preentrenado de una FCN. Los resultados de la segmentación muestran una precisión alta con un IoU medio del 91.6% y el índice Dice medio del 93.8%. Cabe señalar que el modelo ha sido probado con imágenes sin la restricción de que los límites de la calle estén marcados.

Existen algunos trabajos que han abordado la detección de RV señalizados donde se aplican técnicas basadas de imágenes (TBI), aprendizaje automático (AA) y visión estéreo (VE); el Cuadro 5.11 muestra una comparación de los resultados obtenidos en este trabajo, sobre la detección de RV señalizados, con los resultados reportados en los trabajos relacionados. El trabajo de Irhebhude et al. (2019) combina las técnicas TBI y AA donde, a pesar de obtener una exactitud del 100%, solo se consideran calles asfaltadas; Varma et al. (2018) combina las técnicas de visión estéreo y aprendizaje profundo donde se obtiene una exactitud del 97.44%, sin embargo, las cámaras ZED empleadas son de corto alcance y no funcionan para exteriores; Babu et al. (2020) y Devapriya et al. (2015) aplican TBI donde obtienen una exactitud del 90% y 85% respectivamente, sin embargo, dado que no se aplican algoritmos de aprendizaje automático, la detección puede causar errores altos debido a las variaciones de color y de iluminación en las calles. En el presente trabajo se combinan TBI y AA, obteniendo una exactitud para RV señalizados del 89%, lo cual es aceptable con respecto al trabajo relacionado. A diferencia de los trabajos relacionados, en este trabajo se consideran

Cuadro 5.6: Comparación de la detección de RV señalizados con otros enfoques

<b>Autor</b>	<b>Técnicas</b>	<b>Exactitud</b>	<b>Consideraciones</b>
<a href="#">Irhebhude et al. (2019)</a>	TBI, AA	100 %	Solo consideran calles de asfalto
<a href="#">Varma et al. (2018)</a>	VE, AA	97.44 %	Las cámaras ZED usadas no funcionan en distancias largas
<a href="#">Babu et al. (2020)</a>	TBI	90 %	No se basa en AA
Trabajo actual	TBI, AA	89 %	Funciona en escala real
<a href="#">Devapriya et al. (2015)</a>	TBI	85 %	No se basa en AA

varios colores y tipos de materiales de las calles como: asfalto, pavimento y adoquín. Además, en este trabajo se utilizan videocámaras de largo alcance que funcionan en escenarios a escala real.

La detección de RV sin señalamientos ha sido estudiada recientemente en la literatura; en el Cuadro 5.11 se presenta una comparación de los resultados obtenidos en este trabajo, sobre la detección de RV sin señalamientos, con otros trabajos relacionados. En el trabajo de Dewangan et al. [Dewangan & Sahu \(2020\)](#) se usa un modelo de aprendizaje profundo para la detección y se reporta una exactitud del 98.54 % que, a pesar de que es muy alta, no funciona a escala real. En el trabajo de Babu et al. [Babu et al. \(2021\)](#) se aplican TBI, donde se muestra una exactitud del 95 %. En el trabajo de Varma et al. [Varma et al. \(2018\)](#) se presenta una exactitud del 94 % al aplicar el proceso de visión estéreo, sin embargo, se utiliza una cámara ZED que es de corto alcance y no funciona en escala real. El resultado obtenido en el presente trabajo, sobre la detección de RV sin señalamientos, presenta una exactitud del 78 % que, a pesar de estar por debajo de los trabajos relacionados, considera condiciones del entorno distintas. A diferencia de algunos trabajos relacionados, donde se emplean conjuntos de imágenes a escala miniatura o donde se emplean cámaras de corto alcance, la metodología propuesta funciona en entornos a escala real debido a que las videocámaras empleadas son de largo alcance. Además, a diferencia del trabajo relacionado, donde solo se consideran calles de asfalto, en este trabajo se consideran diversos tipos de materiales en las calles (como adoquín, pavimento y asfalto) debido a que la detección de RV por elevaciones en las superficies es independiente del material de la calle y es más precisa que la detección por reconocimiento de patrones cuando los señalamientos no están pintados.

Cuadro 5.7: Comparación de la detección de RV sin señalamientos con otros enfoques

<b>Autor</b>	<b>Técnicas</b>	<b>Exactitud</b>	<b>Consideraciones</b>
<a href="#">Dewangan &amp; Sahu (2020)</a>	AP	98.54 %	No funciona a escala real
<a href="#">Babu et al. (2021)</a>	TBI	95 %	Solo consideran calles de asfalto
<a href="#">Varma et al. (2018)</a>	VE, AA	94 %	Las cámaras ZED usadas no funcionan en distancias largas
Trabajo actual	VE, AA	78 %	Funciona en escala real y con varios tipos de material de las calles

## 5.12. Síntesis

Se ha mostrado que la segmentación semántica de calles con un modelo FCN de aprendizaje profundo presenta resultados con alta precisión bajo condiciones controladas del ambiente. Se ha construido un modelo de clasificación de RV usando técnicas de procesamiento de imágenes para segmentar los señalamientos y máquinas de vectores de soporte para entrenar al modelo y se obtienen aceptables cuando el patrón de señalamiento está bien marcado. En contraste, cuando el RV no tiene señalamientos marcados se reconstruyen mallas de superficies 3D mediante visión estéreo y se entrena un modelo usando también máquinas de vectores de soporte para detectar RV por elevaciones en las superficies a partir de la extracción de características basadas en valores propios. A pesar de que los resultados están por debajo de los resultados reportados por los enfoques relacionados, en este trabajo se consideran ambientes urbanos y semiurbanos a escala real con varios tipos de materiales de las calles a diferencia de los trabajos relacionados.

# Capítulo 6

## Conclusiones

En este trabajo se presentó una metodología para detectar reductores de velocidad tanto señalizados como no señalizados para la toma de decisiones de vehículos autónomos. Para ello, se captura un conjunto de imágenes estéreo con un sistema de visión. Se realiza una segmentación semántica de la calle correspondiente a la región de interés aplicando un modelo preentrenado basado en una red neuronal convolucional FCN. Los resultados presentan alta precisión con una media de IoU de 91.6% y una media del índice Dice de 93.08%.

Se ha entrenado un modelo para detectar RV señalizados donde se extraen características de histogramas de patrones binarios locales a partir de las imágenes y se aplica el algoritmo de máquinas de vectores de soporte para la clasificación. Los resultados de la clasificación son aceptables con una precisión del 84%, *recall* del 98%, una exactitud del 89% y el AUC del 91%.

Para la detección de RV sin señalamientos marcados se aplica el proceso de visión estéreo para generar mapas de disparidad entre pares de imágenes y en consecuencia hacer la reconstrucción 3D de calles segmentadas mediante nubes de puntos. Las nubes de puntos se convierten en mallas de superficies triangulares aplicando una triangulación de Delaunay. Las mallas son procesadas para eliminar datos atípicos y para simplificar el tamaño de las mallas. Se ha entrenado un modelo para detectar RV sin señalamientos donde se extraen características basadas en valores propios a partir de las mallas de superficies. Se entrena un modelo para clasificar mallas aplicando el algoritmo de máquinas de vectores de soporte donde los resultados obtenidos son: precisión de 77%, *recall* del 82%, exactitud del 78% y el AUC del 74%.

A pesar de que los resultados sobre la detección tanto de RV señalizados como sin señalamiento presentan un error relativamente alto, la metodología propuesta mejora las



técnicas existentes ya que considera ambientes urbanos y semiurbanos a escala real y es independiente del material de las calles.

Los resultados obtenidos tienen una importante contribución en la detección de irregularidades en las superficies de calles que afectan la seguridad y el confort para la navegación de vehículos autónomos. Además, la metodología propuesta mejora las técnicas existentes al considerar ambientes urbanos y semiurbanos con varios tipos de material de las calles, asimismo, los resultados en la detección son aceptables con respecto a los trabajos relacionados. El tiempo promedio total de ejecución del algoritmo para cada imagen es de 60.37 segundos; a pesar de que el algoritmo no es eficiente para ejecutarse en tiempo real con el equipo de cómputo empleado, ya que la reconstrucción 3D es una tarea con alto costo computacional por la cantidad de datos generados, el tiempo total puede reducirse sustancialmente al emplear equipos de cómputo con GPU y la paralelización de las coordenadas 3D y las normales de las mallas triangulares. Cabe mencionar que la segmentación de calles reduce el tiempo de procesamiento debido a que solo se hace la reconstrucción 3D de la región de interés.

Como trabajo a futuro se espera mejorar la exactitud en la detección de RV sin señalamientos mediante la aplicación de técnicas como *crowdsourcing* que permitan recuperar la ubicación geográfica de los RV evitando cálculos innecesarios pero que en combinación con la visión estéreo logren detectar de forma precisa los RV. Además, es posible estimar las distancias a los RV para que la velocidad de los vehículos pueda ser ajustada y de esa forma evitar daños o accidentes y reducir el tiempo total de ejecución del algoritmo.

# Referencias

- Afrin, M., Mahmud, M. R., & Razzaque, M. A. (2015). Real time detection of speed breakers and warning system for on-road drivers. In *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, (pp. 495–498). IEEE.
- Al-Shargabi, B., Hassan, M., & Al-Rousan, T. (2020). A novel approach for the detection of road speed bumps using accelerometer sensor. *Technology education management informatics*, *9*(2), 469–476.
- Alam, M. Y., Nandi, A., Kumar, A., Saha, S., Saha, M., Nandi, S., & Chakraborty, S. (2020). Crowdsourcing from the true crowd: Device, vehicle, road-surface and driving independent road profiling from smartphone sensors. *Pervasive and Mobile Computing*, *61*, 101103.
- Alvarez, J. M., Gevers, T., & Lopez, A. M. (2010). 3d scene priors for road detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 57–64). IEEE.
- Alvarez, J. M., Lopez, A., & Baldrich, R. (2008). Illuminant-invariant model-based road segmentation. In *2008 IEEE Intelligent Vehicles Symposium*, (pp. 1175–1180). IEEE.
- Arunpriyan, J., Sajith Variyar, V., & Adarsh, S. (2019). Real-time speed bump detection using image segmentation for autonomous vehicles. *Intelligent Computing, Information and Control Systems: ICICCS 2019*, *1039*, 308.
- Astarita, V., Caruso, M. V., Danieli, G., Festa, D. C., Giofrè, V. P., Luele, T., & Vaiana, R. (2012). A mobile application for road surface quality control: Uniquallroad. *Procedia-Social and Behavioral Sciences*, *54*, 1135–1144.
- Babu, C. N. K., Priya, W. D., Srihari, T., & Nandakumar, R. (2020). Speed-bump detection

- using otsu's algorithm and morphological operation. *International Journal on Emerging Technologies*, 11(3), 989–994.
- Babu, N., Priya, W. D., & Srihari, T. (2021). Real-time detection of unmarked speed bump for indian roads. *European Journal of Molecular & Clinical Medicine*, 7(5), 1745–1752.
- Badino, H., Franke, U., & Pfeiffer, D. (2009). The stixel world-a compact medium level representation of the 3d-world. In *Joint Pattern Recognition Symposium*, (pp. 51–60). Springer.
- Bahena, J. M. R., Cruz, C. A., Lopez, A. Z., & Ferreyra-Ramirez, A. (2010). Speed booms detection for a ground vehicle with computer vision. In *Proceedings of the 12th WSEAS international conference on Mathematical and computational methods in science and engineering*, (pp. 258–264). World Scientific and Engineering Academy and Society (WSEAS).
- Beucher, S. (1992). The watershed transformation applied to image segmentation. *Scanning Microscopy*, (6), 28.
- Beucher, S. & Bilodeau, M. (1994). Road segmentation and obstacle detection by a fast watershed transformation. In *Proceedings of the Intelligent Vehicles' 94 Symposium*, (pp. 296–301). IEEE.
- Beucher, S., Bilodeau, M., & Yu, X. (1990). Road segmentation by watershed algorithms. In *Prometheus Workshop, Sophia Antipolis, France*.
- Broggi, A., Cardarelli, E., Cattani, S., & Sabbatelli, M. (2013). Terrain mapping for off-road autonomous ground vehicles using rational b-spline surfaces and stereo vision. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, (pp. 648–653). IEEE.
- Buehler, M., Iagnemma, K., & Singh, S. (2009). *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. Springer.
- Burger, W. & Burge, M. J. (2010). *Principles of digital image processing: fundamental techniques*. Springer Science & Business Media.
- Caltagirone, L., Scheidegger, S., Svensson, L., & Wahde, M. (2017). Fast lidar-based road detection using fully convolutional neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, (pp. 1019–1024). IEEE.

- 
- Campbell, N. W., Thomas, B. T., & Troscianko, T. (1997). Automatic segmentation and classification of outdoor images using neural networks. *International Journal of Neural Systems*, 8(01), 137–144.
- Cavalli, M., Tarolli, P., Marchi, L., & Dalla Fontana, G. (2008). The effectiveness of airborne lidar data in the recognition of channel-bed morphology. *Catena*, 73(3), 249–260.
- Chen, H.-T., Lai, C.-Y., Hsu, C.-C., Lee, S.-Y., Lin, B.-S. P., & Ho, C.-P. (2014). Vision-based road bump detection using a front-mounted car camcorder. In *2014 22nd International Conference on Pattern Recognition*, (pp. 4537–4542). IEEE.
- Chen, Y., Zhou, M., Zheng, Z., & Huo, M. (2019). Toward practical crowdsourcing-based road anomaly detection with scale-invariant feature. *IEEE Access*, 7, 67666–67678.
- Choi, J., Lee, J., Kim, D., Soprani, G., Cerri, P., Broggi, A., & Yi, K. (2012). Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(2), 974–982.
- Choong, C. S., Nasir, A. F. A., Majeed, A. P. A., Zakaria, M. A., & Razman, M. A. M. (2020). Machine learning approach in identifying speed breakers for autonomous driving: An overview. In *RITA 2018* (pp. 409–424). Springer.
- Chugh, G., Bansal, D., & Sofat, S. (2014). Road condition detection using smartphone sensors: A survey. *International Journal of Electronic and Electrical Engineering*, 7(6), 595–602.
- Corsini, M., Cignoni, P., & Scopigno, R. (2012). Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE transactions on visualization and computer graphics*, 18(6), 914–924.
- cpu-monkey (enero de 2022). [www.cpu-monkey.com/es/cpu-amd\\_a10\\_8700p-1193](http://www.cpu-monkey.com/es/cpu-amd_a10_8700p-1193).
- Cristianini, N., Shawe-Taylor, J., et al. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.
- Cyganek, B. & Siebert, J. P. (2011). *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons.

- Danti, A., Kulkarni, J., & Hiremath, P. (2013). A technique for bump detection in indian road images using color segmentation and knowledge base object detection. *International Journal of Scientific & Engineering Research*, 4(8), 2229–5518.
- Daraghmi, Y.-A. & Daadoo, M. (2016). Intelligent smartphone based system for detecting speed bumps and reducing car speed. In *MATEC Web of Conferences*, volume 77, (pp. 09006). EDP Sciences.
- Darwiche, M. & El-Hajj-Chehade, W. (2019). Speed bump detection for autonomous vehicles using signal-processing techniques. *BAU Journal-Science and Technology*, 1(1), 5.
- de la Escalera, A., Armingol, J. M., Pech, J. L., & Gómez, J. J. (2010). Detección automática de un patrón para la calibración de cámaras. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 7(4), 83–94.
- Devapriya, W., Babu, C. N. K., & Srihari, T. (2015). Advance driver assistance system (adas)-speed bump detection. In *2015 IEEE international conference on computational intelligence and computing research (ICCIC)*, (pp. 1–6). IEEE.
- Devapriya, W., Babu, C. N. K., & Srihari, T. (2016). Real time speed bump detection using gaussian filtering and connected component approach. *Circuits and Systems*, 7, 2168–2175.
- Dewangan, D. K. & Sahu, S. P. (2020). Deep learning-based speed bump detection model for intelligent vehicle system using raspberry pi. *IEEE Sensors Journal*, 21(3), 3570–3578.
- Dey, M. R., Satapathy, U., Bhanse, P., Mohanta, B. K., & Jena, D. (2019). Magtrack: detecting road surface condition using smartphone sensors and machine learning. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, (pp. 2485–2489). IEEE.
- Edwan, E., Sarsour, N., & Alatrash, M. (2019). Mobile application for bumps detection and warning utilizing smartphone sensors. In *2019 International Conference on Promising Electronic Technologies (ICPET)*, (pp. 50–54). IEEE.
- Fekry, M., Hamdy, A., & Atia, A. (2013). Anti-bump: A bump/pothole monitoring and broadcasting system for driver awareness. In *International Conference on Human-Computer Interaction*, (pp. 561–570). Springer.

- 
- Fernández, C., Gavilán, M., Llorca, D. F., Parra, I., Quintero, R., Lorente, A. G., Vlacic, L., & Sotelo, M. (2012). Free space and speed humps detection using lidar and vision for urban autonomous navigation. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, (pp. 698–703). IEEE.
- Foucher, P., Sebsadji, Y., Tarel, J.-P., Charbonnier, P., & Nicolle, P. (2011). Detection and recognition of urban road markings using images. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, (pp. 1747–1752). IEEE.
- Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11), 1231–1237.
- gpu-monkey (enero de 2022). [www.gpu-monkey.com/es/gpu\\_benchmark-fp32-3](http://www.gpu-monkey.com/es/gpu_benchmark-fp32-3).
- Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., & Yang, B. (2014). Using mobile laser scanning data for automated extraction of road markings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 93–107.
- Hamzah, R. A. & Ibrahim, H. (2016). Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016.
- Harikrishnan, P. & Gopi, V. P. (2017). Vehicle vibration signal processing for road surface monitoring. *IEEE Sensors Journal*, 17(16), 5192–5197.
- Hernández, J. & Marcotegui, B. (2009). Filtering of artifacts and pavement segmentation from mobile lidar data. In *ISPRS Workshop Laserscanning 2009*.
- Hernández Orallo, J., Ferri Ramirez, C., & Ramirez Quintana, M. J. (2004). *Introducción a la Minería de Datos*. Pearson Prentice Hall,.
- Irhebhude, M., Adeyemi, O., & Kolawole, A. (2019). Speed breakers, road marking detection and recognition using image processing techniques. *European Journal of Applied Sciences*, 7(5), 30–42.
- Jain, M., Singh, A. P., Bali, S., & Kaul, S. (2012). Speed-breaker early warning system. In *NSDR*.
- Jain, R., Kasturi, R., & Schunck, B. G. (1995). *Machine vision*, volume 5. McGraw-Hill New York.

- Joon, J. H., Renata, D. A., Yoon, S. H., Choi, Y., & Jembre, Y. Z. (2019). Data mining for speed bump detection from car wheels and gps sensors using random forest. In *ICTC*, (pp. 740–743).
- Jung, S., Lee, U., Jung, J., & Shim, D. H. (2016). Real-time traffic sign recognition system with deep convolutional neural network. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on*, (pp. 31–34). IEEE.
- Kavitha, D. & Ravikumar, S. (2021). Designing an iot based autonomous vehicle meant for detecting speed bumps and lanes on roads. *Journal of Ambient Intelligence and Humanized Computing*, 12(7), 7417–7426.
- Khan, J. F., Bhuiyan, S. M., & Adhami, R. R. (2011). Image segmentation and shape analysis for road-sign detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(1), 83–96.
- Kiran, A. G. & Murali, S. (2014). Automatic hump detection and 3d view generation from a single road image. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (pp. 2232–2238). IEEE.
- Klette, R., Schluns, K., & Koschan, A. (2001). *Computer vision: three-dimensional Data from Images*. Springer-Verlag New York.
- Kong, H., Audibert, J.-Y., & Ponce, J. (2009). Vanishing point detection for road detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 96–103). IEEE.
- Lanjewar, B., Khedkar, J., Sagar, R., Pawar, R., & Gosavi, K. (2015). Survey of road bump and intensity detection algorithms using smartphone sensors. *International Journal of Computer Science and Information Technologies*, 6(6), 5133–5136.
- Lee, J.-H., Kim, H.-J., Cho, B.-J., Choi, J.-H., & Kim, Y.-J. (2018). Road bump detection using lidar sensor for semi-active control of front axle suspension in an agricultural tractor. *IFAC-PapersOnLine*, 51(17), 124–129.
- Lee, J.-K. & Yoon, K.-J. (2018). Temporally consistent road surface profile estimation using stereo vision. *IEEE Transactions on Intelligent Transportation Systems*, 19(5), 1618–1628.
- LIDAR, V. (2018). *Velodyne LIDAR*.

- 
- Lillo-Castellano, J., Mora-Jiménez, I., Figuera-Pozuelo, C., & Rojo-Álvarez, J. L. (2015). Traffic sign segmentation and classification using statistical learning methods. *Neurocomputing*, *153*, 286–299.
- Lion, K. M., Kwong, K. H., & Lai, W. K. (2018). Smart speed bump detection and estimation with kinect. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, (pp. 465–469). IEEE.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 3431–3440).
- Madli, R., Hebbar, S., Pattar, P., & Golla, V. (2015). Automatic detection and notification of potholes and humps on roads to aid drivers. *IEEE sensors journal*, *15*(8), 4313–4318.
- Maldonado-Bascón, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gómez-Moreno, H., & López-Ferreras, F. (2007). Road-sign detection and recognition based on support vector machines. *IEEE transactions on intelligent transportation systems*, *8*(2), 264–278.
- Manikandan, B. & Bharathi, M. (2018). Speed breaker detection using blob analysis. *International Journal of Pure and Applied Mathematics*, *118*(20), 3671–3677.
- Melo, S., Marchetti, E., Cassidy, S., Hoare, E., Bogoni, A., Gashinova, M., & Cherniakov, M. (2018). 24 ghz interferometric radar for road hump detections in front of a vehicle. In *2018 19th International Radar Symposium (IRS)*, (pp. 1–9). IEEE.
- Mery, D. (2015). Computer vision for x-ray testing. *Switzerland: Springer International Publishing*, *10*, 978–3.
- Mohit, S., Rezaul, K. M., & Pradeep, P. (2018). *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python*. Packt Publishing Ltd.
- Mukherjee, A. & Majhi, S. (2016). Characterisation of road bumps using smartphones. *European Transport Research Review*, *8*(2), 13.
- Müller, A. C. & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. O Reilly Media, Inc.
- Nixon, M. & Aguado, A. S. (2012). *Feature extraction and image processing for computer vision*. Academic Press.



- Oniga, F. & Nedevschi, S. (2011). Curb detection for driving assistance systems: A cubic spline-based approach. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, (pp. 945–950). IEEE.
- Patterson, J. & Gibson, A. (2017). *Deep learning: A practitioner's approach*. O'Reilly Media, Inc."
- Perttunen, M., Mazhelis, O., Cong, F., Kaupilla, M., Leppänen, T., Kantola, J., Collin, J., Pirttikangas, S., Haverinen, J., Ristaniemi, T., et al. (2011). Distributed road surface condition monitoring using mobile phones. In *International Conference on Ubiquitous Intelligence and Computing*, (pp. 64–78). Springer.
- Prieto, M. S. & Allen, A. R. (2009). Using self-organising maps in the detection and recognition of road signs. *Image and Vision Computing*, 27(6), 673–683.
- Rasmussen, C. et al. (2004). Texture-based vanishing point voting for road shape estimation. In *BMVC*, (pp. 1–10). Citeseer.
- Salau, H. B., Onumany, A. J., AB Nu, A. M., Onwuka, E. N., Duk Ya, J. J., & ZE, H. (2019). A survey of accelerometer-based techniques for road anomalies detection and characterization. *International Journal of Engineering Science and Application*, 3(1), 8–20.
- Sebsadji, Y., Tarel, J.-P., Foucher, P., & Charbonnier, P. (2010). Robust road marking extraction in urban environments using stereo images. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, (pp. 394–400). IEEE.
- Serna, A. & Marcotegui, B. (2014). Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 243–255.
- Shah, S. & Deshmukh, C. (2019). Pothole and bump detection using convolution neural networks. In *2019 IEEE Transportation Electrification Conference (ITEC-India)*, (pp. 1–4). IEEE.
- Silva, N., Soares, J., Shah, V., Santos, M. Y., & Rodrigues, H. (2017). Anomaly detection in roads with a data mining approach. *Procedia computer science*, 121, 415–422.
- Singh, G., Bansal, D., Sofat, S., & Aggarwal, N. (2017). Smart patrolling: An efficient road surface monitoring using smartphone sensors and crowdsourcing. *Pervasive and Mobile Computing*, 40, 71–88.

- 
- Singh, G., Kumar, R., & Kashtriya, P. (2018). Detection of potholes and speed breaker on road. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, (pp. 490–495). IEEE.
- Singh, M., Shukla, S., & Krishnan, U. (2017). Detection of humps and potholes on roads and notifying the same to the drivers. *International Journal of Management and Applied Science*, *3*(1), 130–133.
- Soheilian, B., Paparoditis, N., & Vallet, B. (2013). Detection and 3d reconstruction of traffic signs from multiple view color images. *ISPRS journal of photogrammetry and remote sensing*, *77*, 1–20.
- Soquet, N., Aubert, D., & Hautiere, N. (2007). Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. In *2007 IEEE Intelligent Vehicles Symposium*, (pp. 160–165). IEEE.
- Sotelo, M. A., Rodriguez, F. J., Magdalena, L., Bergasa, L. M., & Boquete, L. (2004). A color vision-based lane tracking system for autonomous driving on unmarked roads. *Autonomous Robots*, *16*(1), 95–116.
- Souza, V. M. (2018). Asphalt pavement classification using smartphone accelerometer and complexity invariant distance. *Engineering Applications of Artificial Intelligence*, *74*, 198–211.
- Souza, V. M., Cherman, E. A., Rossi, R. G., & Souza, R. A. (2017). Towards automatic evaluation of asphalt irregularity using smartphone's sensors. In *International Symposium on Intelligent Data Analysis*, (pp. 322–333). Springer.
- Sucgang, N. J., Ramos Jr, M., & Arriola, N. A. (2017). Road surface obstacle detection using vision and lidar for autonomous vehicle. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1.
- Sujitha, M. S., Ramesh, N., & Kotamraju, S. K. (2016). Automatic speed controlling of vehicle and detection and notification of potholes and humps. *Journal of Engineering and Applied Sciences*, *100*(9), 1921–1924.
- Tedeschi, A. & Benedetto, F. (2017). A real-time automatic pavement crack and pothole recognition system for mobile android-based devices. *Advanced Engineering Informatics*, *32*, 11–25.

- Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., & Urtasun, R. (2018). Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, (pp. 1013–1020). IEEE.
- Varma, V., Adarsh, S., Ramachandran, K., & Nair, B. B. (2018). Real time detection of speed hump/bump and distance estimation with deep learning using gpu and zed stereo camera. *Procedia computer science*, *143*, 988–997.
- Vimalkumar, K., Vinodhini, R., Archanaa, R., et al. (2017). An early detection-warning system to identify speed breakers and bumpy roads using sensors in smartphones. *International Journal of Electrical and Computer Engineering*, *7*(3), 1377.
- Vosselman, G. & Zhou, L. (2009). Detection of curbstones in airborne laser scanning data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *38*, 111–116.
- Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., & Cottrell, G. (2018). Understanding convolution for semantic segmentation. In *2018 IEEE winter conference on applications of computer vision (WACV)*, (pp. 1451–1460). IEEE.
- Wang, P., Hu, Y., Dai, Y., & Tian, M. (2017). Asphalt pavement pothole detection and segmentation based on wavelet energy field. *Mathematical Problems in Engineering*, *2017*.
- Wang, S., Pan, H., Zhang, C., & Tian, Y. (2014). Rgb-d image-based detection of stairs, pedestrian crosswalks and traffic signs. *Journal of Visual Communication and Image Representation*, *25*(2), 263–272.
- Wöhler, C. (2012). *3D computer vision: efficient methods and applications*. Springer Science & Business Media.
- Yang, B., Fang, L., Li, Q., & Li, J. (2012). Automated extraction of road markings from mobile lidar point clouds. *Photogrammetric Engineering & Remote Sensing*, *78*(4), 331–338.
- Yun, H.-S., Kim, T.-H., & Park, T.-H. (2019). Speed-bump detection for autonomous vehicles by lidar and camera. *Journal of Electrical Engineering & Technology*, *14*(5), 2155–2162.

---

Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., & Liu, W. (2016). Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing*, 214, 758–766.