

COMPARATIVA DE TÉCNICAS DE PROCESAMIENTO DE IMÁGENES PARA LA DETECCIÓN DE PEATONES CON SVM

Álvaro Ruiz Gutiérrez, Lidia Jiménez Soriano, Víctor Flores González, Óscar Menéndez Márquez

25 de febrero de 2025

Resumen

En este trabajo se analiza y compara la eficiencia de cuatro métodos de extracción de características en imágenes: Harris Corner Detection, SURF, SIFT y HOG, aplicados a la detección de peatones mediante un modelo SVM preentrenado. Se estudia la teoría detrás de cada método, extrayendo y explicando sus principios fundamentales de manera didáctica. Posteriormente, se implementan o adaptan implementaciones existentes, detallando los parámetros utilizados y los efectos que tienen en el rendimiento del modelo. Para evaluar la eficiencia de cada método, se definen métricas clave como precisión, recall, F1-score y tiempo de procesamiento. Se presentan los resultados obtenidos mediante experimentación con un conjunto de datos de imágenes, analizando el impacto de cada técnica en la identificación de peatones. Finalmente, se expone de manera clara y estructurada el funcionamiento de cada método, utilizando ejemplos gráficos y explicaciones detalladas. Este estudio proporciona una visión comparativa de estos enfoques en el contexto de la visión por computador, facilitando la selección de técnicas óptimas para aplicaciones similares.

Palabras clave: Harris Corner Detection, SURF, SIFT, HOG, procesamiento de imágenes, detección de peatones, SVM.

1. Introducción

El procesamiento de imágenes digitales es un campo clave dentro de la visión por computador, con aplicaciones en reconocimiento de patrones, inteligencia artificial y análisis de imágenes. Dentro de este ámbito, la detección de objetos y, en particular, la identificación de peatones, representa un desafío fundamental con implicaciones en seguridad, movilidad y automatización.

Este trabajo se centra en el estudio y comparación de cuatro técnicas de extracción de características utilizadas en la identificación de peatones: **Harris Corner Detection**, **SURF (Speeded Up Robust Features)**, **SIFT (Scale-Invariant Feature Transform)** y **HOG (Histogram of Oriented Gradients)**. Estas técnicas permiten identificar estructuras relevantes en una imagen, facilitando su uso en tareas de reconocimiento y clasificación. Para evaluar su rendimiento, se utilizará un **modelo SVM (Support Vector Machine) preentrenado**, el cual procesará los descriptores generados por cada método para realizar la identificación de peatones en imágenes.

El objetivo principal de este trabajo es ofrecer una explicación didáctica de los métodos estudiados, tanto desde un punto de vista teórico como práctico. Para ello, el desarrollo del proyecto se estructura en tres fases clave:

- **Análisis teórico:** Se investigarán los principios fundamentales de cada método, extrayendo las ideas principales de fuentes científicas, sin recurrir a transcripciones literales.
- **Implementación práctica:** Se desarrollará o adaptará la implementación existente de cada técnica, detallando los pasos intermedios, el significado de los parámetros y la experimentación con distintas configuraciones.
- **Presentación didáctica:** Se explicarán de forma estructurada todos los pasos necesarios para la detección de peatones utilizando cada método, complementando con ejemplos gráficos y análisis de resultados.

Dado que el desarrollo del trabajo está orientado a la investigación y aplicación de técnicas avanzadas no vistas en clase, se apoyará en artículos científicos recientes y capítulos de libros especializados en visión por computador. La planificación y ejecución del proyecto se realizará en un equipo de trabajo, distribuyendo **70 horas por persona**, según lo indicado en las directrices académicas. Además, se utilizarán herramientas de gestión de proyectos para organizar y documentar las tareas realizadas, asegurando un seguimiento adecuado de los avances.

Este documento recoge el desarrollo del trabajo en sus distintas fases, proporcionando un análisis comparativo de los métodos seleccionados, su implementación y evaluación en la detección de peatones mediante un modelo SVM.

2. Planteamiento teórico

La sección *Planteamiento teórico* debe plantear el problema a resolver, así como describir los algoritmos propuestos para su resolución.

La realización del trabajo está programada en **4 fases**, cada una de las cuales termina con un **Seguimiento**.

FASE 1

En la primera fase, **los miembros del grupo** realizarán la **propuesta del trabajo dirigido** (TD), acordando con el profesorado el trabajo a realizar.

Para la búsqueda de libros y bases de datos de revistas, se puede consultar el catálogo fama de la US <http://fama.us.es/>. Se recomiendan las bases de datos ScienceDirect (Elsevier), IEEE Xplore y Springerlink. Si se accede desde fuera de la Universidad, se tendrá que usar el UVUS. Se pueden encontrar recursos para buscar ideas en el documento *Enlaces de interés* del contenido en ev.

Antes de que termine la **Fase 1**, se debe entregar la **planificación inicial** y la **lista de objetivos** del TD. Se recuerda que en la planificación se debe detallar cómo se distribuirán las **70 horas/persona** que el grupo debe dedicar al TD, incluidas las sesiones de clase dedicadas al trabajo. En esta fase debe comprenderse la teoría relativa al trabajo a realizar y quedar perfectamente definido.

FASE 2

En esta fase se comenzará a trabajar en la parte práctica y explicación del método. Se recomienda comenzar a escribir la documentación desde el principio.

Al finalizar la Fase 2, se mostrará una primera versión de la implementación, algo mínimo. El algoritmo (o algoritmos) a implementar debe estar perfectamente claro y diseñado.

FASE 3

En la Fase 3 se seguirá con la realización del trabajo. Al final de la Fase 3, se mostrará una primera versión de la documentación y segunda de la implementación, ahora ya sí, un producto mínimo viable. Se mostrará la hoja de tiempos realizada hasta ese momento y **se revisarán los objetivos iniciales** planteados.

FASE 4

En la última fase se realizará la experimentación para poder evaluar y comparar el método implementado y poder inferir conclusiones. Al final de la Fase 4, se tendrá una primera versión de la presentación (aunque sólo sea un índice), junto con una revisión de la documentación y de la implementación.

En la siguiente tabla se resumen los hitos que deben alcanzarse (y entregarse) al final de cada Fase:

Seguimiento 1	Propuesta del TD basándose en una fuente científica. Planificación inicial y lista de objetivos. Estudio y comprensión del método elegido. Hoja de tiempos individual.
Seguimiento 2	Primera versión de la implementación. Hoja de tiempos individual.
Seguimiento 3	Primera versión de la documentación y segunda de la implementación. Hoja de tiempos individual. Revisión de los objetivos.
Seguimiento 4	Primera versión de la presentación. Revisión de la documentación y de la implementación.

Los días dedicados a cada Seguimiento aparecen en el Calendario publicado en ev.

Todos los miembros del grupo deberán **intervenir en la exposición final**, en la que deberán defender su trabajo y comentar los logros teóricos y de implementación obtenidos. Además, deberá hacerse algún ejemplo usando la implementación desarrollada (**20 minutos en total**). Las exposiciones se realizarán en horario de clase en las dos últimas semanas. Después se abrirá un turno de preguntas.

Para la entrega, cada grupo deberá subir, en respuesta a una actividad creada para tal efecto en la ev, los siguientes elementos:

- **documentacion.pdf**. Consiste en un fichero **.pdf** correspondiente a la documentación.

Estructura de la documentación:

- Título
- Autores
- Resumen
- 1. Introducción
- 2. Planteamiento teórico
- 3. Implementación
- 4. Experimentación
- 5. (Manual de usuario)
- 6. Conclusiones
- 7. Autoevaluación de cada miembro
- 8. Tabla de tiempos
- Bibliografía

Se puede usar el fichero **main.tex** disponible en enseñanza virtual como plantilla de LaTeX¹, que es el que se ha usado en este documento. Se recomienda el uso de editores online de LaTeX como **Overleaf.com**.

Las referencias se citan así: bla, bla [2], bla, bla [1]. La bibliografía debe seguir el estilo de este documento, con las referencias ordenadas alfabéticamente por autores.

También se puede documentar el trabajo en un Notebook de Jupyter, en el que también se puede incluir código LaTeX e imprimirlo como pdf.

- **codigo.zip**. Si se ha desarrollado una aplicación propia, debe contener todo el código utilizado, comentado y modulado. Debe contener el ejecutable, si existiera, junto con todos los archivos necesarios para su ejecución de forma que no dé errores de compilación. Debe contener un **leeme.txt** con instrucciones para ejecutar el código. Se debe incluir una carpeta **images** con imágenes de muestra, incluyendo las imágenes usadas en la experimentación.

En lugar de subir el código a la enseñanza virtual, se puede subir a un repositorio y referenciarlo en la documentación.

En caso de que exista ya una implementación en un repositorio asociado al trabajo escogido, se debe desarrollar una versión propia que incluya explicaciones, paso a paso, ejemplos, modificaciones, etc. Por ejemplo, se puede hacer un notebook de jupyter para esto.

¹Véase el Anexo para algunos apuntes rápidos sobre escritura de textos en LaTeX (tomados de **Overleaf.com**).

- **presentacion.pdf**. Consiste en un fichero **.pdf** con la presentación que se usará en clase para exponer el trabajo.

La evaluación del trabajo se realizará en base a una rúbrica que se proporcionará en Blackboard.

3. Implementación

Esta sección de la documentación debe incluir la descripción de la implementación (pero no el código), especificando las tecnologías usadas, el diseño, módulos o partes principales. **Debe quedar muy claro cuál es la parte original implementada en el programa, qué librerías se han usado y de dónde se han cogido, con referencias apropiadas.** Se debe describir (tanto en la aplicación, como en la documentación, como en la presentación) los pasos seguidos para resolver el problema que se plantea de la forma más divulgativa posible.

Una opción muy práctica y recomendable es la de usar la librería **OpenCV** con Python mediante el gestor de paquetes de Anaconda <https://www.anaconda.com> y usar los notebook de Jupyter en los que se puede intercalar ejecución de código con explicaciones teóricas.

Siempre se debe redactar un fichero **Leeme explicando todo lo necesario para poder ejecutar y usar la implementación** en cualquier ordenador (o al menos con el mismo sistema operativo).

4. Experimentación

Una sección de ejemplos comentados y pruebas realizadas con el programa desarrollado es imprescindible en este trabajo. Si existen distintos parámetros que intervienen en el método, se deben comprender y experimentar con ellos. También puede ser de interés probar distintos tipos de imágenes dentro del contexto del trabajo. Igualmente importantes (o más) serán las conclusiones que se puedan obtener de la experimentación realizada.

5. Manual de usuario

Se debe incluir un breve manual de usuario en caso de que se desarrolle una aplicación. Si se implementa un notebook de Jupyter, no es necesario un manual de usuario, sino que debe quedar muy guiado, explicando todos los pasos y parámetros.

6. Conclusiones

Se debe introducir una sección de conclusiones que incluyan propuestas claras de mejora o extensión del trabajo (por ejemplo, si no se han podido alcanzar todos los objetivos iniciales). También conclusiones sobre los resultados obtenidos, en qué medida difieren de los esperados. además, son apropiadas conclusiones sobre las desviaciones en cuanto a la planificación inicial, así como conclusiones sobre la experiencia de la realización del trabajo (lecciones aprendidas).

7. Autoevaluación

Siguiendo la rúbrica dada, se debe detallar, por parte de **cada miembro del grupo**, la puntuación para el trabajo que se considera justa en cada apartado.

8. Tabla de tiempos

Se debe justificar el trabajo realizado por cada componente del grupo, indicando el **tiempo total que cada miembro del grupo ha dedicado** al trabajo (lo que puede implicar diferencia de notas obtenidas por los distintos miembros del grupo). El trabajo realizado debe ser de **70 horas por estudiante**. Además, debe haber un plan de trabajo **individual** detallado con las **tareas realizadas por cada miembro** del grupo. Para esto último, se puede usar la tabla siguiente o bien documentos generados por la herramienta de gestión de proyectos que se use, como Projetsii o Microsoft Project, por ejemplo.

Fecha de la actividad	Tiempo (h)	Miembro	Actividad realizada

Referencias

- [1] U. N. Experto, *Un libro que escribí*, Editorial, 1996.
- [2] Y. O. Mismo, “Título del artículo”, *Revista Publicación Periódica*, Vol. 17, pp. 1-100, 1997.

More about how to write an article

8.1. How to include Figures

First you have to upload the image file from your computer using the upload link the project menu. Then use the `includegraphics` command to include it in your document. Use the figure environment and the caption command to add a number and a caption to your figure. See the code for Figure 1 in this section for an example.



Figure 1: This frog was uploaded via the project menu.

8.2. How to add Tables

Use the `table` and `tabular` commands for basic tables. See Table 1, for example.

Item	Quantity
Widgets	42
Gadgets	13

Cuadro 1: An example table.

8.3. How to write Mathematics

L^AT_EX is great at typesetting mathematics. Let X_1, X_2, \dots, X_n be a sequence of independent and identically distributed random variables with $E[X_i] = \mu$ and $\text{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \dots + X_n}{n} = \frac{1}{n} \sum_i^n X_i$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

8.4. How to create Sections and Subsections

Use section and subsections to organize your document. Simply use the section and subsection buttons in the toolbar to create them, and we'll handle all the formatting and numbering automatically.

8.5. How to add Lists

You can make lists with automatic numbering or bullet points . . .

1. Like this,
2. and like this.
 - Like this,
 - and like this.