Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería de Sistemas de Telecomunicaciones

Trabajo Fin de Grado

Sistema Automático de estimación de densidad de tráfico en imágenes de videovigilancia en carretera

Autor: Beatriz Torre Jiménez

Director: Roberto Javier López Sastre

TRIBUNAL:

Presidente: D. Sergio Lafuente Arroyo

Vocal 1º: D. José Carlos Nieto Borge

Vocal 2º: D. Roberto Javier López Sastre

FECHA:.....

Sistema automático de estimación de densidad de tráfico en imágenes de videovigilancia en carretera

Beatriz Torre Jiménez

22 de julio de 2014

Dime y lo olvido, enséñame y lo recuerdo,
involúcrame y lo aprendo.
-Benjamin Franklin-

Agradecimientos

Me gustaría agradecerle a mi tutor, Roberto, por brindarme la oportunidad de realizar este trabajo de Fin de Grado con él. Por su apoyo y seguimiento semanalmente en el que me ha ido guiando, incluso estando de Erasmus en Suecia. También quiero agradecerles su apoyo a mis compañeros del grupo de investigación GRAM, especialmente a Ricardo que siempre me ha ayudado y por supuesto a todos mis compañeros de clase que me han acompañado durante estos años de carrera, quiénes me han demostrado que no son sólo compañeros sino amigos.

También quiero agradecerles a mi familia el apoyo que me han dado siempre. A mis padres que siempre han creído en mi y me han levantado el ánimo en aquéllos momentos en los que pensaba que no iba a poder, han tenido más fé en mi de la que muchas veces yo he mostrado. Gracias por demostrarme que en esta vida no hay nada inalcanzable, solo hazme falta propornerse llegar a la meta y con constancia todo se consigue, sobretodo gracias por vuestra paciencia. También quiero agradecerle a mi hermano, Álvaro, todo el cariño que me ha brindado porque él siempre ha confiado en mi y me lo ha sabido demostrar dándome fuerzas para continuar cuando pensaba que no podría. Gracias a mis abuelos, que pese a la distancia siempre han estado ahí a mi lado, mostrándome toda su confianza e ilusión, ya que para ellos no hay nadie mejor que yo. A Mario por todas esas horas que hemos pasado estudiando y haciendo prácticas, por tu paciencia ya que a veces aguantarme puede ser díficil por lo cabezona que soy y sobretodo por compartir conmigo tantos buenos momentos. Al primo Luis quién siempre me ha ayudado cuando he tenido problemas con programación y me incita a aprender cosas nuevas.

Gracias a todos por vuestro apoyo incondicional y por hacer posible que alcance todas mis metas y objetivos.

Índice general

Lista de figuras

Lista de tablas

Resumen

Se ha creado un sistema de validación experimental para sistemas de estimación de la congestión del tráfico. El objetivo principal del trabajo ha sido la implementación y evaluación de un sistema de estimación precisa del número de vehículos presentes en la escena. Para ello, en lugar de detectar y localizar la posición individual de los objetos, se ha planteado un esquema que es capaz de estimar la densidad de los vehículos, de modo que, una vez obtenida la misma, la cuenta de los objetos presentes en una zona de la imagen puede aproximarse computando la integral de la densidad estimada.

Para la evaluación experimental, se ha utilizado una base de datos que hemos recopilado expresamente para este proyecto. En ella se incluyen imágenes reales, tomadas por cámaras de videovigilancia de tráfico, que han sido debidamente anotadas para evaluar las soluciones. A la vista de los resultados obtenidos, destacamos dos conclusiones. La primera es que las imágenes recopiladas suponen un gran desafío, incluso para sistemas considerados como el estado del arte en cuanto a conteo de objetos se refiere. La segunda es que para obtener unos resultados satisfactorios con el modelo implementado, resulta fundamental realizar un ajuste de los parámetros del sistema.

Palabras clave: Vehículos, tráfico, distancia MESA, densidad.

Abstract

In this work, we have developed a system for estimating the number of vehicles in images of video surveillance cameras. The goal of this project is the implementation and experimental evaluation of a system for accurately estimating the number of vehicles on the scene. For this purpose, instead of detecting and locating the position of individual objects, we have proposed a system that is able to estimate the density of the vehicles. When the estimated density is obtained, the count of the objects present in any region of the image can be approximated by computing the integral over the estimated density.

For the experimental evaluation, we have used a database specifically built for this project. It includes real images taken by traffic video surveillance cameras. These images have been properly annotated for evaluating the solutions. Regarding to the results, we highlight two conclusions. The first one is that the images collected represent a big challenge, even for systems considered the state-of- the- art in terms of counting objects. The second one is that to obtain satisfactory results with the implemented model, it is essential to make an adjustment of the system parameters.

Keywords: Vehicles, traffic, MESA distance, density.

Resumen Extendido

La idea que ha llevado a desarrollar este proyecto reside en el reto de contar objetos en imágenes. Este problema consiste en la estimación exacta del número de objetos en las mismas, viéndose considerablemente aumentada la dificultad de esta tarea cuando aparecen múltiples objetos en la escena, y por lo tanto oclusiones entre ellos. Este inconveniente se presenta en muchas aplicaciones del mundo real, como por ejemplo a la hora de monitorizar grupos de personas en los sistemas de vigilancia, en el control del nivel de congestión de las carreteras, en la realización del censo de la fauna o al contar el número de árboles en una imagen aérea de un bosque.

Autores como, por ejemplo, Lempitsky et al. [16], han propuesto recientemente una solución al problema de contar objetos en imágenes, aprendiendo sistemas que nos permiten estimar la densidad de los objetos existentes en las imágenes, y que han sido utilizados con éxito en el problema de conteo de células en imágenes de laboratorio. En este proyecto se ha analizado como funcionan las soluciones basadas en [16], en un entorno distinto, con el objetivo de contar el número de vehículos en imágenes de videovigilancia de baja resolución.

El primer paso que se ha llevado a cabo es la adaptación del código propuesto por Lempitsky et al. [16] para utilizarlo con las imágenes de tráfico. Dicho software está principalmente dividido en dos partes: entrenamiento y test. A continuación se han desarrollado las herramientas pertinentes para la obtención de las imágenes de las cámaras de tráfico que proporciona la Dirección General de Tráfico (DGT), y para la realización de la anotación de éstas. Como decimos, la adquisición de imágenes se ha realizado utilizando las cámaras que ofrece la DGT para el control de las carreteras, eligiéndose un total de 11 cámaras. Durante un periodo de tres semanas se ha realizado la captación de las imágenes necesarias y se ha llevado a cabo, así mismo, el filtrado de las mismas. Dado que la idea principal de este proyecto consiste en contar vehículos en imágenes, con la dificultad añadida de que éstas posean una alta densidad de tráfico, se ha creado una base de datos para este proyecto que ha permitido determinar el funcionamiento del sistema. Esta base de datos está constituída por las imágenes filtradas, obteniéndose un total de 1244 imágenes, que se han categorizado en tres grupos. Los grupos se han realizado en función de la semana en que las imágenes han sido capturadas y su futuro uso: las de la primera semana para el entrenamiento del sistema, las de la segunda para la validación

de los sistemas y ajustes de parámetros, y por último, las de la tercera semana para el testeo final del sistema.

Como paso previo al proceso de captación de imágenes, se llevó a cabo un estudio exhaustivo de las carreteras, pudiéndose determinar cuales de ellas eran las óptimas para el desarrollo del proyecto, así como establecer qué cámaras y qué horarios eran los más adecuados para poder proceder a la adquisición de los datos. Como consecuencia del citado análisis se estableció que las primeras horas de la mañana, entre las 7:30h y las 10:00h, cuando las carreteras poseen una mayor concentración de tráfico, eran las apropiadas. En cuanto a las cámaras, se han seleccionado principalmente las que están ubicadas en las carreteras A2, A5 y A6 a la entrada de Madrid. La Figura 1 muestra varios ejemplos de carreteras con una alta concentración de tráfico.

(a) **(b)** **(c)**

Figura 1: Ejemplos de imágenes con altas densidades de tráfico correspondientes a las horas y carreteras seleccionadas a) A2 b) A5 c) A6.

Una vez obtenidas las imágenes necesarias, y después de categorizarlas en los tres grupos (*entrenamiento, validación* y *test*), el siguiente paso consistió en seleccionar una Región de Interés (ROI) para cada imagen. Debido a que las cámaras de la DGT sufren muchos cambios (ampliación de Zoom, cambio de posición, etc.) no se podía utilizar una ROI estática. Ésta es la razón por la cual se decidió utilizar una ROI por cada imagen en lugar de una ROI global para todas ellas. Así, una vez filtradas las imágenes, y las ROIs definidas, se procedió a la anotación propiamente dicha, es decir, en cada imagen se anotaron los vehículos presentes, marcando cada uno con un punto, como muestra la la Figura 2. Una vez concluido todo el proceso se obtuvo la base de datos adecuada para nuestro sistema.

Llegados al punto en que la base de datos ha sido creada, se procedió al análisis y adaptación del sistema descrito en [16] para poder trabajar con imágenes de tráfico y resolver el problema del conteo de vehículos.

La idea descrita en [16] es muy sencilla. Dada una imagen, se genera una función de densidad de objetos, $\mathcal{F}$, que aplicada a cada píxel, produce la densidad estimada

(a) **(b)**

Figura 2: Imágenes que componen la base de datos a) Imagen original. b) Imagen con ROI y anotación.

dependiendo de su apariencia. Una vez conocida la función de densidad $\mathcal{F}$, el número total de objetos en la imagen se obtiene integrando el resultado de aplicar $\mathcal{F}$ sobre la imagen. Así mismo, si se integra la densidad sobre una subregión, perteneciente a la imagen, se obtiene la estimación total de objetos en esa subregión.

En la ecuación 1 se muestra cómo se modela la función de densidad mediante una transformación lineal.

$$\mathcal{F}(p) = w^T x_p, \tag{1}$$

donde w representa los parámetros del modelo que debemos aprender durante el entrenamiento y x_p el vector de características que representa a cada píxel.

En el esquema de la Figura 3 se describe el procedimiento seguido para el análisis de nuestro sistema. Se divide en dos fases: entrenamiento y testeo del sistema. Se utilizaron, como características visuales, las conocidas como palabras visuales de un modelo Bag-of-Words (BoW) [24], como en [16], aunque se podría trabajar con cualquier tipo de características extraídas por píxel.

Durante la fase de entrenamiento se obtiene el vector de pesos w, que es de vital importancia para obtener el número de vehículos en cada imagen. El resultado final se obtiene en la fase de testeo aplicando la Ecuación 1.

La comprobación del funcionamiento del sistema se ha realizado ejecutando tres experimentos:

- Experimento 1: Diccionario de tamaño 200 y Gaussiana (tamaño= 10*6, σ=10)

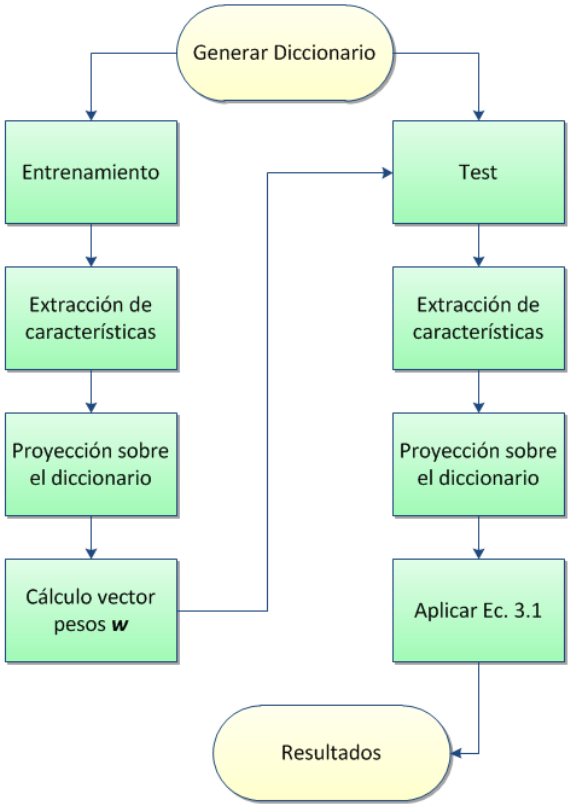


Figura 3: Proceso de entrenamiento del modelo.

- Experimento 2: Diccionario de tamaño 2000 y Gaussiana (tamaño= 10*6, σ=10)
- Experimento 3: Diccionario de tamaño 2000 y Gaussiana (tamaño =15*6, σ=15)

Como conclusión final de nuestro estudio, en la Tabla 1 se muestra una comparación de los resultados cuantitativos obtenidos. En la Figura 4 se realiza una comparación de los resultados cualitativos. A partir de ambas comparaciones se concluye que es mejor trabajar con un diccionario de tamaño 2000 y con una Gaussiana de mayor tamaño.

	Error medio
Experimento 1	17.345227
Experimento 2	13.594236
Experimento 3	12.915479

Tabla 1: Error medio obtenido para los experimentos realizados.

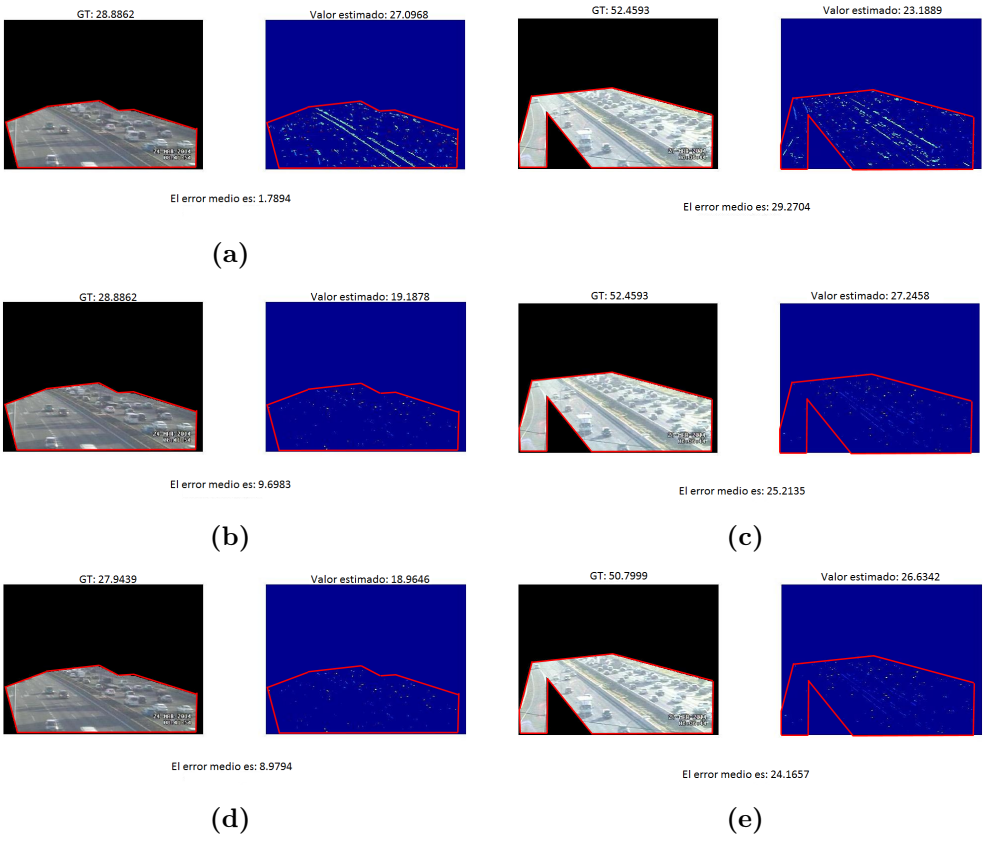


Figura 4: Resultados cualitativos. a) Experimento 1. b) Experimento 2. c) Experimento 3.

Glosario

- **DGT** Dirección General de Tráfico. Es el organismo español encargado principalmente de la vigilancia y control del tráfico entre otras actividades.
- **Tracking** También se le denomina seguimiento. Permite asociar un objeto en un instante preciso con un instante posterior, de ese mismo objeto, permitiendo así la obtención de información acerca de su movimiento y trayectoria.
- **Bounding Box** Término usado para denominar a la región encerrada por un rectángulo en la que se encuentra el objeto detectado.
- **ROI** (Region of Interest) Delimita aquellas zonas que son útiles para el análisis.
- **Ground truth** Este término se refiere a las medidas tomadas de forma manual y se utiliza como comparación para la evaluación de los sistemas automáticos.
- **Variable slack** Es una variable que se utiliza en casos de optimización. Esta variable sustituye una reestricción de desigualdad con una restricción de igualdad.
- **Cutting-plane** Este término se utiliza en optimización matemática. Se emplea como término genérico para referirse a métodos de optimización iterativa que redefinen un posible conjunto o funciones por medio de desigualdades lineales.
- **Validación cruzada** Esta técnica se utiliza para evaluar los resultados obtenidos de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba.
- **Grid-search** Consiste en una búsqueda exhaustiva a través de un subconjunto de hiperparámetros especificados manualmente.

Capítulo 1

Introducción

La apariencia de los objetos de una misma clase en imágenes naturales varía en gran medida debido a diversas causas como, por ejemplo, cambios en la iluminación, en las condiciones de la imagen, escala, etc. Estos factores, así como las oclusiones, hacen que la detección de objetos sea una tarea desafiante, aunque en los últimos años ha habido un gran progreso [9, 10, 11, 23].

En este TFG proponemos seguir avanzando en este campo. En concreto, el proyecto se centra en el estudio de soluciones para realizar un conteo de vehículos de baja resolución. En este contexto se ha desarrollado un sistema para la detección y conteo de vehículos a partir de las imágenes tomadas por las cámaras de tráfico de la DGT. La base de nuestro estudio está cimentada en el modelo propuesto por Lempitsky et al. [16], donde se detalla un modelo capaz de realizar una estimación muy precisa de la densidad de los objetos presentes en la imagen, incluso en situaciones de alta densidad. Aunque en [16] todo el trabajo de centra en contar persona y células, en este proyecto se ha adaptado el sistema para que funcione en el caso particular de estimar la presencia de vehículos en imágenes de baja resolución. Con esta finalidad, se ha construido, específicamente, una base de datos para el desarrollo de este proyecto, lo cual ha permitido el desarrollo y ejecución de cada uno de los experimentos necesarios para la evaluación del funcionamiento del sistema.

La idea es que el sistema desarrollado permita realizar la predicción del nivel de congestión de las carreteras de una forma precisa. No obstante, se aborda en este proyecto el gran reto de efectuar el recuento de vehículos en situaciones de tráfico denso, en las cuales tiene lugar el efecto de solapamiento entre los mismos (véase la Figura 1.1) .

1.1. Motivación y objetivos

Ante la necesidad de poder procesar imágenes de tráfico de baja resolución, como la mostrada en la Figura 1.1, en las que poder estimar de forma precisa el número de vehícu-

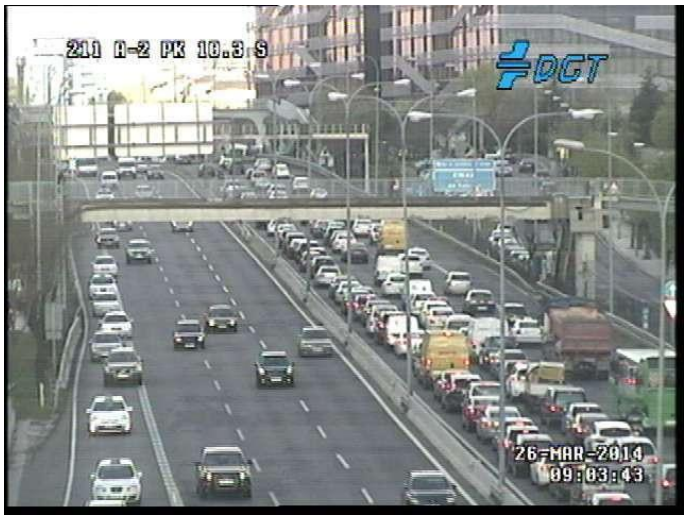


Figura 1.1: Imagen con tráfico denso.

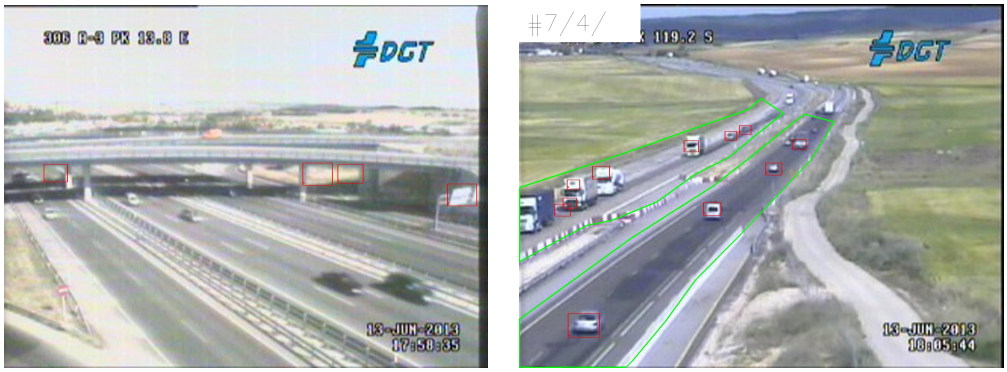
los presentes, ha surgido este proyecto, enmarcado dentro del proyecto de investigación STIMULO (IPT-2012-0808-370000), concedido al grupo de investigación GRAM de la Universidad de Alcalá.

Uno de los objetivos del proyecto STIMULO, es lograr predecir de la forma más exacta posible el nivel de congestión en las vías que dan acceso al puerto para los accesos por carretera a Valencia. Este TFG se enmarca por tanto en el proyecto STIMULO y tiene como objetivo principal la creación de un sistema de validación experimental para sistemas de detección y conteo de vehículos a partir de imágenes tomadas por las cámaras de tráfico. Se va a implementar y evaluar también el modelo descrito en [16] que permite la predicción del número de vehículos estimando la densidad en la imagen. Se propone este modelo ya que para la aplicación concreta no es necesario el conocimiento de la posición exacta de los vehículos, proporcionados por los detectores tradicionales (p. ej. HOG [6]) los cuales han sido descartados ya que no funcionan correctamente con imágenes de baja resolución. En la Figura 1.2(a) se observa como las cajas proporcionadas por [7, 12](bounding box) no detectan los vehículos, lo cual constituye el objetivo principal de este proyecto. En la Figura 1.2(b) vemos que incluso colocando una ROI en la imagen para delimitar la zona en la que el detector tiene que localizar los vehículos, éste sigue equivocándose.

1.2. Campos de aplicación

Dado que el sistema desarrollado permite determinar el nivel de congestión del tráfico en imágenes de videovigilancia, los principales campos de aplicación son:

1. Sistema de estimación de congestión automática para accesos a núcleos urbanos: el



(a) **(b)**

Figura 1.2: a) Detección de vehículos aplicando [7]. b) Detección de vehículos aplicando [7] y una ROI. La ROI definida es la zona que se encuentra delimitada por las líneas verdes.

sistema será capaz de estimar, de forma precisa, la densidad total de tráfico a partir de imágenes de videovigilancia.

2. Sistema de análisis de la congestión para mejorar la eficiencia energética en entornos urbanos: al poder estimar la congestión de tráfico en una zona, donde la información geográfica/cartográfica es conocida (p. ej. en las principales vías de acceso a Madrid), ésta podrá ser utilizada para realizar un análisis de la sobrecarga de las infraestructuras, y poder planificar dónde invertir más recursos para mejorar la eficiencia energética de las ciudades.

3. Mejoras de la seguridad vial: el sistema puede ser utilizado también para detectar aquellas zonas de circulación con una alta congestión, de modo que se puedan tomar las medidas necesarias para reforzar la seguridad de peatones y vehículos en las mismas.

4. Supresión del exceso de contaminación acústica en entornos urbanos: el sistema se puede emplear para detectar aquellos entornos que presentan una mayor congestión de tráfico, pudiéndose así redirigir a los vehículos y evitando un aumento del nivel de ruido el cual puede resultar muy molesto para los ciudadanos.

Capítulo 2

Estado del arte

Existen diversos enfoques que abordan el problema de contar objetos en imágenes de forma no supervisada. Algunos trabajos se basan en la realización de la agrupación en base a la autosemejanza [1] (véase la Figura 2.1(a)) o por similitud de movimiento [21](véase la Figura 2.1(b)).

(a) **(b)**

Figura 2.1: a) Selección de los contornos. Las rodajas de naranja tienen un tamaño, forma y color similar, y su colocación es uniforme estadísticamente. b) Anotación de peatones. Las líneas azules y naranjas muestran gente alejándose y acercándose a la cámara respectivamente con patrones de movimiento similares y que permiten agruparlos y contarlos.

Sin embargo, la precisión de estos métodos no supervisados es limitada y por tanto, en la literatura se consideran otros enfoques basados en aprendizaje supervisado. Éstos se dividen en las siguientes categorías:

- Conteo por detección. Estos modelos basan su cuenta en la utilización de un detector de objetos, que localiza la posición de cada objeto en la imagen. Dadas localizaciones de todos los casos, el conteo es algo trivial. Sin embargo, la detección de objetos está muy lejos de estar resuelta [8], especialmente cuando tienen lugar superposiciones.

- Conteo por regresión. Estos métodos evitan resolver el difícil problema de la detección. En su lugar, una asignación directa de algunas características globales de la imagen (fundamentalmente histogramas formados por varias características) de los objetos es aprendida de modo que se estima por cada imagen una cuenta de los objetos presentes en la misma, siguiendo un esquema de regresión. Algunos ejemplos son [5, 15, 18].

  Kong et al. [15] proponen un modelo que tiene en cuenta la normalización de las características para tratar con la proyección de la perspectiva y las diferentes orientaciones de la cámara. Las características que se entrenan incluyen la orientación del borde y los histogramas resultantes de la detección del borde y la sustracción del fondo. Un mapa de densidad que mide el tamaño relativo de los individuos y una escala global que mide la orientación de la cámara son estimados y usados para la normalización de las características. La relación entre los histogramas de características y el número de peatones en las multitudes es aprendido a partir del etiquetado de los datos de entrenamiento. (Ver la Figura 2.2).

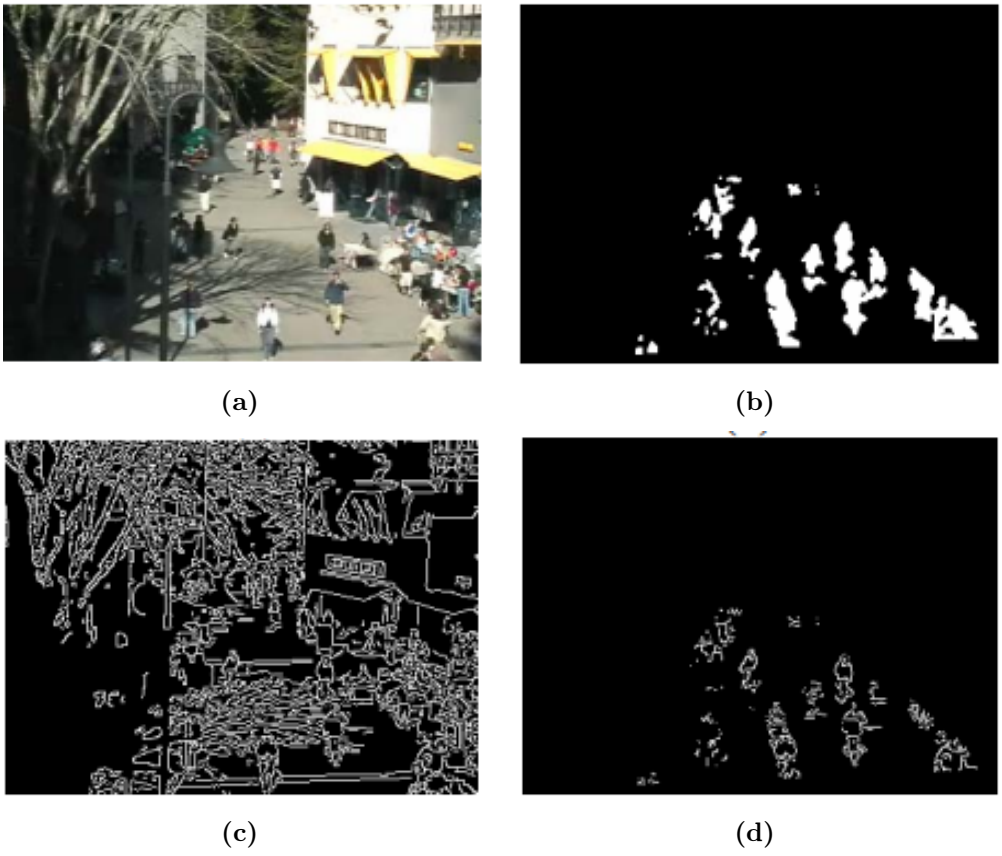


Figura 2.2: a) Imagen original. b) Máscara de la imagen del plano principal. c) Mapa de detección de bordes. d) Mapa de bordes después de una operación 'AND' entre b) y c). (Imagen tomada de [15])

- Procedimientos para contar por segmentación: [4, 22] pueden considerarse como híbridos de los enfoques anteriores (conteo por detección y conteo por regresión). Estas tácticas segmentan los objetos en grupos separados y luego realizan una regresión de la propiedades globales de cada grupo al número total de objetos existentes.

  Ryan et al. [22] han propuesto un algoritmo que utiliza funciones de seguimiento (tracking) y características locales, para contar el número de personas en cada grupo representado por un conjunto de segmentos, de modo que la estimación total será la suma de los tamaños de los grupos. El seguimiento se utiliza para mejorar la estimación total mediante el análisis de la historia de cada grupo, incluyendo la división y fusión de los eventos ocurridos.

Figura 2.3: a) Extracción correcta de los individuos, con ruido adicional (es decir, una pequeña barra cerca del centro). b) Persona (arriba, centro) se fragmenta en dos manchas, una de ellas se fusiona con otra mancha cercana. c) Persona (izquierda) se fragmenta en dos manchas. d) Persona (arriba, centro) se funde con el fondo dejando pocos píxeles de primer plano. Esta persona es apenas visible para el ojo humano. (Imagen tomada de [22])

Este proyecto está basado en una solución planteada en [16] para contar objetos. Lempitsky et al. [16] han propuesto sistemas que nos permitan estimar la densidad de los objetos existentes en las imágenes, y que fueron utilizados con éxito en el problema de conteo de células en imágenes de laboratorio como la mostrada en la Figura 2.4.

En cuanto al conteo de vehículos, son muchos los trabajos que abordan el problema [13, 20, 25]. Todos necesitan que la cámara este fija y que proporcione vídeo de una buena calidad, lo que es inviable con las cámaras de la DGT. Además, en situaciones de alta congestión no han sido exhaustivamente evaluados. Es por ello que este TFG resulta

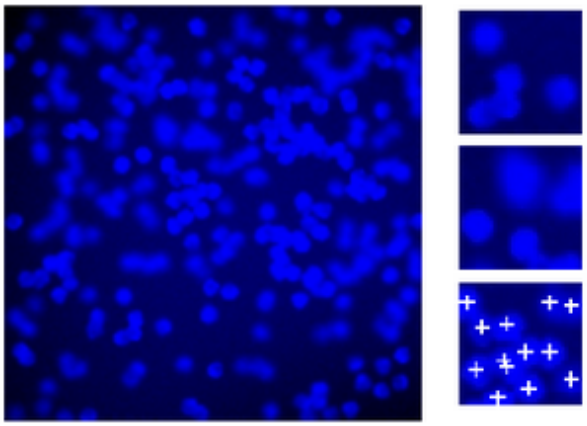


Figura 2.4: Ejemplo de problema de contar objetos, contar células.

interesante y nos va a permitir avanzar en la búsqueda de una solución satisfactoria al problema planteado.

Capítulo 3

Modelo para la estimación del número de objetos en imágenes

3.1. Descripción teórica

El problema de contar objetos consiste en una estimación, lo más precisa posible, del número de objetos en una imagen o fotograma de vídeo. Se ha optado por elegir un enfoque de aprendizaje supervisado para resolver este problema, que se basa en modelos de regresión local. En concreto, en este TFG hemos trabajado con la propuesta descrita en [16] y que describimos en detalle en este capítulo.

En [16], la idea que se describe para determinar el número de vehículos de las imágenes, es sencilla. Dada una imagen I, el objetivo es conseguir el aprendizaje de una función de densidad F como una función real de los píxeles de la imagen. Es decir, se trata de aprender una función de densidad de objetos F, que aplicada a cada píxel, genere la densidad estimada en función de su apariencia.

Conociendo la estimación de la función de densidad F, el número total de objetos en la imagen se estima integrando el resultado de F sobre la imagen. Por otra parte, integrar la densidad sobre una subregión de la imagen $S \subset I$ proporciona una estimación del total de objetos en esa subregión.

En este enfoque, se parte de que cada píxel p en una imagen se representa por un vector de características x_p. A partir de estas características se procede a modelar la función de densidad como una transformación lineal de x_p:

$$\mathcal{F}(p) = w^T x_p, \tag{3.1}$$

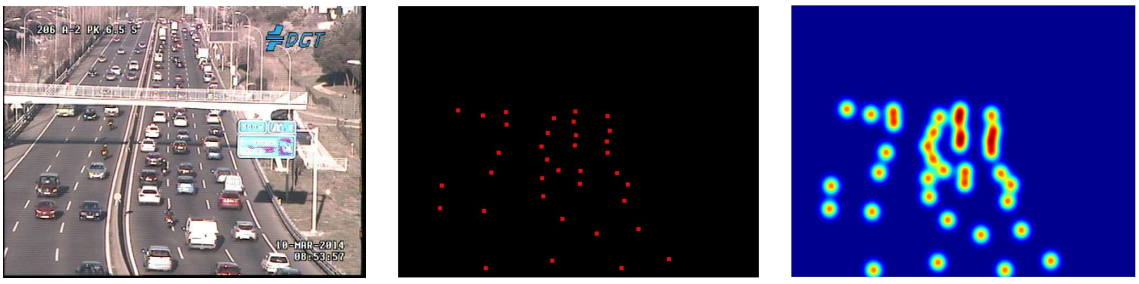
donde w representa los parámetros del modelo que debemos aprender durante el entrenamiento.

Así, a partir de un conjunto de imágenes de entrenamiento se aprende el vector w, de modo que las estimaciones de las funciones de densidad para las imágenes de entrenamiento coincidan con las densidades anotadas.

Las funciones de densidad sobre las rejillas de los píxeles son de gran importancia en el enfoque utilizado para contar vehículos, cuyas integrales en las áreas de la imagen coincidirán con el número de objetos contados. Para una imagen de entrenamiento I_i, se ha definido la función de densidad como una densidad basada en Kernels Gaussianos centrados en los puntos donde hay un objeto anotado. Ésta se define como sigue, basada en los puntos proporcionados:

$$\forall p \in I_i, \quad F_i^0(p) = \sum_{P \in P_i} \mathcal{N}(p; P, \sigma^2 1_{2x2}), \tag{3.2}$$

donde, p indica un píxel, $\mathcal{N}(p; P, \sigma^2 1_{2x2})$ indica un kernel gaussiano 2D normalizado evaluado sobre p, con media en el punto situado por el usuario P, y una matriz de covarianza isotrópica con un σ de valor pequeño (generalmente, unos píxeles). En la Figura 3.1 se representa el proceso de generación de las funciones de densidad en los puntos anotados.Con esta definición, la suma de la densidad del *ground truth* $\sum_{P \in P_i} F_i^0(p)$ sobre toda la imagen no coincidirá exactamente con el recuento de puntos C_i, ya que para los puntos que se encuentran muy cerca del límite de la imagen la masa de su probabilidad Gaussiana se encuentra parcialmente fuera de la imagen. Este comportamiento es natural y deseado en la mayoría de las aplicaciones, debido a que en muchos casos un objeto que está situado parcialmente fuera de los límites de la imagen no debería ser contado como un objeto en su totalidad sino como una fracción de objeto.



(a) **(b)**

Figura 3.1: a) Imagen. b) Anotación de vehículos mediante puntos rojos. c) Representación de las funciones de densidad.

Una vez definido el modelo y la construcción de los mapas de densidad según las anotaciones, podemos proceder a describir como se desarrolla el aprendizaje del modelo definido en la ecuación 3.1.

Dado un conjunto de imágenes de entrenamiento, junto con su *ground truth*, el objetivo principal es aprender la transformación lineal de la representación de las características que se aproxima a la función de densidad de cada píxel:

$$\forall p \in I_i,\ F_i\left(p|w\right) = w^T\, x_p^i, \tag{3.3}$$

donde $w \in R^K$ es el vector de parámetros de la transformación lineal el cual ha sido nuestro objetivo aprender a partir de los datos de entrenamiento, y $F_i\ (\cdot \mid w)$ es la estimación de la función de densidad para un valor determinado de w. Por tanto, se trata de aprender un vector w de manera que minimice la diferencia entre el *ground truth* y las funciones de densidad estimada, y para ello se plantea el siguiente problema de optimización:

$$w = \arg\min_w \left(w^T w + \lambda \sum_{i=1}^{N} \mathcal{D}\left(F_i^0\left(\cdot\right),\ F_i\left(\cdot|w\right)\right)\right). \tag{3.4}$$

En esta ecuación, λ es un hiperparámetro que controla el nivel de regularización. Como se observa, la ecuación busca minimizar la distancia $\mathcal{D}$ entre la densidad estimada y la proporcionada en la anotación.

Para el entrenamiento del modelo, seguimos el proceso de optimización descrito en [16]. En el diagrama de la Figura 3.2 se observa como se realiza el proceso de entrenamiento y test del modelo propuesto.

Una vez que ha sido aprendido el vector w a partir de los datos de entrenamiento, el sistema puede producir una estimación de la densidad de una imagen I mediante una simple ponderación lineal del vector de características calculado en cada píxel, según es sugerido en la Ecuación 3.3. Por tanto, el problema ha quedado reducido a la elección de la correcta función de distancia $\mathcal{D}$ y el cálculo óptimo del vector de pesos w en la ecuación 3.4 para las mismas.

La distancia $\mathcal{D}$ en la ecuación 3.4 mide la desigualdad entre el *ground truth* y la densidad estimada. Ésta tiene un significante impacto en el rendimiento de todo el entorno de aprendizaje. Hay dos elecciones naturales para $\mathcal{D}$:

- Se puede elegir $\mathcal{D}$ para que sea alguna función de métrica L_p, por ejemplo la métrica L_1 (suma de la diferencia absoluta de píxeles) o el cuadrado de la métrica L_2 (suma de cuadrados de la diferencia de píxeles). Estas opciones convierten la Ecuación 3.4 en un problema de regresión estándar, donde cada píxel en cada imagen de entrenamiento proporciona una muestra en el conjunto de entrenamiento.

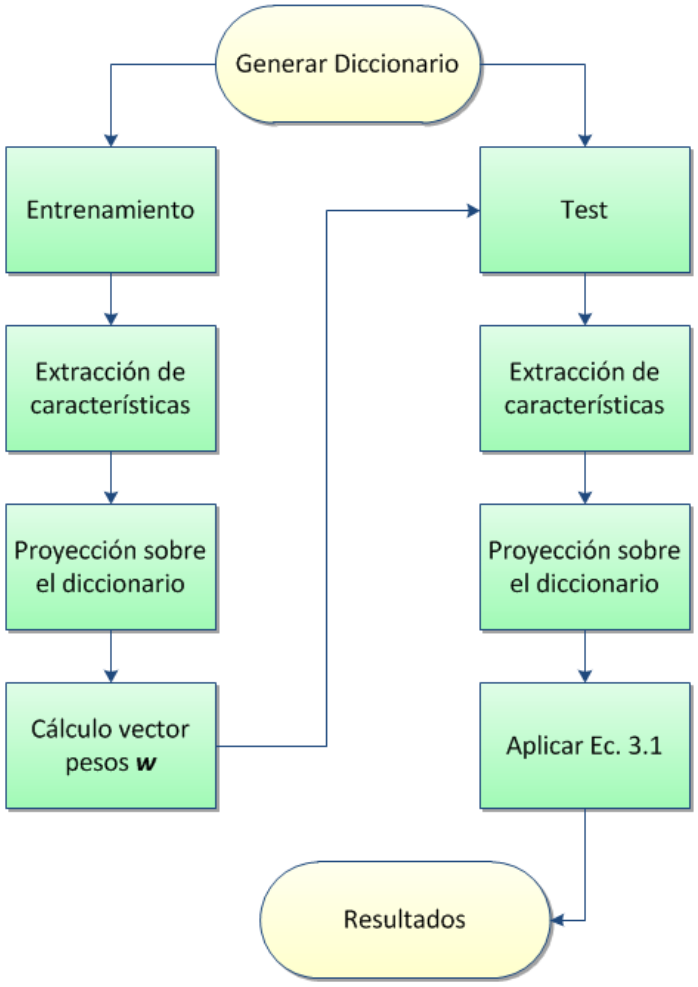


Figura 3.2: Proceso de entrenamiento del modelo.

- Como la suma total es en definitiva lo importante, se puede elegir que $\mathcal{D}$ sea una diferencia absoluta, o de cuadrados, entre las sumas globales sobre la totalidad de las imágenes para los dos argumentos, por ejemplo:

$$\mathcal{D}(F_1(\cdot), F_2(\cdot)) = \left| \sum_{p \in I} F_1(p) - \sum_{p \in I} F_2(p) \right|, \tag{3.5}$$

  donde ahora cada muestra del entrenamiento corresponde a la imagen de entrenamiento entera. Por lo tanto, aunque esta elección coincide con nuestro objetivo final de aprender a contar correctamente, se requiere de muchas imágenes anotados para el entrenamiento, ya que la información espacial, en la anotación, se descarta.

Teniendo en cuenta las significativas desventajas que muestran las anteriores distan-

cias, se ha optado por elegir una alternativa diferente, la distancia MESA, propuesta en [16]. Dada una imagen I, la distancia MESA $\mathcal{D}_{MESA}$ entre dos funciones $F_1(p)$ y $F_2(p)$ sobre la rejilla de píxeles, es definida como la diferencia absoluta más grande entre las sumas de $F_1(p)$ y $F_2(p)$ sobre todos los subconjuntos en I:

$$\mathcal{D}_{MESA}(F_1, F_2) = \max_{B \in \mathbf{B}} \left| \sum_{p \in B} F_1(p) - \sum_{p \in B} F_2(p) \right|. \tag{3.6}$$

En este caso, $\mathbf{B}$ es el conjunto de todos los subconjuntos de cajas de I.

La distancia MESA puede ser considerada como una distancia L_∞ entre vectores de las sumas de los subconjuntos. Esta distancia posee dos propiedades muy deseables:

1. **Robustez**. La distancia MESA es robusta a las perturbaciones locales aditivas de sus argumentos, tales como el ruido independiente o la señal de alta frecuencia, siempre y cuando las integrales de estas perturbaciones sobre la región más grande sean próximas a cero. Por lo tanto, no importa mucho cómo se defina exactamente la densidad *ground truth* a nivel local, siempre y cuando las integrales de la densidad *ground truth* sobre las regiones más grandes calculen el total correctamente. Podemos entonces definir la densidad *ground truth* para una anotación de puntos como la suma de gaussianas normalizadas centradas en los puntos.

2. **Computabilidad**. La distancia MESA se puede calcular con exactitud a través de un algoritmo combinatorio eficiente (máximo sub-array [3]).

3.1.1. Características visuales

El modelo presentado permite trabajar con cualquier tipo de características extraídas por píxel. En este trabajo, como en [16], proponemos un modelo basado en diccionarios de palabras visuales. Podemos comparar nuestro sistema con el aprendizaje de un idioma, donde también existen diccionarios visuales de consulta que nos permiten aprender el significado de cada una de las palabras o traducirlas a nuestro idioma materno. En nuestro caso se proyectan las características extraídas por píxel sobre el diccionario visual para poder determinar de qué se trata, es decir, en nuestro caso si se corresponde o no con un vehículo. Gracias a esto aprendemos a identificar los vehículos lo que finalmente permite la correcta detección y conteo de los mismos.

A grades rasgos, el sistema de extracción de características funciona del siguiente modo. De un conjunto de imágenes de entrenamiento procedemos a extraer descriptores visuales por cada píxel. Estos descriptores son vectores, de una alta dimensionalidad,

que describen el contenido visual de forma robusta (p.ej. SIFT [26, 28], SURF [2, 14]). Una vez han sido extraídos, procedemos a organizarlos en grupos, utilizando técnicas de clusterings, como K-means. Cada grupo identificado por la técnica de clustering es considerado como una palabra visual. De este modo, cada vector queda asociado al grupo o cluster en el que el proceso de clustering lo insertó. Es decir, que al final, cada píxel de la imagen es asignado a un cluster del vocabulario, lo que se puede entender como un código o número. El diagrama de la Figura 3.3 representa de forma gráfica todo este proceso.

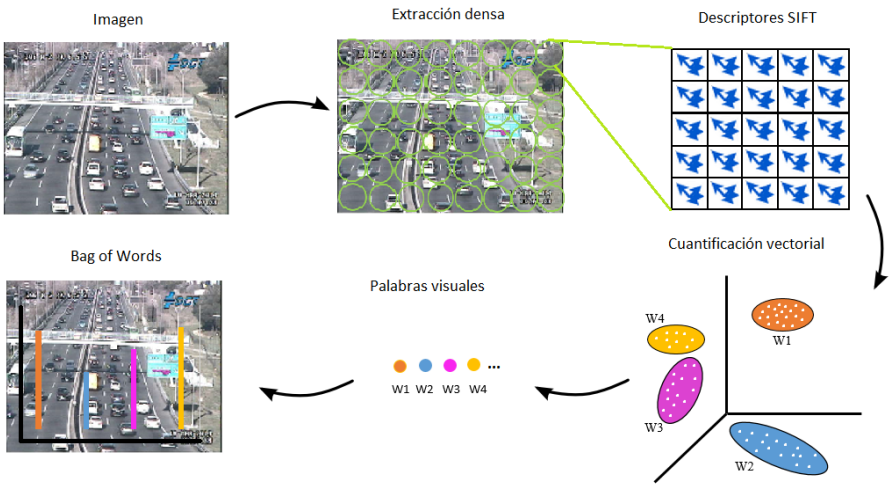


Figura 3.3: Proceso de obtención de la representación Bag-of-Words.

Así, el primer paso que proponemos es transformar cada una de la imágenes en color a escala de grises, realizándose posteriormente una extracción de las características. En el proceso de generación del diccionario se van a utilizar solo las imágenes de entrenamiento. Las características se han obtenido mediante vectores SIFT (se explicarán en la sección 3.2), concretamente a partir de una extracción densa de estos vectores en toda la imagen.

Posteriormente, se ha establecido un número máximo de características a extraer de todas las imágenes. Si el número total de las mismas supera al límite fijado, se seleccionan de forma aleatoria hasta llegar al máximo elegido. Por último, para la creación del diccionario se ha hecho uso del algoritmo de clustering K-means. Se trata de un algoritmo de agrupamiento cuyo objetivo es devolver una serie de puntos que por su posición central en los grupos representan al resto de puntos y que son denominados centroides. El algoritmo funciona del siguiente modo: conocido el valor de K y los vectores a agrupar, se generan k centroides aleatorios. A continuación se calcula la distancia de los vectores hasta los centroides y se agrupan los objetos según la mínima distancia a los mismos. Los centroides

se recalculan según los nuevos grupos y se vuelve a iterar hasta que el proceso converja. El diagrama de la Figura 3.4 ejemplifica el funcionamiento del algoritmo K-means. Finalmente, en la Figura 3.5 mostramos un ejemplo de agrupación de datos bidimensionales, utilizando K-means, y fijando K=5. La matriz de centroides que devuelve el algoritmo *k-means* es nuestro diccionario.

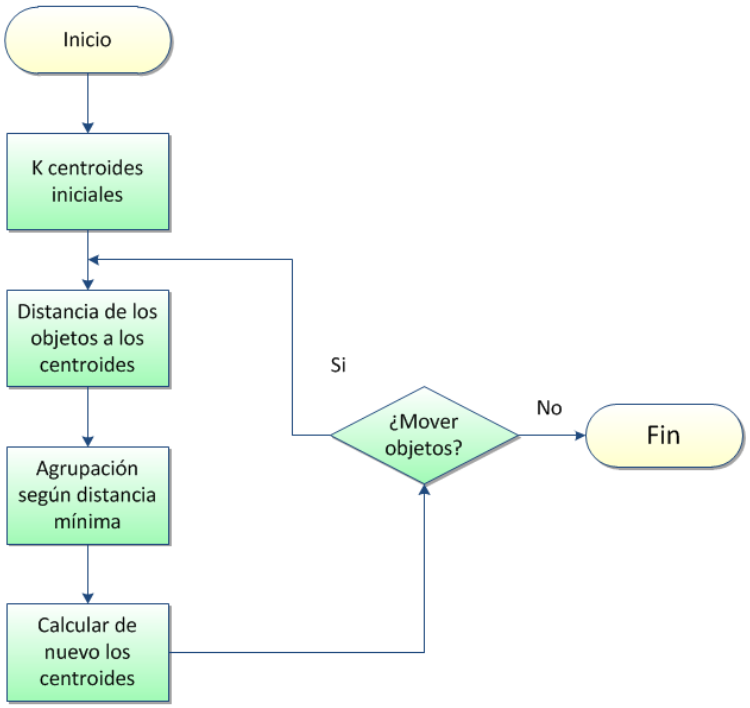


Figura 3.4: Funcionamiento del algoritmo kmeans.

3.2. Implementación técnica del modelo de software

Para la implementación del sistema propuesto de ha utilizado la eficiente librería de visión por computador VlFeat [27] y el código original que acompaña al paper [16] liberado por Lempitsky et al. Se trata de una librería libre que contiene algoritmos útiles para técnicas de visión artificial. Concretamente se ha hecho uso de dos algoritmos. El primero para la extracción de características mediante el uso de descriptores SIFT de forma densa (vl_dsift). El segundo algoritmo que se ha utilizado ha permitido la proyección de las características extraídas a partir de las imágenes sobre el diccionario creado.

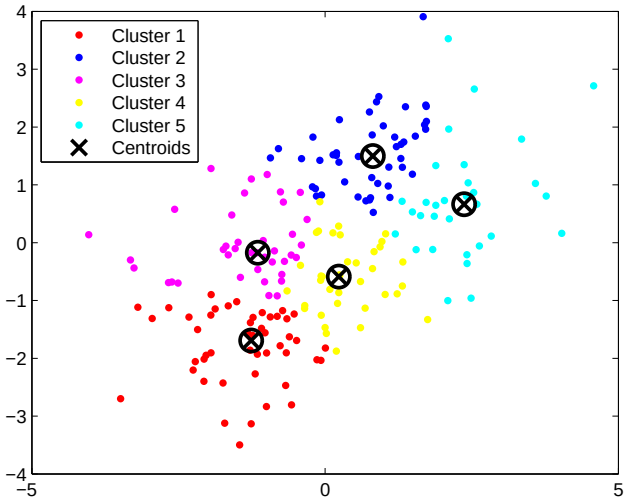


Figura 3.5: Ejemplo de agrupamiento de datos mediante k-means. Los datos se particionan en 5 clusters diferentes (k=5) con sus respectivos centroides (uno por grupo).

3.2.1. Extracción de descriptores SIFT

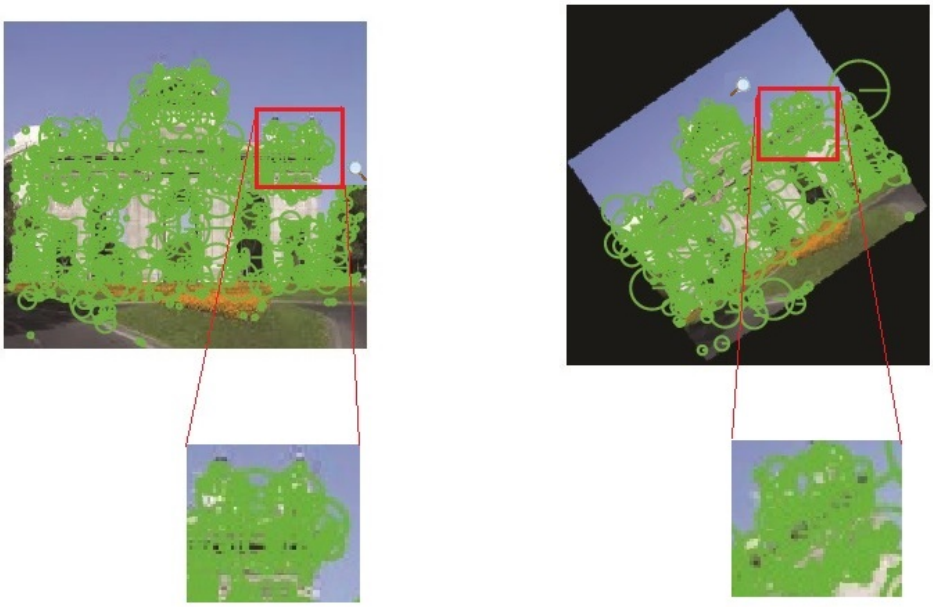
Hasta el momento se ha hablado de descriptores SIFT [17] pero no hemos explicado qué son.

Estos descriptores surgen ante la necesidad de describir de manera robusta el contenido de la información visual. Los descriptores SIFT fueron publicados por primera vez por David Lowe [17].

Se define un SIFT frame como una orientación dominante que se especifica por cuatro parámetros: el centro t_x, t_y, la escala $\mathbf{s}$, y la rotación θ (en radianes) obteniendo como resultado un vector formado por cuatro parámetros (s, θ,t_x, t_y,). En la Figura 3.6(a) a la izquierda se muestra la imagen de la Puerta de Alcalá y a la derecha ésta misma, pero ahora presenta un escalado y una rotación de 30 grados. En la Figura 3.6(b) se aprecian las imágenes anteriores con sus respectivos SIFT frames detectados. Si se examinan estas dos imágenes podemos comprobar que las detecciones suceden en las mismas regiones de la escena, aunque los círculos se han desplazado de su posición original en la imagen y el radio ha sido modificado. Esto demuestra el alto grado de invarianza que presentan este tipo de descriptores ante rotaciones y cambios de escala.

Finalmente en la Figura 3.7 puede observarse el resultado de realizar una extracción densa de descriptores SIFT en las imágenes de tráfico que utilizamos en este TFG.

(a)



(b)

Figura 3.6: a) Imagen original e imagen escalada y rotada 30 grados. b) Visualización de los vectores de características SIFT (frames).

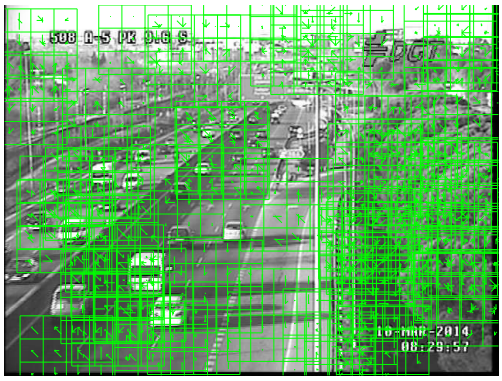


Figura 3.7: Representación aleatoria de los gradientes obtenidos mediante una extracción densa.

Capítulo 4

Base de Datos

Para la realización de la validación experimental del modelo descrito en el capítulo 3, hemos necesitado recopilar una base de datos con imágenes de tráfico, que han tenido que ser debidamente anotadas, manualmente. En este capítulo describimos la base de datos GRAM-LOQUATRAF.

4.1. Descripción de la base de datos

Para generar la base de datos GRAM-LOQUATRAF se han observado varias carreteras de la provincia Madrid, buscando aquéllas en las que la densidad de tráfico fuera alta. También se han analizado a distintas horas del día, eligiendo finalmente las primeras horas de la mañana (desde las 7:30 horas hasta las 10:00 horas), que es cuando las carreteras poseen una mayor concentración de tráfico. Las cámaras seleccionadas han sido las siguientes:

- Carretera A-2 (Km 6,5; Km 10,3 y Km 17,4).
- Carretera A-6 (Km 3,5; Km 4,1 y Km 14,3).
- Carretera M-40 (Km 12,9).
- Carretera A-5 (Km 7,9; Km 9,6 y Km 12,6).
- Carretera A-42 (Km 12,3).

En la Figura 4.1 se muestran dónde están ubicadas geográficamente las cámaras que se han seleccionado para la generación de la base de datos.

Una vez seleccionadas las cámaras, se han capturado imágenes durante tres semanas consecutivas (las imágenes de la primera semana se utilizarán para el entrenamiento del

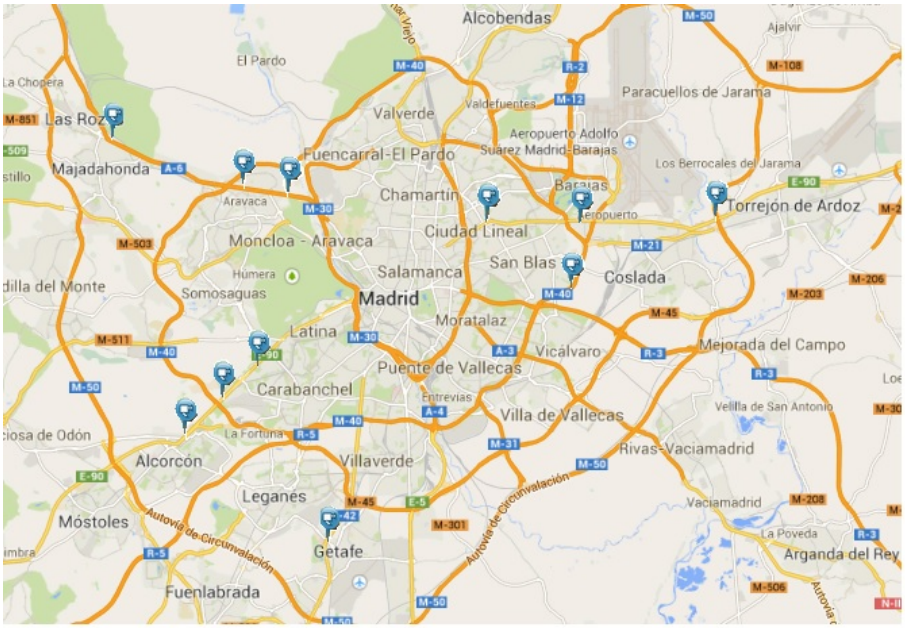


Figura 4.1: Geoposicionamiento de las cámaras seleccionadas.

sistema, las de la segunda para la validación del sistema y las de la tercera para el testeo del sistema). Para ello, se desarrolló una herramienta en C++ que se distribuye junto con el código del TFG.

Una vez las imágenes fueron capturadas, la generación de la base de datos se ha realizado siguiendo el procedimiento mostrado en la Figura 4.2. La Figura 4.3 muestra un ejemplo de anotación realizada.

Semana	1	2	3
Finalidad de las imágenes	Entrenamiento	Validación	Testeo
Imágenes	403	420	421
Imágenes anotadas	403	420	421
Total objetos anotados	12383	16359	16086
Número de objetos medios por imagen	31	39	38

Tabla 4.1: Imágenes que forman la base de datos y objetos anotados.

La base de datos obtenida está formada por un total de 1244 imágenes. Como decíamos éstas corresponden a tres semanas distintas. El grupo de imágenes de la primera semana es denominado, *training*, contiene 403 imágenes que se han utilizado para el entrenamiento del sistema. El segundo grupo (segunda semana), ha sido denominado *validation*, y está

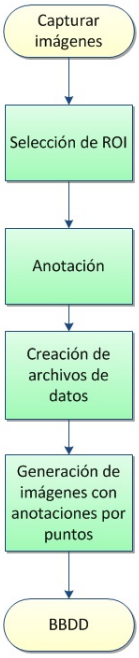


Figura 4.2: Diagrama de flujo del proceso de elaboración la base de datos.

formado por 420 imágenes que son utilizadas para ajustar los parámetros del sistema, sin necesidad de utilizar las imágenes reservadas para test que podrían falsear los resultados. Por último, el tercer grupo (tercera semana), *test*, contiene 421 imágenes, que se utilizan para testear soluciones finales y dar los resultados de cuenta de vehículos (véase la Tabla 4.1).

(a) **(b)**

Figura 4.3: Imágenes que componen la base de datos a) Imagen original. b) Imagen con ROI y anotación.

Como se ha indicado previamente, la base de datos elaborada está formada por un total de 1244 imágenes proporcionadas por un conjunto de cámaras públicas de la DGT, que supervisan diferentes carreteras de Madrid. Estas imágenes son de baja calidad, siendo su

resolución de 640 x 480 píxeles. Al tratarse de cámaras de exterior, hay diversos factores que afectan a la calidad de las imágenes que producirán: acumulación de polvo; el comportamiento de algunos animales (como es el caso de las arañas que crean sus telarañas en las cámaras), cambios extremos de iluminación (esto ocurre tanto en imágenes de día como de noche) e incluso factores meteorológicos como la lluvia. Se ha llevado a cabo un filtrado de las imágenes con el objetivo de eliminar aquéllas en las que la detección y conteo de los vehículos sea de una dificultad extremadamente alta, ya que en algunas es prácticamente imposible la detección de los mismos. Ejemplos de estos factores se pueden observar en la Figura 4.4. Además de lo ya expuesto hay, que añadir que las cámaras de la DGT no son fijas, es decir, rotan apuntando a diferentes zonas de la carretera. Los operarios las mueven para prestar especial atención al pavimento o a los accidentes. Estas variaciones llevadas a cabo por los trabajadores impiden la utilización de una ROI estática, y cada imagen deba llevar una ROI propia asociada. Éstas son las causas principales que hacen que las imágenes almacenadas en la base de datos constituyan un gran desafío a la hora de realizar el conteo de los vehículos.

Finalmente se ha desarrollado un proceso de verificación que permite asegurar que tanto la anotación como la ROI elegida para cada imagen de la base de datos son las correctas. Para ello, se ha superpuesto sobre la imagen original la ROI y los puntos de la anotación agrandados. (Véase Figura 4.5)

4.2. Anotación

La forma de anotación elegida consiste en especificar la posición de cada objeto en la imagen mediante un punto. Señalar utilizando puntos es la forma natural de contar para las personas, al menos cuando el número de objetos es elevado. Por tanto, se puede decir que a una persona no le resultaría más complicado proporcionar las anotaciones correspondientes a las imágenes de entrenamiento que proporcionar el resultado final. Además, cabe destacar que en general la anotación mediante puntos es menos laboriosa que una anotación realizada con bounding boxes.

En la Figura 4.6(a) se puede observar una anotación realizada con puntos (un punto por cada vehículo). Esta anotación resulta mucho más rápida que la mostrada en la Figura 4.6(b) puesto que solo hay que poner un punto en el centro del vehículo y actualmente gracias a la evolución de las nuevas tecnologías, la mayoría de dispositivos poseen pantallas táctiles, podría realizarse la anotación de una forma muy rápida, puesto que con ir tocando sobre los diferentes vehículos éstos ya quedarían anotados. En la utilización de bounding boxes se requiere ser más preciso ya que hay que realizar un recuadro lo más próximo posible al vehículo, resultando una tarea dura y lenta.

Para agilizar la anotación se ha creado una herramienta de Matlab que permite de forma rápida y sencilla seleccionar los vehículos. La función "ginput" permite distinguir los vehículos en cada una de las imágenes devolviendo las coordenadas x e y correspondientes a cada uno de ellos. Posteriormente estas coordenadas se guardan en un archivo de texto, lo que permite acceder a ellas en caso de que sea necesario. Por tanto, se concluye que cada imagen de entrenamiento I_i ha sido anotada con un conjunto de puntos 2D P_i=P_1,...,$P_{C(i)}$, siendo C(i) el número total de objetos anotados por el usuario.

En un paso previo a la anotación, se ha delimitado una región de interés (ROI) para cada una de las imágenes, ya que las cámaras de la DGT se mueven constantemente cambiando el área enfocada y por tanto no se puede fijar una ROI común para todas las imágenes. Esta ROI se ha utilizado en la sección de test del sistema en la cual se procede a calcular el valor estimado a partir de las densidades estimadas.

Para generar la ROI se ha empleado la función de MATLAB "roipoly" (véase Algoritmo 1). Esta función permite especificar una región de interés, ROI, dentro de una imagen. "roipoly" devuelve una imagen binaria del mismo tamaño que la imagen original donde los píxeles que se encuentran dentro de la región deseada tienen valor 1 y los que están fuera valor 0.

Algoritmo 1 Algoritmo que permite la creación de una ROI de manera rápida y sencilla.

```
BW = roipoly(imagen);
```

Por tanto, cada imagen tiene asociado un archivo de texto en el que se guardan las anotaciones de los vehículos, una imagen en formato "*dots.png" donde los vehículos vendrán representados por puntos rojos y una archivo "*mask.mat" que contiene la ROI.

4.3. Distribución de la Base de Datos

Las imágenes de la base de datos se encuentran divididas en tres carpetas: training, validation y test. Como sus nombres indican, las imágenes de la carpeta training se utilizan para el entrenamiento del sistema, las de la carpeta validation para el ajuste y validación de parámetros, y por último las correspondientes a la carpeta test, para el testeo del sistema. Cada una de estas carpetas contiene las imágenes capturadas, en formato JPG, y asociadas a ellas se encuentran las anotaciones, tanto en formato texto, es decir, las coordenadas x e y de donde se sitúan los vehículos, como la representación mediante puntos rojos de dichas anotaciones. Además, dado que ha sido necesario generar una ROI para cada imagen, también existe un archivo *.mat, que contiene la información referente

a la ROI. En la siguiente figura se esquematiza cómo está organizada la base de datos (ver Figura 4.7).

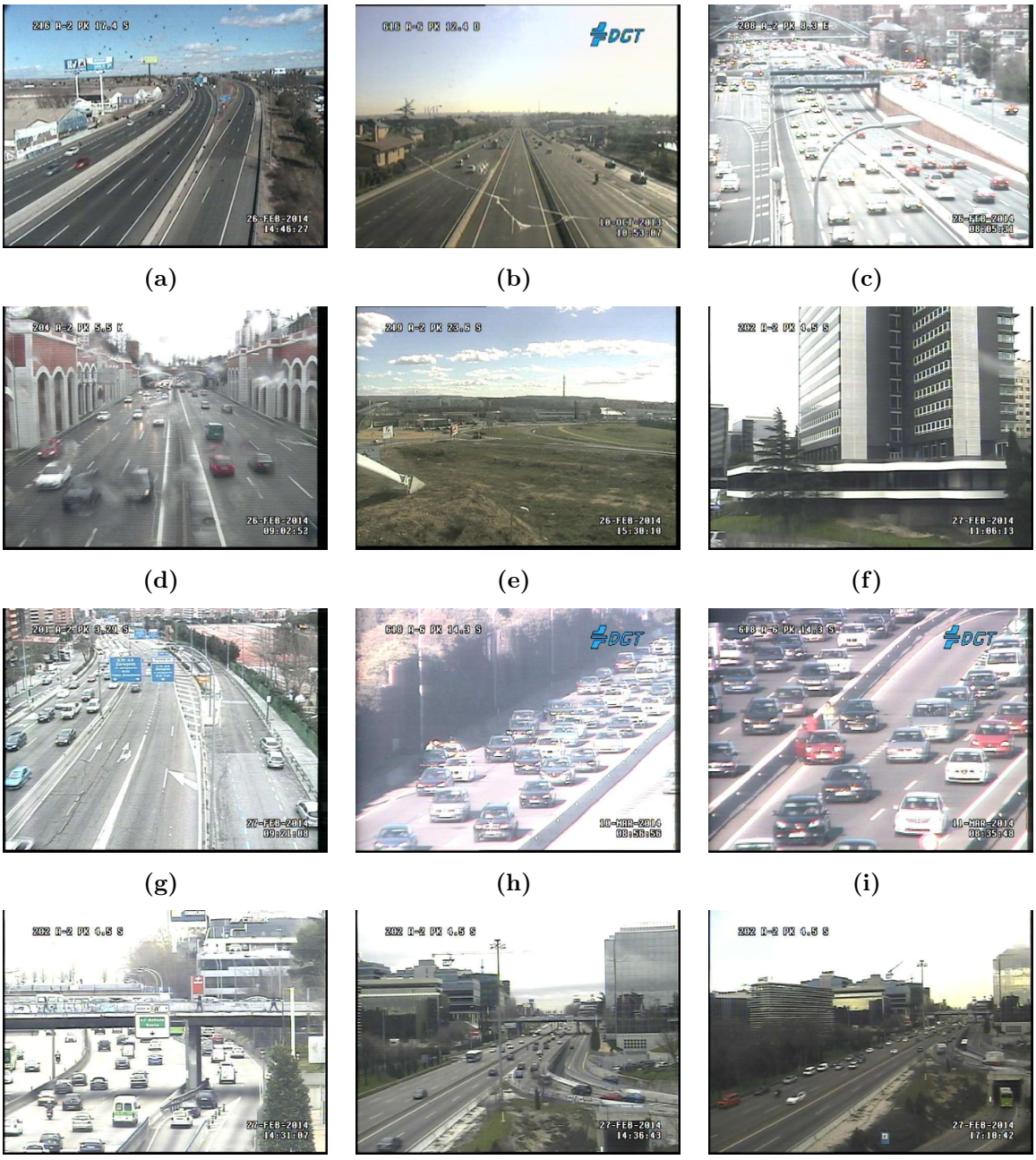


Figura 4.4: Desde la imagen a) hasta la d) se pueden observar los factores externos que deterioran la calidad de la imagen como el polvo, telarañas, la luz o la lluvia. Las imágenes e) - i) muestran los cambios que tienen las cámaras, desde enfocar a edificios hasta accidentes. Las tres últimas imágenes $\Big(j), h), i)\Big)$ muestran como los operarios modifican el zoom de las cámaras de una misma carretera, lo que impide utilizar una ROI estática.

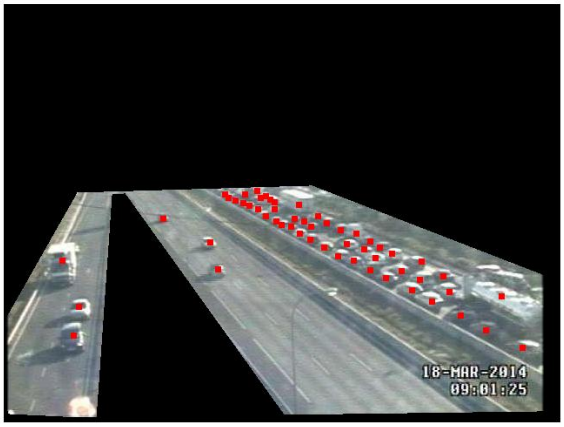
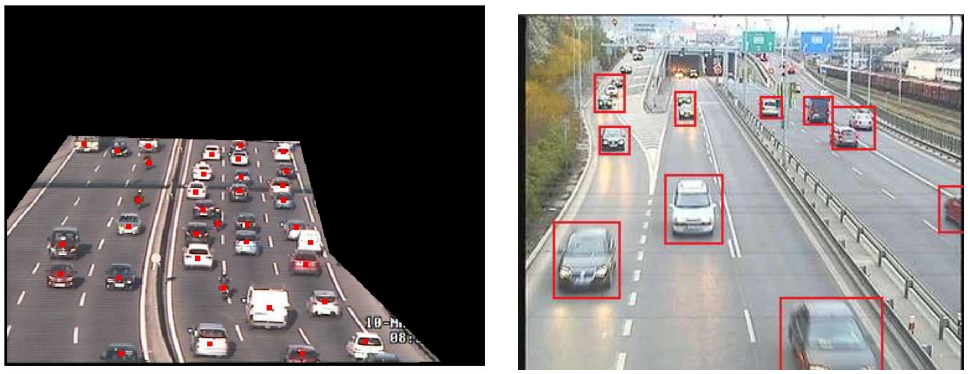


Figura 4.5: Imagen original con la ROI y la anotación sobrepuesta para comprobar que está realizado correctamente.



(a) **(b)**

Figura 4.6: a) Anotación de vehículos mediante puntos. b) Anotación de vehículos con bounding boxes [29].

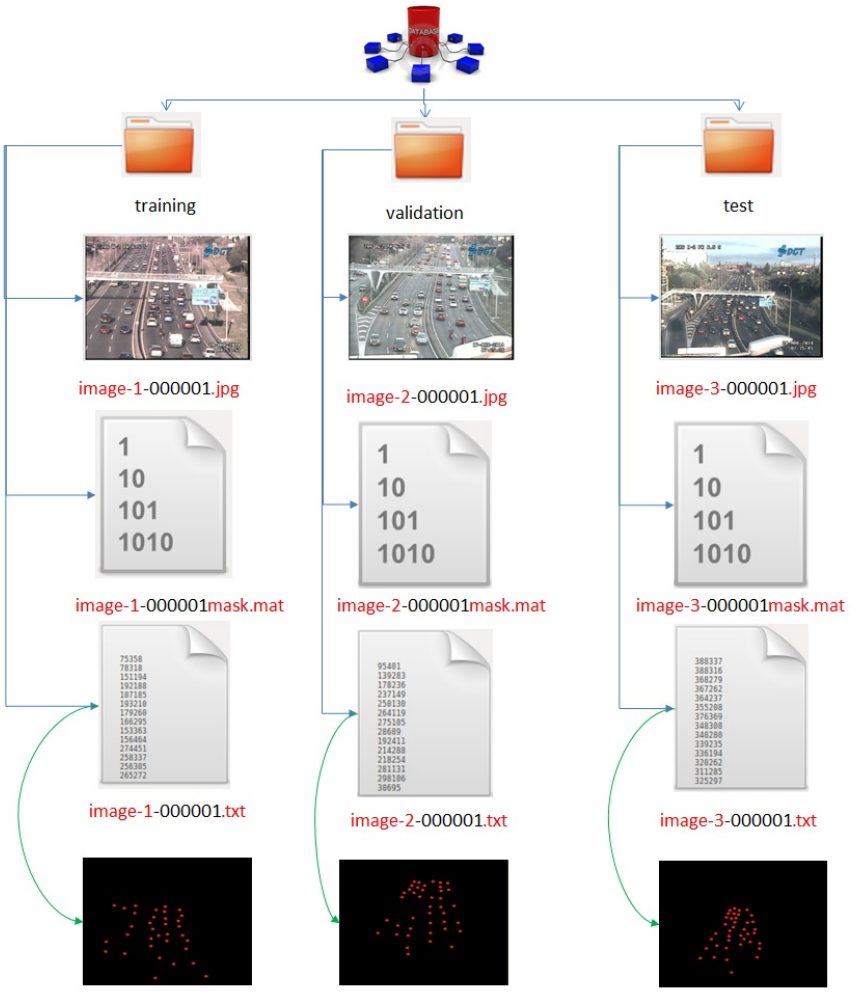


Figura 4.7: Esquema representativo de la organización de la base de datos.

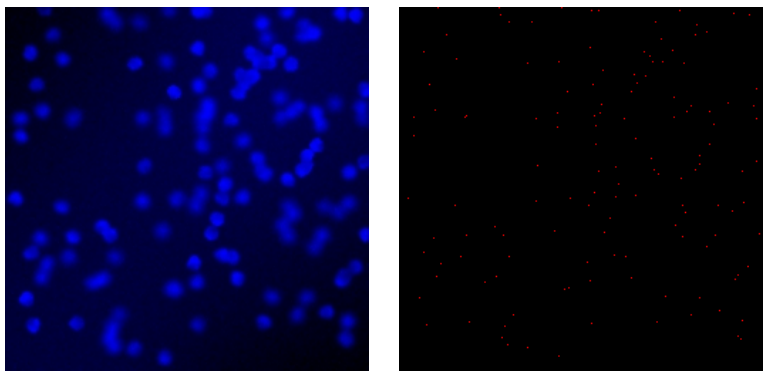
Capítulo 5

Resultados

Para comprobar el correcto funcionamiento del sistema se han realizado varios experimentos. En este capítulo vamos a proceder a presentar las aproximaciones realizadas, así como los resultados, cuantitativos y cualitativos, que se han obtenido. Gracias a ellos se ha tenido una idea de cómo se comporta nuestro sistema frente a diversas situaciones.

5.1. Conteo de células

El primer experimento realizado, se basa en la base de datos de las células proporcionadas en [16] (véase la Figura 5.1). Se ha creado un diccionario con una dimensión de 256, a partir de descriptores SIFT, y a continuación se ha ejecutado el sistema para comprobar los resultados reportados en [16]. Con este experimento buscamos verificar nuestra implementación del sistema, asegurándonos que obtenemos resultados similares a los reportados en [16].



(a) **(b)**

Figura 5.1: Imagen de las células (a) y su anotación (b).

	Regularización L1	Regularización Tikhonov
1. Density Learning Lempitsky [16]	3.6189	3.8591
2. Density Learning	3.276699	9.209553
3. Density Learning	4.2352	13.9550

Tabla 5.1: **1.**Resultados originales de Lempitsky. **2.** Resultados obtenidos generando el diccionario. **3.** Resultados obtenidos aplicando las Gaussianas seleccionadas para los vehículos.

En la Tabla 5.1 se comparan los resultados reportados en [16] con los resultados obtenidos en este TFG. Para calcular el error medio se procede del siguiente modo. Primeramente, se calcula la diferencia entre el total de las anotaciones y el total de las estimaciones para cada imagen. A continuación, se calcula el valor medio del valor absoluto de las diferencias, obteniéndose finalmente el error medio medido. Su unidad de medida variará en función de si contamos células, vehículos u otros objetos.

Se han realizado dos experimentos con la base de datos de las células, el primero generando nuestros propios diccionarios del mismo tamaño que el proporcionado por [16] y el segundo modificando los parámetros de las Gaussianas (tamaño y σ). Este segundo experimento se ha realizado ya que se desea validar cómo afecta el tamaño o el σ de las Gaussianas, dado que para los vehículos va a ser necesario variarlos y se ha decidido probar primeramente con las células.

Si nos fijamos en los resultados de la Tabla 5.1, concretamente en la regularización L1 se corrobora que los diccionarios están creados correctamente, pues la diferencia entre errores es mínima (obsérvese los resultados entre la primera y segunda fila de la Tabla 5.1. También se observa que los parámetros que se han aplicado para los vehículos podrían valer para las células. Sin embargo, para la regularización L2 (Tikhonov) todos los resultados empeoran bastante, esto se debe a que en la estimación de la densidad se introduce más ruido (ver Figura 5.2(c) y Figura 5.2(f))

En la Figura 5.2 se realiza una comparativa entre las Gaussianas originales y las que se ha decidido modificar sus parámetros para su posterior aplicación en los vehículos. Si nos fijamos en la Figura 5.2(a) vemos cómo la Gaussiana es más redonda y pequeña que la mostrada en la Figura 5.2(d), esta característica es importante para el caso de las células, ya que son pequeñas y circulares por lo que las Gaussianas las cubren perfectamente. En el caso de los vehículos nos interesa que el tamaño de las Gaussianas sea un poco más grandes para asegurarnos que cubre completamente a estos. Al comparar la Figura 5.2(b) y 5.2(e) vemos que las densidades estimadas para la regularización L1 son similares independientemente del tamaño de la Gaussiana, mientras que en el caso de la regularización de Tikhonov si existe una gran diferencia en las densidades estimadas (ver

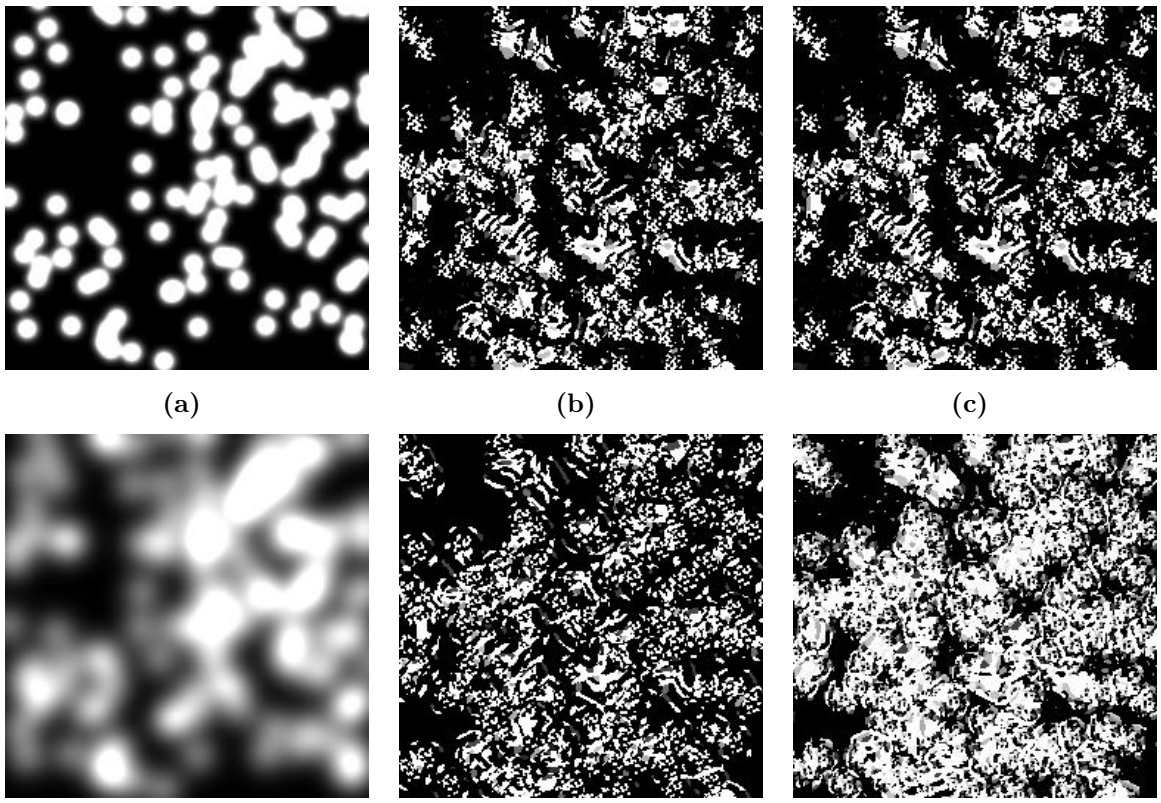


Figura 5.2: a) y d) representan las Gaussianas que se han obtenido, siendo a) las originales y d) las modificadas para los vehículos. El resto de figuras representan las estimaciones de densidad para las dos regularizaciones (b) y e) regularización L1; c) y f) regularización de Tikhonov), siendo las imágenes de arriba las obtenidas mediante el código original y las de abajo para nuestro código.

Figura 5.2(c) y Figura 5.2(f)), introduciendo mayor ruido la Figura 5.2(f). En este caso se observa cómo una Gaussiana de un tamaño superior afecta bastante a las densidades estimadas mediante la regularización de Tikhonov, introduciendo gran cantidad de ruido.

5.2. Cuenta de vehículos

Con el fin de obtener los resultados finales se ha hecho uso de la base de datos GRAM-LOQUATRAF, descrita en el capítulo 4.

Para la correcta evaluación de nuestra solución en la base de datos creada, GRAM-LOQUATRAF, es necesario un ajuste del parámetro de regularización λ de la ecuación 3.4. Para ello seguimos un proceso de validación, utilizando las imágenes de la segunda semana (conjunto de validación) para realizar los ajustes necesarios. Se recomienda una

búsqueda exhaustiva del parámetro λ. Varios valores de λ son probados y con el que se obtenga la mayor precisión en la validación es con el que se realiza el test final.

Como decíamos, para el caso concreto del entrenamiento del sistema de conteo de vehículos, se han utilizado las imágenes del grupo *training* para el entrenamiento, y las del grupo *validation* para la parte de test. El test se ha realizado para varios valores de λ ($\lambda = [10^{-4}, 10^{6}]$). Posteriormente, se han utilizado las imágenes de los grupos *training* y *validation* para realizar el entrenamiento y las del grupo *test* para realizar el test, en este caso se ha utilizado el parámetro λ para el que se había obtenido mejores resultados. Este procedimiento se explica mediante el siguiente diagrama de flujo (véase Figura 5.3).

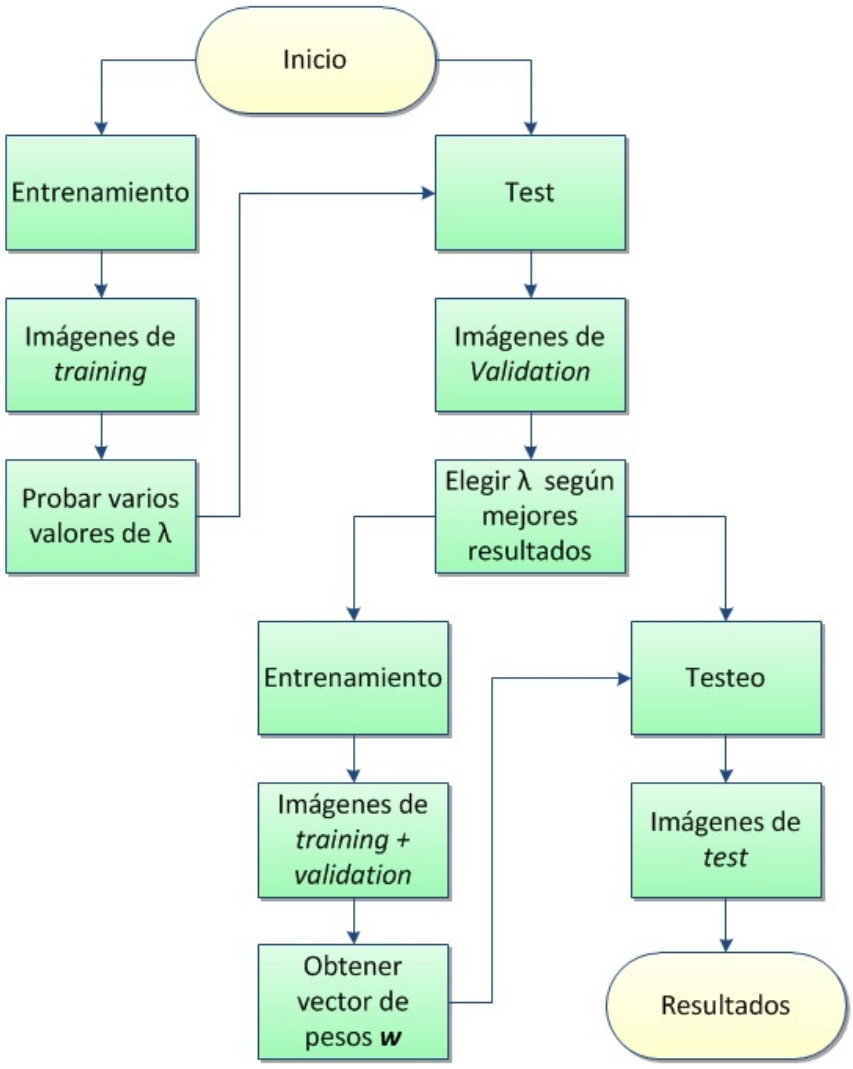


Figura 5.3: Pasos para la ejecución del sistema.

Como hemos descrito en el capítulo 3, para entrenar el sistema, es necesario partir de unas densidades generadas a partir de la anotación por puntos. Por eso, uno de los parámetros de nuestro sistema, es precisamente el tamaño de las Gaussianas que se centran en los puntos anotados. Diferentes σ darán lugar a diferentes resultados, por lo que este parámetro también debe ajustarse. Lo idóneo es que cada Gaussiana cubra a los vehículos para que los resultados obtenidos sean los mejores posibles (Figura 5.4).

(a) **(b)**

Figura 5.4: a) Muestra la imagen original de los vehículos. b) Se observan las respectivas Gaussianas para la anotación de los Vehículos.

Como bien se ha dicho anteriormente, ha sido necesario ajustar dos parámetros de las Gaussianas (tamaño y σ). Se debe ser cauteloso a la hora de modificarlos, ya que están estrechamente relacionados. La Figura 5.5(a) muestra la representación de una Gaussiana. Su tamaño viene dado por el eje x y la varianza (σ) viene definida como la parte superior de la campana. En la Figura 5.5(d) se representa la Gaussiana del modo que se va a presentar en este TFG; el círculo representa σ y los cuadrados dos tamaños de la Gaussiana. Se han realizado dos pruebas con las Gaussianas:

- En la primera se ha tomado un tamaño de la Gaussiana bastante mayor que el valor dado a σ (ver la Figura 5.5(b)). En la Figura 5.5(e) se observa como para estos valores, la Gaussiana es redonda y aparece entera.

- En la segunda prueba se ha seleccionado un tamaño de la Gaussiana muy pequeño comparado con el valor de σ. En la Figura 5.5(c) se puede ver la parte superior de la campana. Por otro lado, si nos fijamos en la Figura 5.5(f) nos damos cuenta de que no aparece la Gaussiana completa, sino el centro de ésta.

Este análisis ha sido importante, debido a que en nuestro sistema queremos que las Gaussianas cubran completamente a los vehículos y que no aparezca sólo el centro de ellas.

Además de estos parámetros, el tamaño del diccionario es fundamental, como veremos en los siguientes experimentos. De nuevo, el error se mide como la media de la diferencia

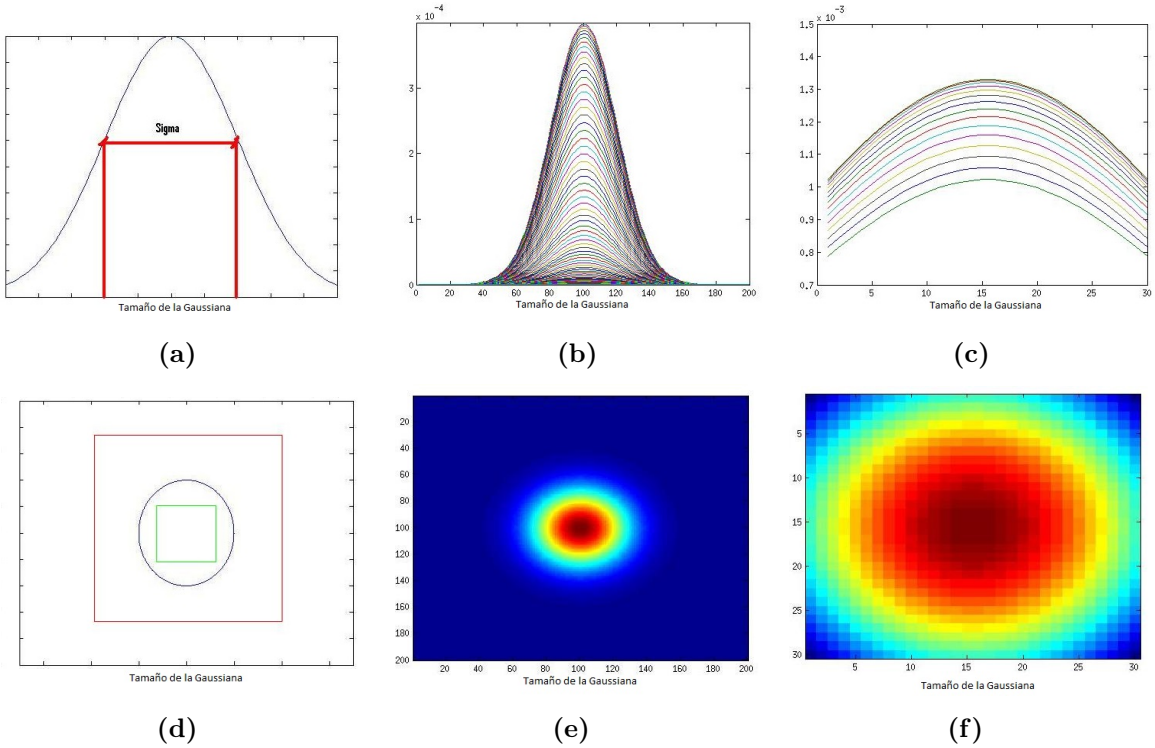


Figura 5.5: a) Indicación de los parámetros σ y tamaño de la Gaussiana. b) Gaussiana de tamaño mayor que σ. c) Gaussiana de tamaño menor que σ. d) Ejemplificación de lo que se aprecia en función de los valores que tomen los parámetros. e) Representación de lo que se extrae cuando el tamaño de la Gaussiana es mayor que σ. f) Representación de lo que se extrae cuando el tamaño de la Gaussiana es menor que σ.

entre el total de las anotaciones y el total de las densidades estimadas. La unidad de medida depende de lo que se esté contando, en este caso en concreto, el error medio vendrá dado por un número de vehículos.

En una primera ronda de experimentos, primero hemos procedido a ajustar el parámetro de regularización λ, tal y como hemos descrito previamente. En la tabla 5.2 se observa que el mejor resultado, para un tamaño de diccionario fijo de 200 es de $\lambda = 1000$. Como conclusión importante, se observa en la tabla que el sistema depende mucho del parámetro λ.

Fijando el λ obtenido, pero probando ahora con las imágenes de test, observamos que el error es similar, lo que garantiza que el procedimiento de validación es efectivo, garantizándose una homogeneidad en los resultados obtenidos con imágenes de validación y de test.

El error medio obtenido no es lo suficientemente bajo, lo que indica que el sistema no está contando correctamente. Si observamos la Figura 5.6(a) vemos que el valor es-

λ	**Error medio**
0.0001	37.446044
0.001	31.225479
0.01	18.352324
0.1	17.063401
1	16.910284
10	16.998290
100	16.921516
1000	**16.888200**
10000	16.942690
100000	16.920817
1000000	16.987068

Tabla 5.2: Errores medios obtenidos variando los valores asignados a λ.

	Regularización L1
Error medio	17.345227

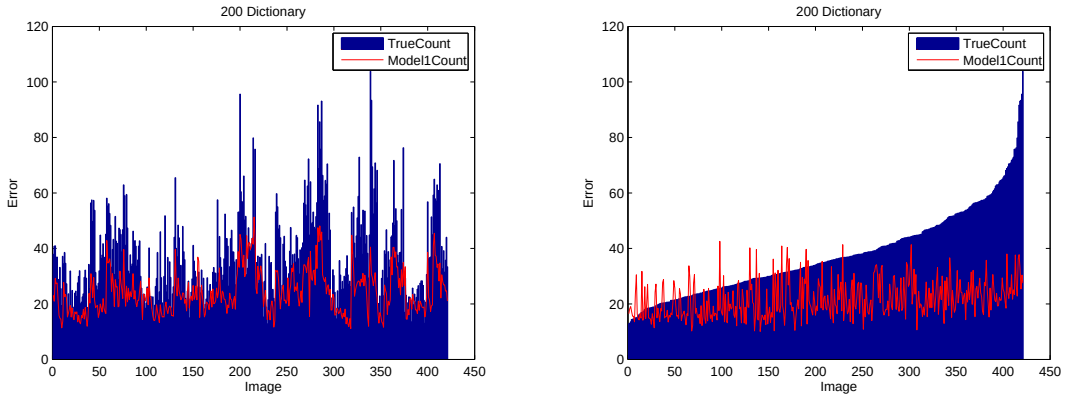
Tabla 5.3: Error medio obtenido para un diccionario de tamaño 200.

timado aumenta cuando lo hacen las anotaciones (valor verdadero obtenido a partir del *groundtruth*), si estos valores los colocamos de menor a mayor densidad de vehículos (ver Figura 5.6(b)) corroboramos el hecho de que no cuenta suficientemente bien, pues el valor estimado queda bastante alejado del verdadero. También se concluye que puede haber un valor residual ya que prácticamente el valor estimado varía de forma constante.

Las Gaussianas elegidas son lo suficientemente grandes como para cubrir los coches el cual era el objetivo que se perseguía (ver Figura 5.7) por lo que no podemos achacar los malos resultados a la elección de una incorrecta gaussiana.

En la Figura 5.8, se muestran algunos resultados cualitativos obtenidos para este experimento. Si nos fijamos en la imagen de la derecha para cada par de figuras, la representación de las densidades, el detalle más notable es que además de contabilizar los vehículos también detecta las líneas de la carretera, induciendo a errores. Además, para los vehículos más pequeños, situados en el fondo de la imagen, vemos cómo apenas se detecta nada, por lo que parece que para vehículos pequeños, el sistema no funciona adecuadamente.

Todo parece apuntar a que las características visuales que parece funcionan perfectamente con el caso de las células, no son lo suficientemente efectivas para resolver el problema del conteo de vehículos.



(a) **(b)**

Figura 5.6: a) Error medio para cada imagen. b) Error medio ordenado de menor número de vehículos a mayor.

(a) **(b)**

Figura 5.7: a) Imagen original. b) Gaussiana correspondiente con la anotación.

Otra posible causa de este error está en el tamaño del diccionario, es decir, que éste sea demasiado pequeño. Por este motivo se ha decidido probar con un diccionario de tamaño 2000.

A partir de los resultados mostrados en la Tabla 5.4 se deduce que para $\lambda = \mathbf{10000}$ se obtiene el error medio más bajo. Con este valor pero probando en las imágenes de test, obtenemos un error similar, pero menor que para el caso de diccionarios de tamaño 200 (ver Tabla 5.5).

	Regularización L1
Error medio	13.594236

Tabla 5.5: Error medio obtenido para un diccionario de tamaño 2000.

El error medio obtenido parece bastante alto, pero no es así dado que las imágenes

λ	**Error medio**
0.0001	37.446044
0.001	37.446044
0.01	20.763670
0.1	15.389460
1	14.927427
10	14.949020
100	14.960149
1000	14.945170
10000	**14.924895**
100000	14.973585
1000000	14.925697

Tabla 5.4: Errores medios obtenidos variando los valores asignados a λ.

empleadas tienen una alta densidad de tráfico lo que complica la tarea de contar los vehículos. Si observamos la Figura 5.9(a) vemos que el valor estimado aumenta cuando lo hace el valor de las anotaciones, lo que indica que el sistema está contando adecuadamente. Además en la Figura 5.9(b) se aprecia que a medida que aumenta el número de anotaciones también lo hace el valor estimado. Todo parece apuntar a que el tamaño del diccionario parece ser un parámetro fundamental del sistema.

Para este experimento, si observamos la Figura 5.10, más concretamente la imagen derecha para cada par, se aprecia que al aumentar el tamaño del diccionario se elimina bastante ruido en comparación con la Figura 5.8 (desaparecen las líneas de la carretera). Sin embargo, como sucedía para el experimento anterior, los vehículos más alejados, no son detectados, lo que induce a pensar que el sistema no funciona correctamente.

En el análisis que se ha realizado, parece claro que el tamaño del diccionario importa. Los resultados revelan que cuanto mayor es el diccionario, menor es el error cometido en la cuenta. A continuación vamos a analizar, partiendo del diccionario de tamaño 2000, si los parámetros que definen los tamaños de las Gaussianas afectan, y en que medida, a las estimaciones.

Aunque las Gaussianas anteriores cumplían con los objetivos deseados, se ha decidido probar con una Gaussiana de tamaño superior (tamaño: 15*6; σ=15) para asegurarnos que no sólo cubre los vehículos, si no también los camiones o autobuses que tienen un tamaño mayor.

En la Figura 5.11(b) se representan las Gaussianas que se han obtenido,en este caso al ser de mayor tamaño que las anteriores podemos observar que donde hay más densidad

de tráfico se genera una masa de Gaussianas, es decir, las Gaussianas se solapan. Este hecho puede incitar a que el sistema cuente como vehículos partes que se corresponden con la carretera introduciendo así ruido.

El primer paso, al igual que antes, ha sido determinar el parámetro λ que mejor calibra nuestro sistema. En este caso es $\lambda = \mathbf{1000}$ en las imágenes de validación (véase Tabla 5.9).

λ	**Error medio**
0.0001	36.287304
0.001	36.287304
0.01	19.751208
0.1	14.674509
1	14.032018
10	14.031634
100	13.910489
1000	**13.839141**
10000	13.905907
100000	13.863769
1000000	13.994107

Tabla 5.6: Errores medios obtenidos variando los valores asignados a λ.

A la hora de probar en la secuencia de test, el error medio obtenido aparece en la Tabla 5.7. Como puede observarse, este error es ligeramente inferior al obtenido con una Gaussiana menor (tamaño = 10*6; $\sigma = 10$), lo que nos indica que este cambio mejora nuestro sistema.

	Regularización L1
Error medio	12.915479

Tabla 5.7: Error medio obtenido para un diccionario de tamaño 2000 y una Gaussiana más grande (tamaño = 15*6; $\sigma = 15$) que la original (tamaño = 10*6; $\sigma = 10$).

Como sucedía antes, si observamos la Figura 5.12(b) vemos que el valor estimado aumenta cuando lo hace el valor de las anotaciones (valor verdadero), lo que indica que el sistema está contando adecuadamente.

A continuación en la Figura 5.13, se muestran algunos resultados representativos para el conteo de vehículos en las imágenes de test. Para este experimento, los resultados han mejorado respecto a los anteriores. El nivel de ruido es mínimo, no aparecen las líneas pintadas sobre la carretera, y las Gaussianas parecen cubrir los vehículos completamente. Además, si observamos el fondo de la imagen, donde los vehículos son más pequeños, se estiman densidades, por lo que el sistema funciona mejor que para el resto de experimentos.

A continuación y para finalizar, ofrecemos un estudio comparativo de los tres experimentos llevados a cabo. En la Figura 5.14 se muestran de manera comparativa los resultados obtenidos en los experimentos anteriores. Se concluye que los mejores resultados se obtienen para un diccionario de tamaño 2000 y una Gaussiana más grande (tamaño =15*6, σ=15))

A modo comparativo entre los tres experimentos, se ha realizado un análisis para tres tramos distintos de concentración de vehículos. El primer tramo es el que posee una menor concentración, para el segundo tramo se incrementa y por último, en el tercer tramo, existe una alta congestión de vehículos. En la Figura 5.15 se especifican los tres tramos seleccionados.

	Primer tramo	Segundo tramo	Tercer tramo
Experimento 1	6.8437	13.8803	31.1627
Experimento 2	4.9117	11.9377	23.8101
Experimento 3	4.6604	11.2316	22.7373

Tabla 5.8: Error medio obtenido para cada uno de los tramos.

Como era de esperar, en la Tabla 5.8, se observa que a medida que el nivel de congestión aumenta, el error medio se ve incrementado ya que se producen oclusiones debido a la alta concentración de tráfico. En línea con las conclusiones obtenidas previamente, se observa que el experimento 3, es el que mejor funciona.

Por último se ha analizado para qué imágenes cometen más error cada una de las 3 aproximaciones. Un hecho curioso, es que los 3 experimentos comenten el mayor error medio para la misma imagen. Esto se debe a que esta imagen es la que tiene un mayor número de vehículos anotados. En la Figura 5.16(a), si observamos la imagen de la derecha, vemos en las densidades cómo se introduce bastante ruido, centrándose más en los bordes de la carretera que en los propios vehículos. Sin embargo, en la Figura 5.16(b) y en la Figura 5.16(c), las densidades estimadas se centran más en los vehículos, en este caso la diferencia entre ambas es prácticamente inapreciable. No obstante, en la Figura 5.16(c), se estima ligeramente mejor los vehículos que en la Figura 5.16(b).

Experimento	**Imagen**	**Error medio**
1	339	76.3635
2	339	62.1181
3	339	61.3202

Tabla 5.9: Mayor error medio cometido para cada experimento.

Finalmente, tras analizar todos los resultados obtenidos, se concluye que el tamaño del diccionario juega un papel muy importante en el funcionamiento del sistema, y aunque en menor medida el tamaño de la Gaussiana también afecta.

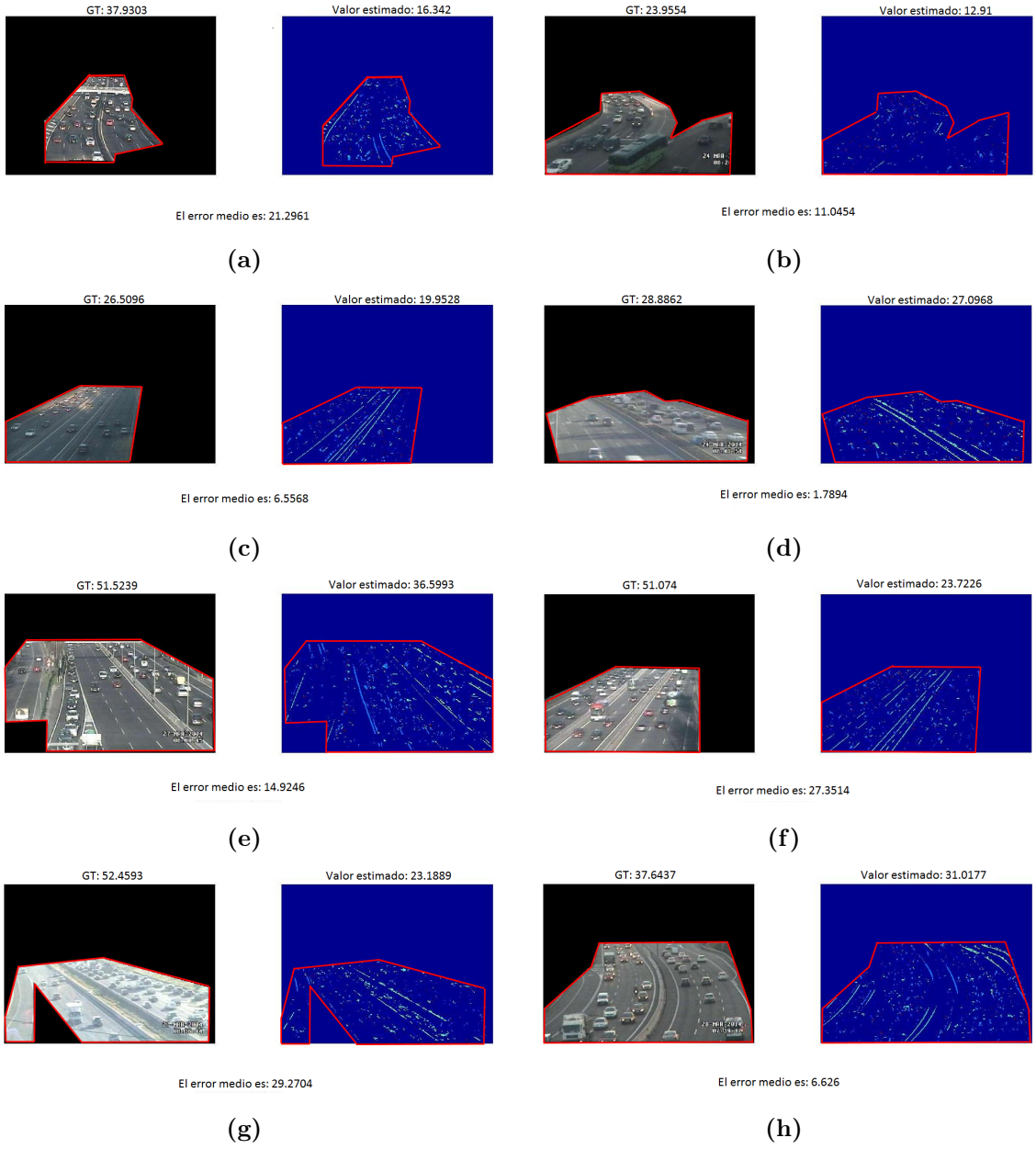
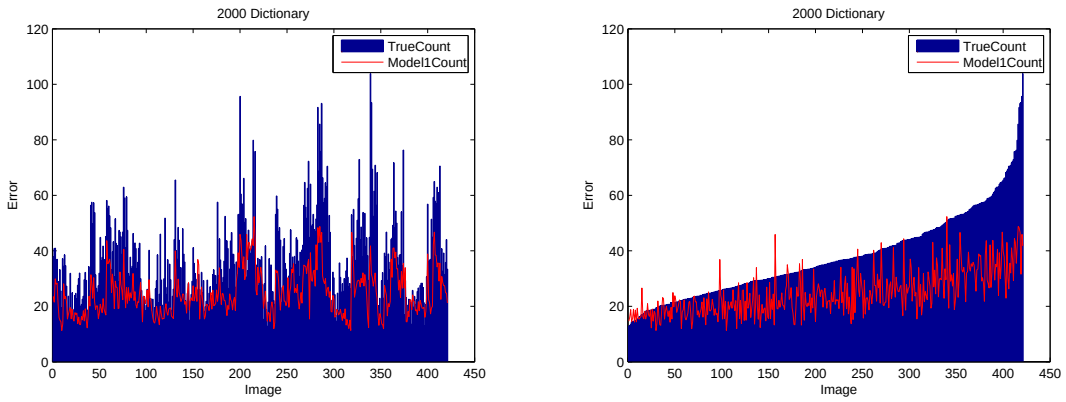


Figura 5.8: Errores medios obtenidos variando los valores asignados a λ.



(a) **(b)**

Figura 5.9: a) Error medio para cada imagen. b) Error medio ordenado de menor número de vehículos a mayor.

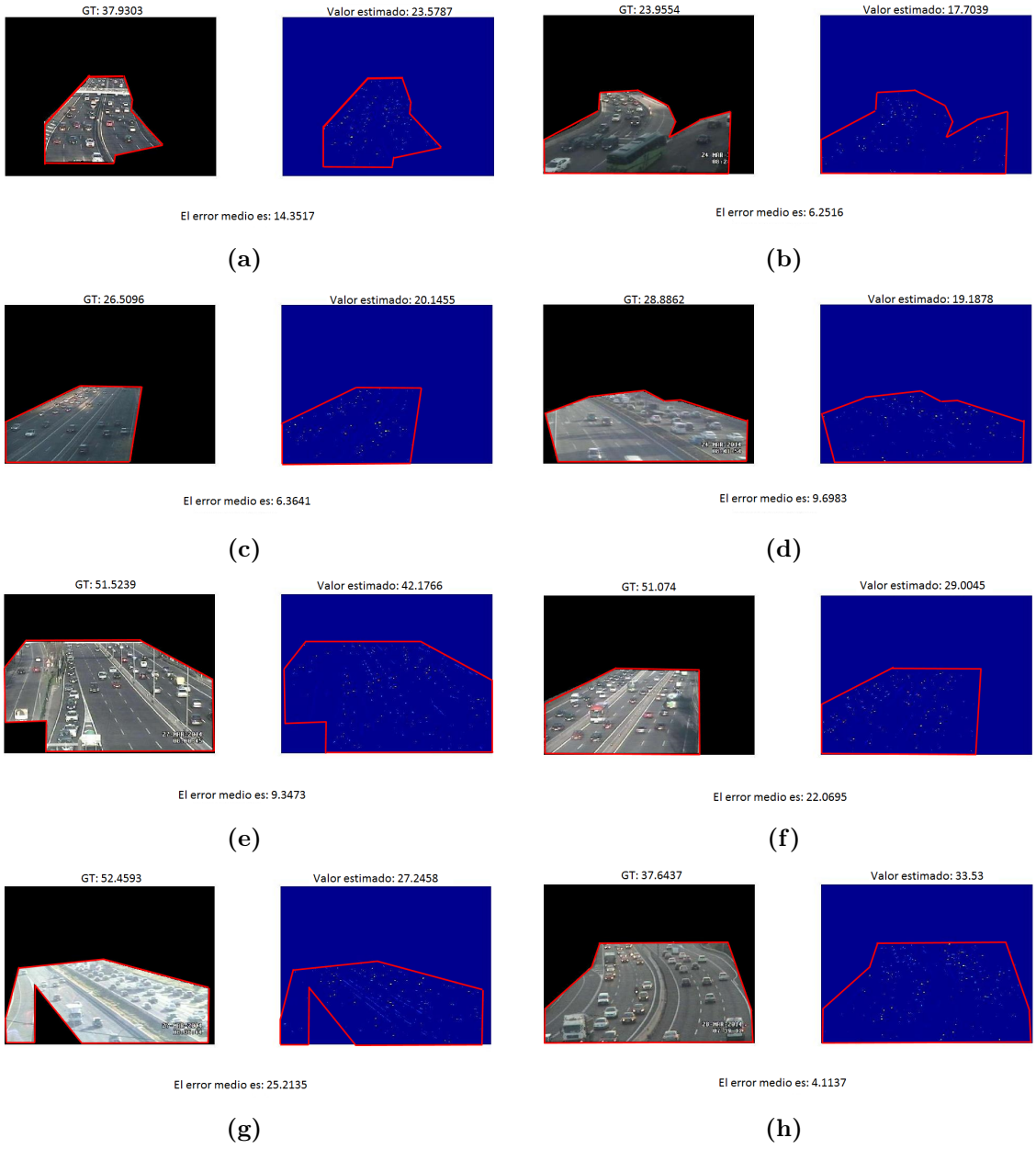
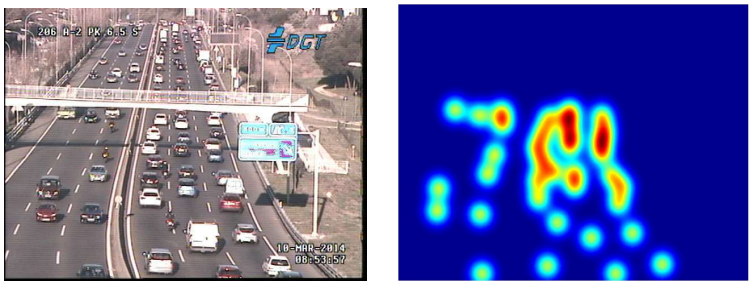
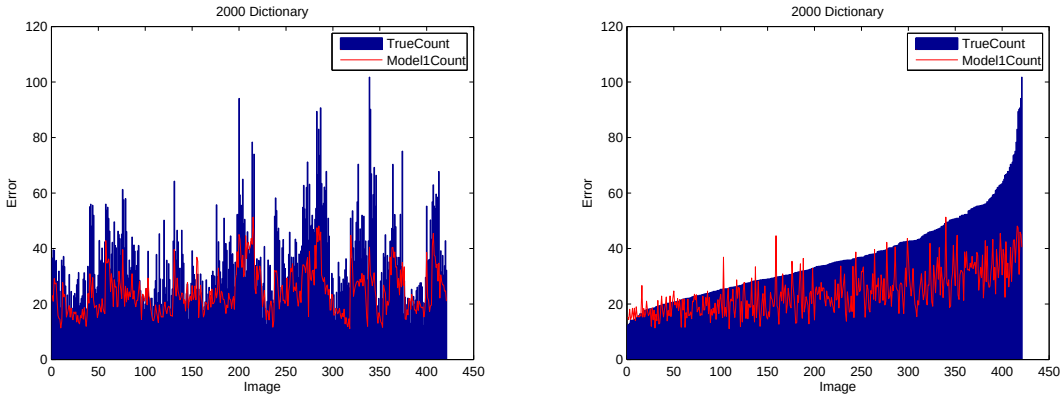


Figura 5.10: Errores medios obtenidos variando los valores asignados a λ.



(a) **(b)**

Figura 5.11: a) Imagen original. b) Gaussiana correspondiente con la anotación.



(a) **(b)**

Figura 5.12: a) Error medio para cada imagen. b) Error medio ordenado de menor número de vehículos a mayor.

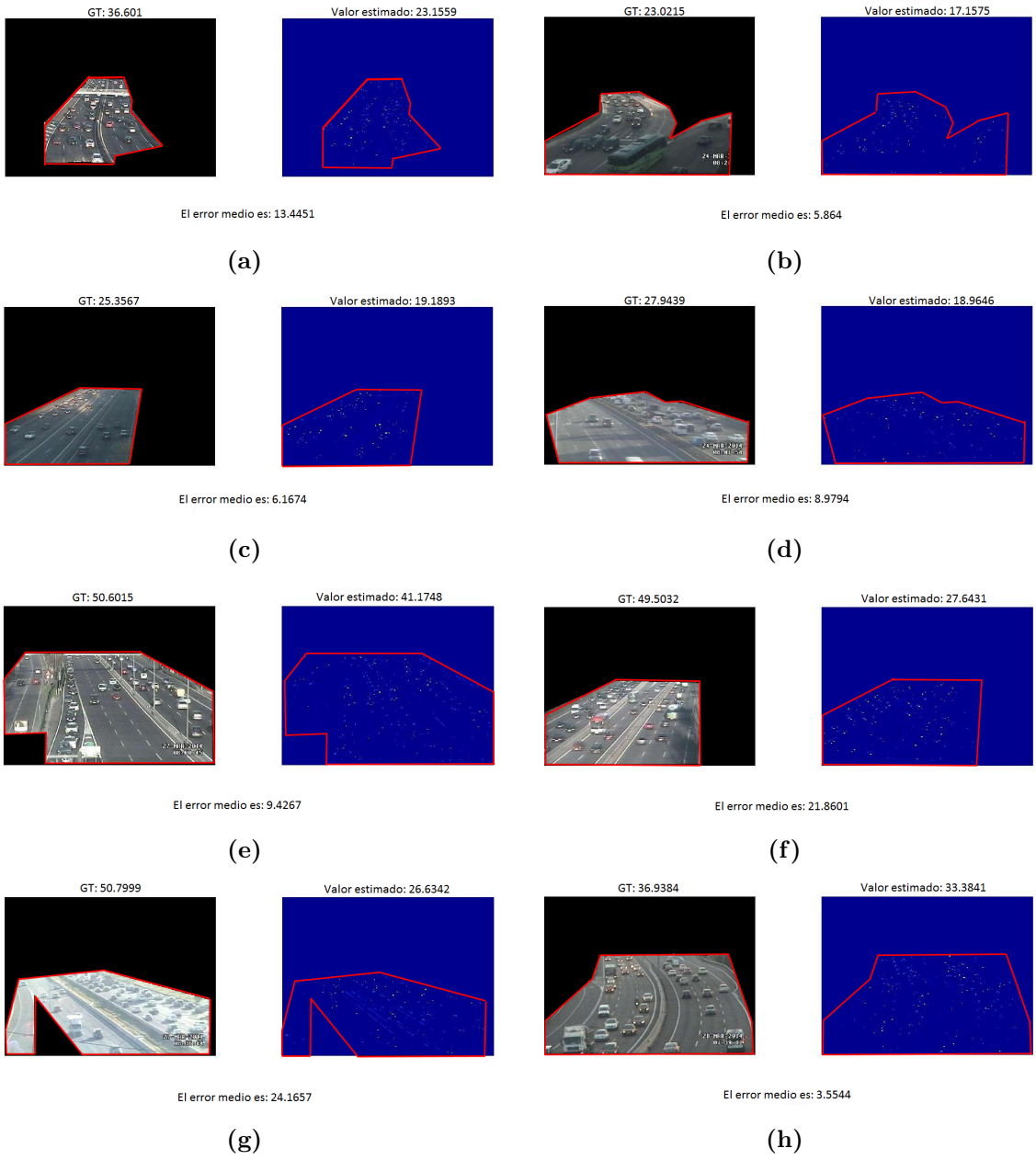


Figura 5.13: Errores medios obtenidos variando los valores asignados a λ.

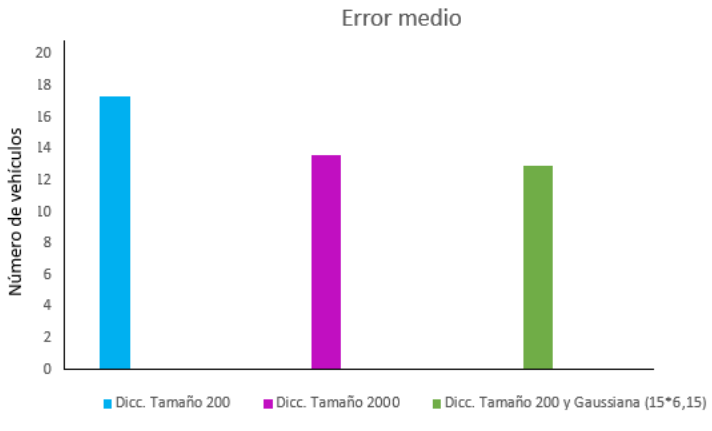


Figura 5.14: Error medio obtenido para los tres experimentos realizados.

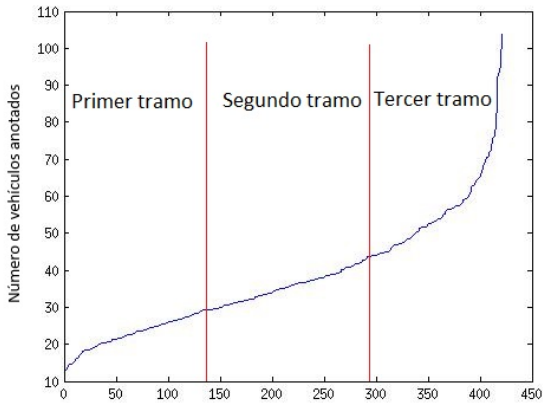
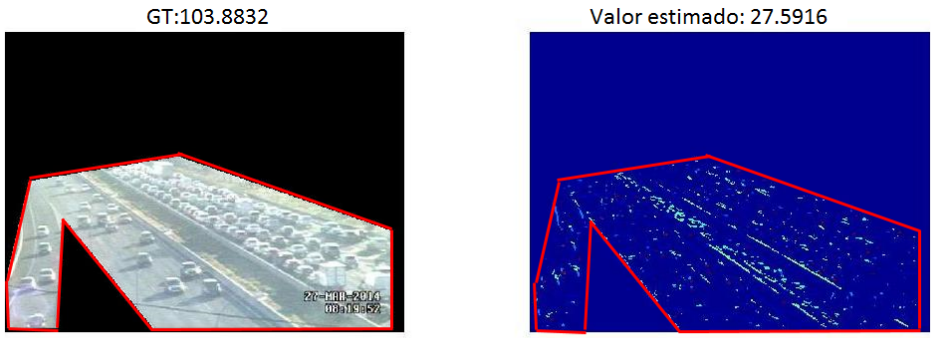
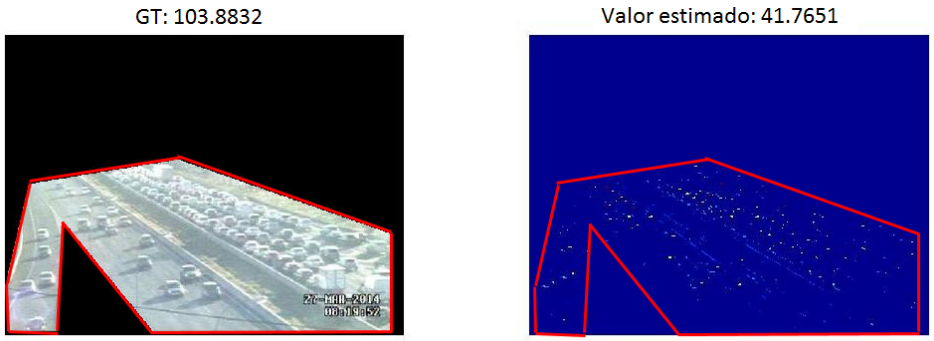


Figura 5.15: División de tramos en función del nivel de congestión.



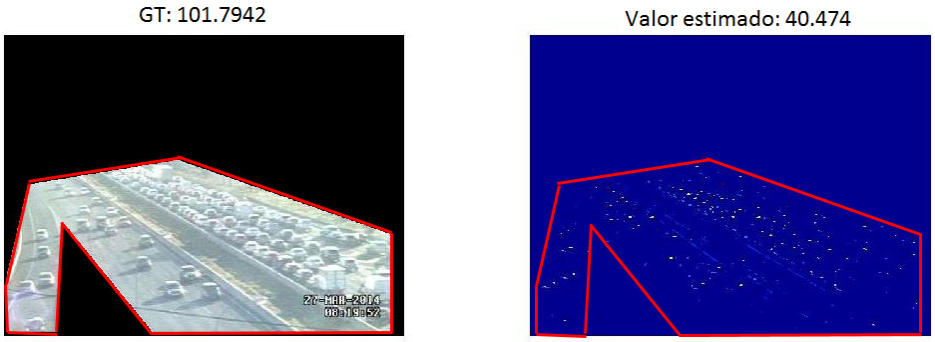
El error medio es: 76.3635

(a)



El error medio es: 62.1181

(b)



El error medio es: 61.3202

(c)

Figura 5.16: Resultados cualitativos para las imágenes con mayor error medio. a) Experimento 1. b) Experimento 2. c) Experimento 3.

Capítulo 6

Conclusiones y futuras líneas

6.1. Conclusiones

Una vez analizados los resultados obtenidos a partir de los experimentos realizados en el Capítulo 5, podemos concluir que los resultados son satisfactorios, aunque se esperaba conseguir errores medios más bajos. No obstante estos resultados cumplen con los objetivos que teníamos, dada la alta complejidad de la base de datos creada. Las conclusiones a las que se ha llegado son las siguientes:

- La anotación realizada es muy sencilla y natural, ya que se asemeja a la forma normal de contar para los seres humanos. Además con las nuevas tecnologías, dispositivos táctiles, se puede hacer de forma muy rápida

- Un diccionario pequeño no funciona correctamente para nuestro sistema. Es necesario un diccionario mayor para evitar que aparezcan densidades en zonas que no contienen vehículos.

- Una buena elección del parámetro λ es muy importante, ya que una correcta calibración del sistema permite conseguir mejores resultados.

- Se debe ser cauteloso cuando se elijan los parámetros característicos de las Gaussianas debido a que la incorrecta relación entre el tamaño de la Gaussiana y su Sigma (σ) puede llevar a no observar lo que se desea. Además si el tamaño no es el adecuado respecto al tamaño de los vehículos, las densidades estimadas podrían no ser correctas.

La conclusión más importante obtenida, es que se pueden contar objetos de forma adecuada a partir de una simple estimación de densidades. Esto se debe a la introducción del modelo de regresión de densidades propuesto en [16]. Este enfoque demuestra que a partir de unos datos limitados de entrenamiento se puede predecir con bastante exactitud, sin necesidad de detectar objetos individualmente.

6.2. Futuras líneas

En base a las conclusiones anteriores, a los experimentos realizados y a las ideas que han ido surgiendo a lo largo de la ejecución de este proyecto se proponen algunas tareas que se deberían seguir en futuras líneas de investigación, que permitan una mejora en los resultados:

- Utilizar otro tipo de vectores para la extracción de características o añadir determinadas características, como podría ser el color de los vehículos. Al ayudarnos del color de lo vehículos, se debe tener en cuenta que también habría error para aquéllos vehículos con colores similares a los del asfalto. Lo que está claro es que las características han jugado y jugarán un papel muy importante en futuras implementaciones.

- Se ha deducido que las Gaussianas deben cubrir los vehículos sin ser excesivamente grandes. Una mejora para el sistema sería escalar el tamaño de las Gaussianas en función de la profundidad de las imágenes, debido a que a medida que los vehículos se alejan de la cámara se hacen más pequeños. En la Figura 6.1 se observa como a medida que los vehículos se van alejando de la cámara van disminuyendo de tamaño. Los círculos rojos representan las Gaussianas, las cuales podrían variar su tamaño en función de la profundidad de la imagen pudiendo conseguirse errores medios más bajos.

Figura 6.1: A medida que nos alejamos los vehículos se van haciendo más pequeños.

- Aplicar métodos que mejoren la calidad de la imagen y permitan identificar mejor los vehículos como por ejemplo técnicas de super resolución.

Dado que el sistema desarrolla permite contar el número de vehículos, surgen aplicaciones relacionadas con las ciudades inteligentes, tales como un control de la congestión del tráfico a los accesos de núcleos urbanos que permita reducir la contaminación acústica

y ambiental o el desarrollo de un sistema de análisis de la congestión para mejorar la eficiencia energética en entornos urbanos.

Bibliografía

[1] N. Ahuja and S. Todorovic. Extracting texels in 2.1d natural textures. In *ICCV*, 2007.

[2] H. Bay, T. Tuytelaars, and L Van Gool. Surf: Speeded up robust features. In *Computer Vision - ECCV 2006*, pages pp 404 – 417, 2006.

[3] J.L. Bentley. Programming pearls: Perspective on performance. *Comm. ACM*, 27(11):1087–1092, 1984.

[4] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008.

[5] S. Y. Cho, T. W. W. Chow, and C.T. Leung. A neural based crowd estimation by hybrid global learning algorithm. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 29(4), pages 535 – 541, 1999.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results, 2009.

[9] V. Ferrari, F Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *IEEE*, 2007.

[10] J. Gall and V Lempitsky. Class-specific hough forests for object detection. In *IEEE*, 2009.

[11] J. Gall, A. Yao, N Razavi, L Van Gool, and V Lempitsky. Hough forests for object detection, tracking, and action recognition. Technical report, 2011. Disponible en http://www.robots.ox.ac.uk/ vilem/tpami2011.pdf.

[12] R. Guerrero-Gomez-Olmedo, R. J. Lopez-Sastre, S. Maldonado-Bascon, and A. Fernandez-Caballero. Vehicle tracking by simultaneous detection and viewpoint estimation. In *IWINAC 2013, Part II, LNCS 7931*, pages 306–316, 2013.

[13] S. JunFang, S. ANing, and X. Ru. A reliable counting vehicles method in traffic flow monitoring. In *Image and Signal Processing (CISP)*, pages 522 – 524, 2011.

[14] N. Y. Khan, B. McCane, and G. Wyvill. Sift and surf performance evaluation against various image deformations on benchmark dataset. In *2011 International Conference on Digital Image COmputing: Techniques and Applications*, 2011.

[15] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *ICPR*, volume 3, pages 1187–1190, 2006.

[16] V. Lempitsky and A. Zisserman. Learning to count objects in images. Technical report, 2010. Disponible en http://www.robots.ox.ac.uk/ vgg/research/counting/-NIPS2010.pdf.

[17] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference*, volume 2, pages 1150–1157. IEEE, Sept. 1999.

[18] A. N. Marana, S.A. Velastin, L.F. Costa, and R. A. Lotufo. Estimation of crowd density using image proceessing. In *Image Processing for Security Applications*, pages 1 – 8, 1997.

[19] MATLAB. `http://www.mathworks.es/es/help/images/ref/roipoly.html`.

[20] H.S. Mohana, M. Ashwathakumar, and G. Shivakumar. Vehicle detection and counting by using real time traffic flux through differential technique and performance evaluation. In *International Conference on Advanced Computer Control. ICACC'09.*, pages 791 – 795, 2009.

[21] V. Rabaud and S. Belongie. Counting crowded moving objects. In *ICCV*, 2006.

[22] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. *DICTA '09: Proceedings of the 2009 Digital Image Computing: Techniques and Applications*, pages 91–88, 2009.

[23] H Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56:151–177, 2004.

[24] A. Sivic, J. & Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV 2003*, 2003.

[25] B. Tamersoy and J.K. Aggarwal. Counting vehicles in highway surveillance videos. In *International Conference on Pattern Recognition (ICPR)*, 2010.

[26] A. Vedaldi. An implementation of sift detector and descriptor. Technical report. Disponible en http://www.robots.ox.ac.uk/ vedaldi/assets/sift/sift.pdf.

[27] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[28] A. Vedaldi and A. Zisserman. Instance-level recognition practical. Technical report, 2012. Disponible en http://www.robots.ox.ac.uk/ vgg/share/SearchPractical2012.html.

[29] Pavel Zemčík, Adam Herout, Vítězslav Beran, Igor Potúček, Otto Fučík, Jozef Honec, Miloslav Richter, Ilona Kalová, and Marek Lisztwan. Image processing in traffic applications. In *Proceedings of GVIP 2005*, page 6, 2005.